**Building fast.com**

[[](https://netflixtechblog.medium.com/?source=post_page-----4857fe0f8adb--------------------------------)](https://netflixtechblog.medium.com/?source=post_page-----4857fe0f8adb--------------------------------)

[[Netflix TechBlog](https://netflixtechblog.com/?source=post_page-----4857fe0f8adb--------------------------------)](https://netflixtechblog.com/?source=post_page-----4857fe0f8adb--------------------------------)

[Netflix Technology Blog](https://netflixtechblog.medium.com/?source=post_page-----4857fe0f8adb--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fc3aeaf49d8a4&operation=register&redirect=https%3A%2F%2Fnetflixtechblog.com%2Fbuilding-fast-com-4857fe0f8adb&user=Netflix+Technology+Blog&userId=c3aeaf49d8a4&source=post_page-c3aeaf49d8a4----4857fe0f8adb---------------------post_header-----------)

Published in

[Netflix TechBlog](https://netflixtechblog.com/?source=post_page-----4857fe0f8adb--------------------------------)

·

7 min read

·

Aug 9, 2016

On [our company blog in May](https://media.netflix.com/en/company-blog/now-available-globally-fast-com-a-new-tool-to-check-your-internet-speed), we introduced [fast.com](http://fast.com/), our new internet speed test. The idea behind fast.com is to provide a quick and simple way for any internet user to test their current internet speed, whether they are a Netflix member or not. Since fast.com was released, millions of internet users around the world have run the test. We have seen a lot of interest in the site and questions about how it works. This blog will give a high-level overview of how we handled some of the [challenges inherent with measuring internet speeds](http://groups.csail.mit.edu/ana/Publications/Understanding_broadband_speed_measurements_bauer_clark_lehr_TPRC_2010.pdf) and the technology behind fast.com.

But first, some news — we are happy to announce a new FAST mobile app, available now for Android or Apple mobile devices. Get the free app from the [Apple App Store](http://apple.co/2aGvpAY) or [Google Play](http://bit.ly/2aHvytW).

**Design goals**

When designing the user experience for the fast.com application, we had several important goals in mind:

* Provide accurate, consistent results that reflect users’ real-life internet use case
* Load and run as quickly as possible
* Provide simple results that are easy to understand
* Work on most devices from the browser without requiring installation of a separate application

We wanted to make sure that fast.com could be easily used and understood by the majority of internet users, without requiring them to have any prior knowledge of computer networking, command line tools, and the like.

**Technical goals**

There are various ways to go about measuring internet speed and many variables that can impact any given measurement, some of which are not under our control. For example — configuration of the user’s local or home network, device or router performance, other users on the network, TCP or network configuration on the device. However, we thought carefully about the variables that are under our control and how they would further our overall goal of a simple but meaningful test.

Variables that are under our control, and which can influence the results of the test, include things like:

* Server location
* Load on the server
* Number of TCP connections used
* Size and type of download content used
* Methodology used to aggregate measurements

One major advantage we have is our [Open Connect](https://openconnect.netflix.com/) CDN, a globally-distributed network of servers (Open Connect Appliances or OCAs) that store and serve Netflix content to our members — representing as much as 35% of last-mile internet peak traffic in some regions. Using our own production servers to test internet speed helps to ensure that the test is a good representation of the performance that can be achieved during a real-life user scenario.

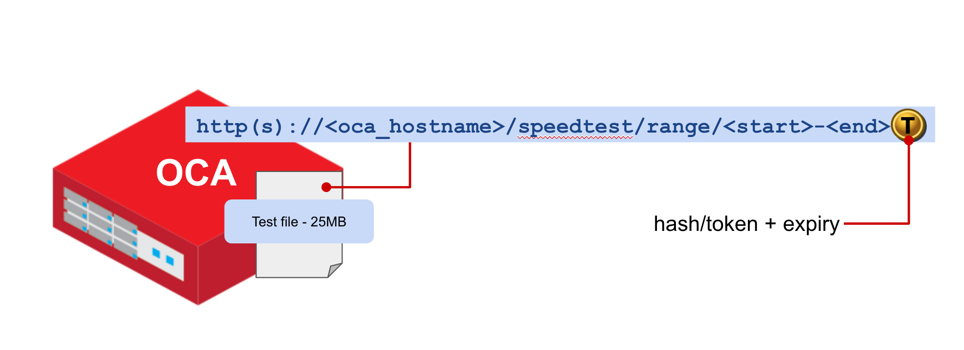
In pursuit of the design goal of simplicity, we deliberately chose to measure only download speed, measuring how fast data travels from server to consumer when they are performing activities such as viewing web pages or streaming video. Downloads represent the majority of activity for most internet consumers.

We also decided on the following high-level technical approaches:

* To open several connections for the test, varying the number depending on the network conditions
* To run the test on several of our wide network of Netflix production OCAs, but only on servers that have enough capacity to serve test traffic while simultaneously operating within acceptable parameters to deliver optimal video quality to members
* To measure long running sessions — eliminating connection setup and ramp up time and short term variability from the result
* To dynamically determine when to end the test so that the final results are quick, stable, and accurate
* To run the test using HTTPS, supporting IPv4 and IPv6

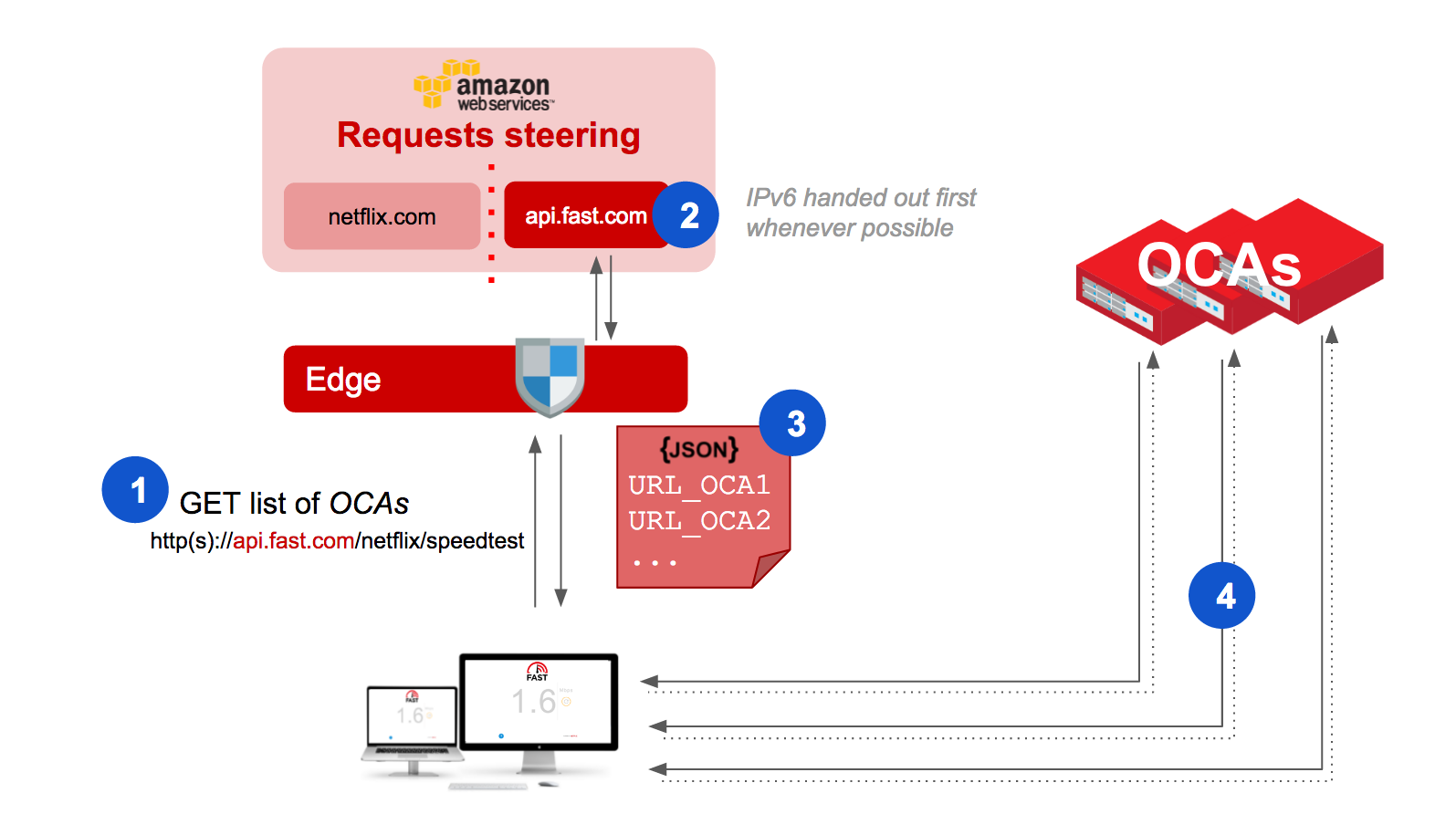
**Architecture**

As mentioned above, fast.com downloads test files from our distributed network of Open Connect Appliances (OCAs). Each OCA server provides an endpoint with a 25MB video file. The endpoint supports a range parameter that allows requests for between a 1 byte to a 25MB chunk of content.



In order to steer a user to an OCA server, fast.com provides an endpoint that returns a list of several URLs for different OCAs that are best suited to run the test. To determine the list, the endpoint uses logic that is similar to the logic that is used to steer netflix.com video delivery. The OCAs that are returned are chosen based on:

* Network distance
* Traffic load for each OCA, which indicates overall server health
* Network structure — each OCA in the list belongs to a different cluster



As soon as the fast.com client receives the URLs, the test begins to run.

**Estimating network speed**

The test engine uses heuristics to:

* Strip off measurements that are collected during connection setup/ramp up
* Aggregate the rest of the collected measurements
* Decide how many parallel connections to use during the test
* Try to separate processing overhead from network time — because fast.com runs in the browser, it has limited visibility into timing of network events like DNS resolution time, processing of packets on the client side and latency to test server
* Make a decision about when the client has collected enough measurements to confidently present the final network speed estimate

We exclude initial connection ramp up, but we do take into account any performance drops during the test. Network performance drops might indicate a lossy network, congested link, or faulty router — therefore, excluding these drops from the test result would not correctly reflect issues experienced by users while they are consuming content from the internet.

**Number of connections**

Depending on network throughput, the fast.com client runs the test using a variable number of parallel connections. For low throughput networks, running more connections might result in each connection competing for very limited bandwidth, causing more timeouts and resulting in a longer and less accurate test.

When the bandwidth is high enough, however, running more parallel connections helps to saturate the network link faster and reduce test time. For very high throughput connections, especially in situations with higher latency, one connection and a 25MB file might not be enough to reach maximum speeds, so multiple connections are necessary.

**Size of downloads**

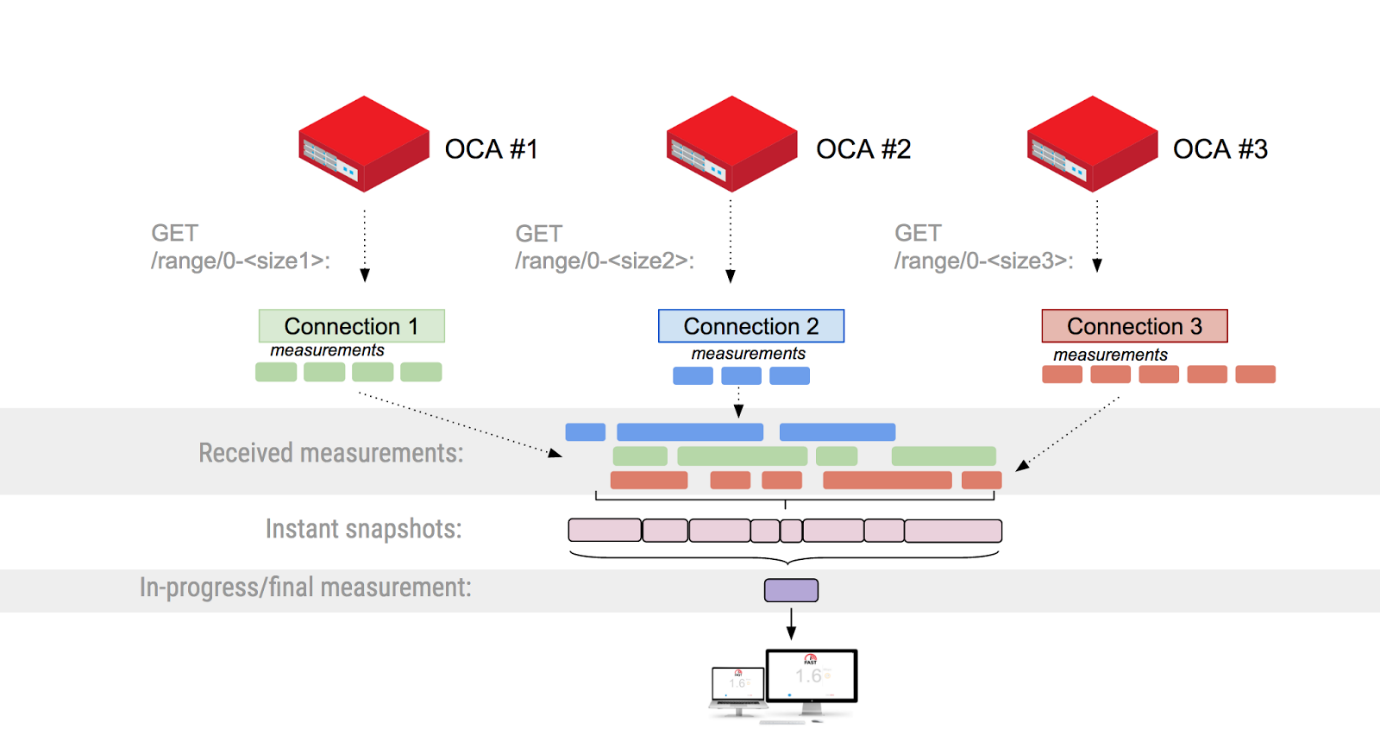
For each connection, the fast.com client selects the size of the chunk of the 25MB file that it wants to download. In situations where the network layer supports periodical progress events, it makes sense to request the whole file and estimate network speed using download progress counters. In cases where the download progress event is not available, the client will gradually increase payload size during the test to perform multiple downloads and get a sufficient number of samples.

**Computing the results**

After the download measurements are collected, the client combines the downloaded content across all connections and keeps the snapshot speed.

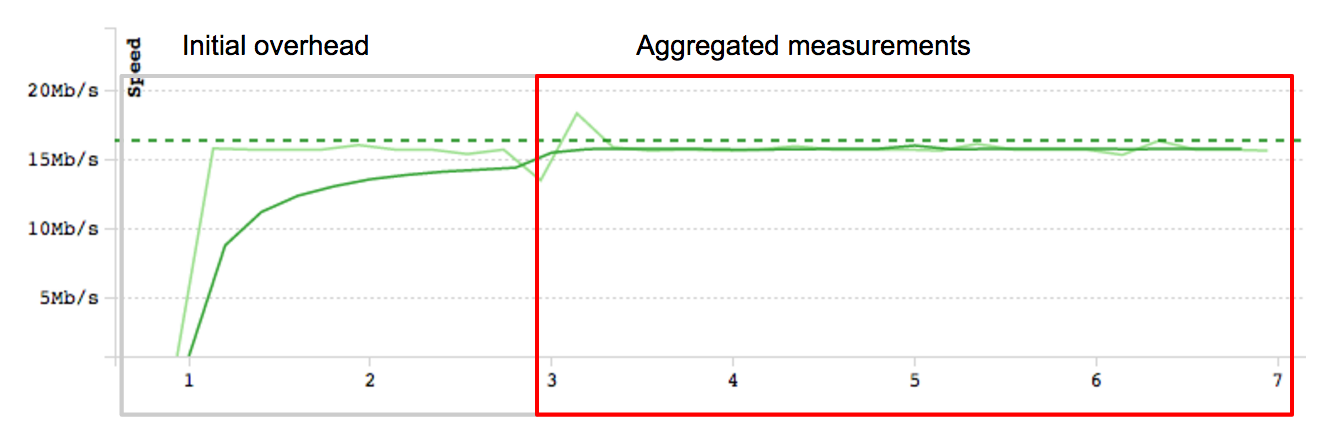
The ‘instant’ network measurements are then passed to the results aggregation module. The aggregation module makes sure that:

* We exclude initial connection ramp up
* We take the rest and compute rolling average of the other measurements

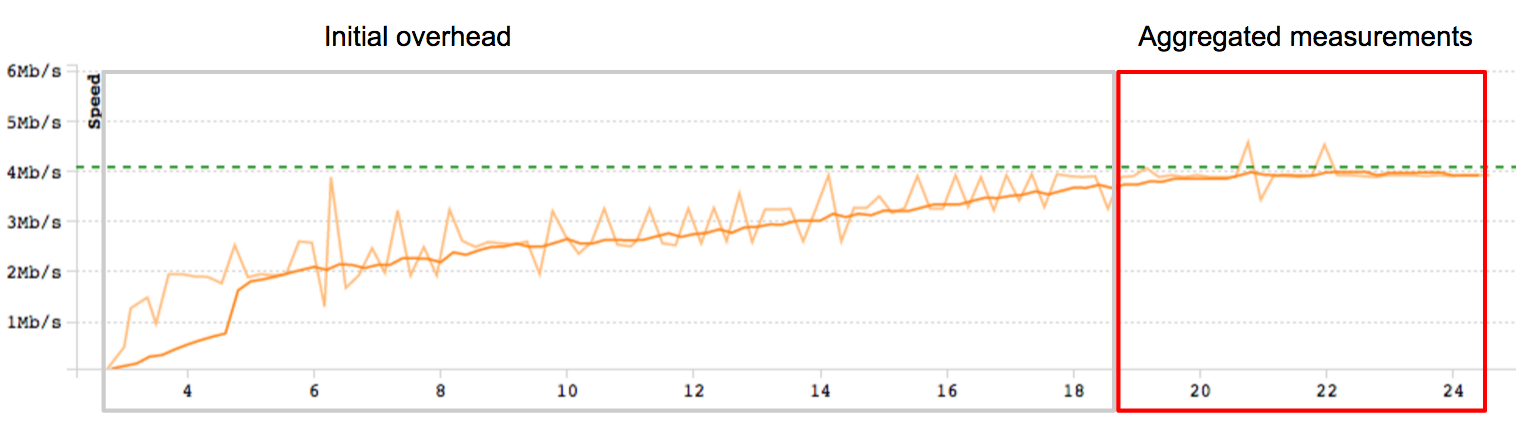


One of the primary challenges for the fast.com client is determining when the estimated speed measurements are ready to be presented as a final estimate. Due to the various environments and conditions that the fast.com test can be run under, the test duration needs to be dynamic.

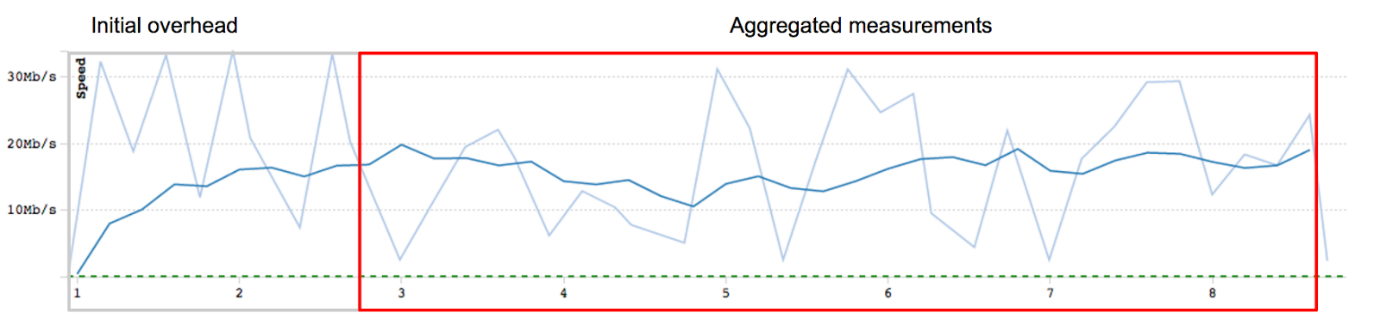
For stable low latency connections, we quickly see growth to full network speeds:



Higher latency connections take much longer to ramp up to full network speed:



Lossy or congested connections show significant variations in instant speed, but these instant variations get smoothed out over time. It is also harder to correctly identify the moment when connections have ramped up to full speed.



In all cases, after initial ramp up measurements are excluded, the ‘stop’ detection module monitors how the aggregated network speed is changing and makes a decision about whether the estimate is stable or if more time is needed for the test. After the results are stable, they are presented as a final estimate to the user.

**Conclusion and Next Steps**

We continue to monitor, test, and perfect fast.com, always with the goal of giving consumers the simplest and most accurate tool possible to measure their current internet performance. We plan to share updates and more details about this exciting tool in future posts.

*— by Sergey Fedorov and Ellen Livengood*

*Originally published at* [*techblog.netflix.com*](http://techblog.netflix.com/2016/08/building-fastcom.html) *on August 9, 2016.*