

Factsheet

Tests – Video streaming

# Video streaming

## Netflix (TCP)

Supported clients: Whiteboxes, Routers

The Netflix test is an application-specific test, supporting the streaming of binary data from Netflix's servers using the same CDN selection logic as their real client uses. The test has been developed with direct cooperation with Netflix.

The test begins by calling a Netflix hosted web API. This API examines the client's source IP address and uses the existing proprietary internal Netflix logic to determine which Netflix server this user's IP address would normally be served content from. This logic will consider the ISP and geographic location of the requesting IP address. Where the ISP participates in Netflix's Open Connect programme, it is likely that one of these servers will be used. The API will return to the client a HTTP 302 redirect to a 25MB binary file hosted on the applicable content server.

The test will then establish an HTTP connection to the returned server and attempt to fetch the 25MB binary file. This runs for a fixed 20 seconds of real-time. HTTP pipelining is used to request multiple copies of the 25MB binary, ensuring that if the payload is exhausted before the 20 seconds are complete, we can continue receiving more data. The client downloads data at full rate throughout; there is no client-side throttling taking place.

It's important to note that this 25MB binary content does not contain video or audio; it is just random binary data. However, with knowledge of the bitrates that Netflix streams content at, we can treat the binary as if it were video/audio content operating at a fixed

rate. This allows us to determine the amount of data consumed for each frame of video (at a set bitrate) and the duration that it represents. Using this, we then can infer when a stall occurred (by examining when our simulated video stream has fallen behind real-time). The test currently simulates videos at bitrates of 235Kbps, 375Kbps, 560Kbps, 750Kbps, 1050Kbps, 1750Kbps, 2350Kbps, 3000Kbps, 4500Kbps, 6000Kbps and 15600Kbps.

This approach also allows us to derive the 'bitrate reliably streamed', using the same methodology as the YouTube test. A small difference here is that we do not need to restart the download at a lower bitrate if a stall is encountered; because the incoming stream of binary data is decoded at a simulated bitrate, we can simply recompute the playback characteristics of the same network stream at a different bitrate entirely on the client side. This simply means that the test uses a predictable amount of bandwidth, even in cases where stalls occur.

The key outputs from this metric are:

- The bitrate reliably streamed
- The startup delay (the time taken to download two seconds of video)
- The TCP connection time
- The number of stalls and their duration
- The downstream throughput achieved

## YouTube (TCP)

Supported clients: Whiteboxes, Routers, Android, iOS

The YouTube test is an application-specific test, supporting the streaming of video and

audio content from YouTube using their protocols and codecs.

The test begins by seeking out the most popular video in the user's country. This is achieved by fetching a list of the most popular YouTube videos from a central SamKnows server. The central list of videos is refreshed once every 12 hours using the YouTube API. We filter for videos that are at least 60 seconds in length and have an HD quality variant. Note that by interacting with the YouTube API from a central location we can ensure that every probe is delivered the same list of videos.

The test running on the probe will now fetch the YouTube web page for the most popular video, and parse the JavaScript contained within the page. Within this JavaScript is held a list of all the encodings of the video in question and the content server hostname. By making this request from the probe we ensure that the test is receiving the same content server as the user would if they were using a desktop computer on the same connection.

The test will then connect to the content server (using whatever server YouTube would normally direct a real client on the same connection to) and begins streaming the video and audio. MPEG4, WebM, Dash (adaptive) and Flash video codecs are supported. Although the adaptive codec is supported, the test does not actually adapt its rate; we stream at full rate all the time, which provides for reproducibility.

The test parses video frames as it goes, capturing the timestamp contained within each video frame. After each frame, we sample how much real-time has elapsed versus video time. If video time > real-time at a sample period, then an underrun has not occurred. Otherwise, one has occurred.

The test downloads 10 seconds of audio and video at a time, with a buffer of 40 seconds. So, on startup, the test will immediately download (at full speed) 40 seconds of video

and audio, and will then download more as required, keeping the 40 second playback buffer full. By default, the test will run for a fixed duration of 20 seconds of real-time.

In its default mode of operation, the test will capture the 'bitrate that can be reliably streamed' on the user's connection. This is achieved through the following process:

1. Find the fastest recent speedtest result that the probe has completed.
2. As described above, fetch the list of YouTube videos, find the most popular one, and then select the highest bitrate encoding which is less than the fastest speedtest result found in step 1.
3. Attempt to stream this video, for a fixed duration of 20 seconds of real-time. If successful, then the "bitrate reliably streamed" for this instance is the bitrate that we just fetched.
4. However, if a stall event occurs, then we immediately abort the test and retry at the next lower bitrate.
5. If we find a bitrate that we can stream without a stall event occurring, then that bitrate is our "bitrate reliably streamed" for this instance.
6. However, if we encounter stalls for every bitrate, then the "bitrate reliably streamed" is zero.

The key outputs from this metric are:

- The bitrate reliably streamed
- The startup delay (the time taken to download two seconds of video)
- The TCP connection time
- The number of stalls and their duration (this is only applicable if the test is not running in the 'bitrate reliably streamed' mode)

#### BBC iPlayer (TCP)

Supported clients: Whiteboxes, Routers

The BBC iPlayer test is an application-specific test, supporting the streaming of video and audio content from iPlayer using their protocols and codecs.

The test begins by fetching a list of the most popular videos from an iPlayer XML API. The most popular video is chosen for playback, on the basis that this is most representative of what users will be watching at the time. Moreover, if there are iPlayer caches present in the ISP's network then this content is more likely to be cached there.

The XML manifest for this video is then fetched from the BBC's webservers. This contains paths to all of the different bitrates that this video is encoded at, and the different CDNs that serve them. At the time of writing BBC iPlayer uses multiple CDNs to serve content. By having the probe directly fetch the XML manifest we can ensure that any decisions the BBC make (e.g. "ISP X should always be served by CDN Y") are followed by our test. The test parses the XML manifest file, honouring the priority assigned to each CDN and building an ordered list of available bitrates.

At this point the test can begin to fetch video content from the content server. This is currently achieved over RTMP, mimicking the iPlayer web browser client. The test parses video frames as it goes, capturing the timestamp contained within each frame. After each frame we sample how much real-time has elapsed versus video time. If video time > real-time at a sample period, then an underrun has not occurred. Otherwise, one has occurred.

As with the YouTube and Netflix tests, the client is configured to start testing at the highest supportable bitrate and then step down if and when stalls occur. This allows us to identify the 'bitrate reliably streamed'.

The key outputs from this metric are:

- The bitrate reliably streamed
- The startup delay (the time taken to download two seconds of video)
- The TCP connection time
- The number of stalls and their duration (this is only applicable if the test is not

running in the 'bitrate reliably streamed' mode)

- The downstream throughput achieved

### Hulu (TCP)

Hulu (TCP)

Supported clients: Whiteboxes, Routers

The Hulu test is another application specific test that makes use of a third parties live infrastructure, ensuring that the test is mimicking the real behavior of applications. The streams a reference video from Hulu's CDN and measures key metrics related to its performance. SamKnows has partnered with Hulu to facilitate these measurements; Hulu has placed a reference video on its infrastructure and provided SamKnows with the ability to stream this video from any source.

At the time of writing, Hulu makes use of two CDNs – Level3 and Edgecast. The test uses these same CDNs and will randomly choose between one of these two CDNs are startup. The test makes use the same DNS names that the live Hulu service uses, ensuring that any CDN redirection logic put in place by the CDN operator, Hulu or the ISP is followed, just like a real client would use.

Hulu have placed a reference video on their two CDNs, transcoded at the same bitrates that Hulu clients typically use. This video is encoded using H264 at the following bitrates: 64kbps, 200kbps, 400kbps, 650kbps, 1Mbps, 1.5Mbps, 2Mbps and 3.2Mbps. The URL of the video encoded at each bitrate, on each CDN, is hard-coded into the Hulu test client.

At startup, the test randomly chooses the CDN to use, and begins to fetch video content using the list of URLs stored in the client. The test parses video frames as it goes, capturing the timestamp contained within each frame. After each frame we sample how much real-time has elapsed versus video time. If video time > real-time at a sample period, then an underrun has not occurred. Otherwise, one has occurred.

As with the other video tests, the client is configured to start testing at the highest supportable bitrate and then step down if and when stalls occur. This allows us to identify the 'bitrate reliably streamed'. Optionally the test may be instructed to continue at a fixed bitrate and simply record the number and duration of underrun events.

The test is capable of working with SD, HD and UHD streams.

The key outputs from this metric are:

- The bitrate reliably streamed
- The startup delay (the time taken to download two seconds of video)
- The TCP connection time
- The number of stalls and their duration (this is only applicable if the test is not running in the 'bitrate reliably streamed' mode)
- The downstream throughput achieved

#### Multicast IPTV (UDP)

Supported clients: Whiteboxes, Routers

This test measures the performance of multicast IPTV carried over RTP, which is commonly used by ISPs worldwide. Four aspects of the multicast delivery mechanism are measured:

- The success and performance of IGMP join/leave instructions (analogous to switching channels)
- The success and performance of retransmission requests
- The jitter observed in the multicast stream
- The level of packet loss on the multicast channel
- Consecutive packet loss events on the channel, including the number of packets and their duration

The test client connects to a multicast test channel as if it were a normal client. Once it has successfully joined the channel it receives the streams for a fixed duration and uses the received stream to calculate the metrics above.