

Computation of broadband performance metrics using M-Lab data

author: Tiziana Refice
last updated: 10/03/2011

The [visualizations](#) on the M-Lab web site present several metrics that aim to estimate some characteristics of broadband performance, as measured by [NDT](#). A detailed description of the data collected by NDT is available [here](#).

The visualizations currently include the following metrics

- [Download and upload throughput](#)
- [Round Trip Time](#)
- [Network-limited and client-limited time ratio](#)
- [Weighted receiver window scale](#)

This document describes how these metrics are computed and the SQL queries used to compute them on [BigQuery](#).

Note: to compute all the following metrics, we exclude the measurement data collected by **incomplete tests** (i.e., tests that lasted less **10 seconds** and/or exchanged less than **8192 bytes** of data).

Download and Upload Throughput

An NDT test consists of a server-to-client test and a client-to-server test.

In order to compute download (upload) throughput values, all the server-to-client (client-to-server) tests run by the **same client**¹ during the **same month** are grouped together. For each server-to-client (client-to-server) group, we compute the **maximum** download (upload) throughput, where

- download throughput = $\text{HCThruOctetsAcked}^2 / \text{Duration}$
- upload throughput = $\text{HCThruOctetsReceived} / \text{Duration}$

The complete BigQuery query to compute download throughput values for a specific month is:

```
SELECT
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
```

¹A client is identified by its IP address.

²`HCThruOctetsAcked`, `HCThruOctetsReceived` and `Duration` are web100 variables. Namely, (i) `HCThruOctetsAcked` represents the total number of bytes sent by an M-Lab server during a test, (ii) `HCThruOctetsReceived` represents the total number of bytes received by an M-Lab server during a test, and (iii) `Duration` represents the duration of a test. See <http://code.google.com/apis/bigquery/docs/tcp-kis.txt> for a full description of all the web100 variables.

```

MAX(web100_log_entry.snap.HCThruOctetsAcked/
    (web100_log_entry.snap.SndLimTimeRwin +
     web100_log_entry.snap.SndLimTimeCwnd +
     web100_log_entry.snap.SndLimTimeSnd))
FROM [table_name]
WHERE
  IS_EXPLICITLY_DEFINED(project)
  AND project = 0
  AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
  AND connection_spec.data_direction = 1
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
  AND web100_log_entry.is_last_entry = True
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsAcked)
  AND web100_log_entry.snap.HCThruOctetsAcked >= 8192
  AND web100_log_entry.snap.HCThruOctetsAcked < 1000000000
  AND (web100_log_entry.snap.SndLimTimeRwin +
       web100_log_entry.snap.SndLimTimeCwnd +
       web100_log_entry.snap.SndLimTimeSnd) >= 9000000
  AND (web100_log_entry.snap.SndLimTimeRwin +
       web100_log_entry.snap.SndLimTimeCwnd +
       web100_log_entry.snap.SndLimTimeSnd) < 3600000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.CongSignals)
  AND web100_log_entry.snap.CongSignals > 0
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip

```

The complete BigQuery query to compute upload throughput values for a specific month is:

```

SELECT
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
MAX(web100_log_entry.snap.HCThruOctetsReceived/
web100_log_entry.snap.Duration)
FROM [table_name]
WHERE
  IS_EXPLICITLY_DEFINED(project)
  AND project = 0
  AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
  AND connection_spec.data_direction = 0
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
  AND web100_log_entry.is_last_entry = True
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsReceived)
  AND web100_log_entry.snap.HCThruOctetsReceived >= 8192
  AND web100_log_entry.snap.HCThruOctetsReceived < 1000000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.Duration)
  AND web100_log_entry.snap.Duration >= 9000000
  AND web100_log_entry.snap.Duration < 3600000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.log_time)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
  web100_log_entry.connection_spec.remote_ip,

```

```
web100_log_entry.connection_spec.local_ip
```

Different charts show throughput values at different levels of granularity (worldwide, country-level, region-level, city-level).

At each location level (city/region/country/world), all the max download (upload) throughput values collected during the **same month** and by clients geolocated in the **same location** are grouped together. For each group, we compute the **median** value.

We don't visualize the locations with less than **200 distinct clients**.

Note that the median within a region is computed as median of all the values in that region and it's **not** computed as the median of the medians in the cities of that region. The same applies to the country-level and world-wide aggregates.

Round Trip Time

In testing network and receiver limitations, NDT attempts to create congestion between a user's machine and the M-Lab server and, as a consequence, RTT values measured during the congestion period of an NDT test do not provide a good estimate of the server-client latency. Therefore, we estimate the RTT of a client during a specific month as the minimum RTT of all the tests run by that client during that month.

Specifically, in order to compute RTT values, all the server-to-client tests run by the **same client**³ during the **same month** are grouped together. For each group, we compute the **minimum** RTT, where

- $RTT = \text{MinRTT}$ ⁴

Using **MinRTT** has the drawback that it might underestimate the connection RTT, because it might be measured on the SYN ACK exchange, or some other tiny transaction which for low speed links does not represent the typical RTT for a full sized data segment. However, other web100 variables don't provide more accurate estimates.

In the RTT computation, we exclude

- Incomplete tests, as defined at the beginning of the document.
- Tests with **CountRTT**⁵ ≤ 10 because they don't have enough samples to accurately estimate the RTT.
- client-to-server tests. Each NDT test consists of a client-to-server test and a server-to-client test. In the RTT computation, we only take into account the server-to-client tests, because the server updates **MinRTT** and **CountRTT** only when it receives an acknowledgement. In client-to-server tests, the server receives an ack only during the 3 way handshake phase.

The complete BigQuery query to compute RTT values for a specific month is:

```
SELECT  
  web100_log_entry.connection_spec.remote_ip,
```

³A client is identified by its IP address.

⁴**MinRTT** is a web100 variable that represents the minimum RTT measured during a test.

⁵**CountRTT** is a web100 variable that represents the number of round trip time sampled during the test.

```

    web100_log_entry.connection_spec.local_ip,
    MIN(web100_log_entry.snap.MinRTT)
FROM [table_name]
WHERE
    IS_EXPLICITLY_DEFINED(project)
    AND project = 0
    AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
    AND connection_spec.data_direction = 1
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
    AND web100_log_entry.is_last_entry = True
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsAcked)
    AND web100_log_entry.snap.HCThruOctetsAcked >= 8192
    AND web100_log_entry.snap.HCThruOctetsAcked < 1000000000
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeRwin)
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeCwnd)
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeSnd)
    AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) >= 9000000
    AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) < 3600000000
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.MinRTT)
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.CountRTT)
    AND web100_log_entry.snap.CountRTT > 10
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.log_time)
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
    AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
    web100_log_entry.connection_spec.remote_ip,
    web100_log_entry.connection_spec.local_ip

```

The metric values are aggregated at city-level, region-level, country-level, and worldwide as described in the [throughput section](#).

Network-limited and client-limited time ratio

In testing network and receiver limitations, NDT attempts to create congestion between a user's machine and the M-Lab server. As a consequence, while running, a test can be in 3 states: **network-limited**, **receiver-limited**, and **server-limited**. By design ttests are never server-limited on the M-Lab platform, because M-Lab servers are specially provisioned. Therefore, each test spends some ratio of the test time in network-limited state and the rest of the time in client-limited state.

In particular, a test is in network-limited state when the throughput is limited by congestion in the network. On the other hand, a test is in receiver-limited state when the user's device limits the throughput (for example, due to configuration errors).

In order to compute network-limited (client-limited) time ratio values, all the server-to-client tests run by the **same client**⁶ during the **same month** are grouped together. For each group, we compute the **minimum** network-limited (client-limited) time ratio, where

⁶A client is identified by its IP address.

- network-limited time ratio = $\text{SndLimTimeCwnd}^7 / (\text{SndLimTimeRwin} + \text{SndLimTimeCwnd} + \text{SndLimTimeSnd})$
- client-limited time ratio = $\text{SndLimTimeRwin} / (\text{SndLimTimeRwin} + \text{SndLimTimeCwnd} + \text{SndLimTimeSnd})$

The complete BigQuery query to compute the network-limited time ratio values for a specific month is:

```
SELECT
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
  MIN(web100_log_entry.snap.SndLimTimeCwnd /
    (web100_log_entry.snap.SndLimTimeRwin +
     web100_log_entry.snap.SndLimTimeCwnd +
     web100_log_entry.snap.SndLimTimeSnd))
FROM [table_name]
WHERE
  IS_EXPLICITLY_DEFINED(project)
  AND project = 0
  AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
  AND connection_spec.data_direction = 1
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
  AND web100_log_entry.is_last_entry = True
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsAcked)
  AND web100_log_entry.snap.HCThruOctetsAcked >= 8192
  AND web100_log_entry.snap.HCThruOctetsAcked < 1000000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeRwin)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeCwnd)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeSnd)
  AND (web100_log_entry.snap.SndLimTimeRwin +
    web100_log_entry.snap.SndLimTimeCwnd +
    web100_log_entry.snap.SndLimTimeSnd) >= 9000000
  AND (web100_log_entry.snap.SndLimTimeRwin +
    web100_log_entry.snap.SndLimTimeCwnd +
    web100_log_entry.snap.SndLimTimeSnd) < 3600000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.log_time)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip
```

The complete BigQuery query to compute the client-limited time ratio values for a specific month is:

```
SELECT
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
  MIN(web100_log_entry.snap.SndLimTimeRwin /
    (web100_log_entry.snap.SndLimTimeRwin +
```

⁷ SndLimTimeCwnd , SndLimTimeRwin and SndLimTimeSnd are web100 variables. Namely, (i) SndLimTimeCwnd represents the cumulative time spent in the "Congestion Limited" state (including retransmission timeouts), (ii) SndLimTimeRwin represents the cumulative time spent in the "Receiver Limited" state, and (iii) SndLimTimeSnd represents the cumulative time spent in the "Sender Limited" state.

```

        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd))
FROM [table_name]
WHERE
  IS_EXPLICITLY_DEFINED(project)
  AND project = 0
  AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
  AND connection_spec.data_direction = 1
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
  AND web100_log_entry.is_last_entry = True
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsAcked)
  AND web100_log_entry.snap.HCThruOctetsAcked >= 8192
  AND web100_log_entry.snap.HCThruOctetsAcked < 1000000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeRwin)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeCwnd)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeSnd)
  AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) >= 9000000
  AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) < 3600000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.log_time)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip

```

The metric values are aggregated at city-level, region-level, country-level, and worldwide as described in the [throughput section](#).

Weighted receiver window scale

TCP allows to increase the receiver window size by using a scale factor called **receiver window scale**, which can be negotiated at the beginning of every TCP connection.

In order to compute receiver window scale values, all the server-to-client tests run by the **same client**⁸ during the **same month** are grouped together. For each group, we compute the **minimum** receiver window scale value, where

- receiver window scale = `WinScaleRcvd`⁹

The complete BigQuery query to compute receiver window scale values for a specific month is:

```

SELECT
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip,
  MIN(web100_log_entry.snap.WinScaleRcvd)

```

⁸A client is identified by its IP address.

⁹`WinScaleRcvd` is a web100 variable that represents value of the received window scale option if one was received; otherwise, a value of -1.

```

FROM [table_name]
WHERE
  IS_EXPLICITLY_DEFINED(project)
  AND project = 0
AND IS_EXPLICITLY_DEFINED(connection_spec.data_direction)
  AND connection_spec.data_direction = 1
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.is_last_entry)
  AND web100_log_entry.is_last_entry = True
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.HCThruOctetsAcked)
  AND web100_log_entry.snap.HCThruOctetsAcked >= 8192
  AND web100_log_entry.snap.HCThruOctetsAcked < 1000000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeRwin)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeCwnd)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.snap.SndLimTimeSnd)
  AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) >= 9000000
  AND (web100_log_entry.snap.SndLimTimeRwin +
        web100_log_entry.snap.SndLimTimeCwnd +
        web100_log_entry.snap.SndLimTimeSnd) < 3600000000
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.log_time)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.remote_ip)
  AND IS_EXPLICITLY_DEFINED(web100_log_entry.connection_spec.local_ip)
GROUP BY
  web100_log_entry.connection_spec.remote_ip,
  web100_log_entry.connection_spec.local_ip

```

The metric values are aggregated at city-level, region-level, country-level, and worldwide as described in the [throughput section](#), except that the values are aggregated by computing **averages** (instead of medians) at each level.