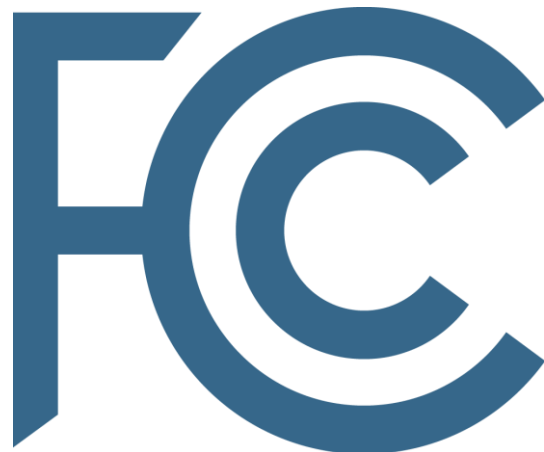# Measuring Broadband America Program

2016 Mobile Measurement
Open Data Release
Technical Description

Office of Engineering and Technology and

Office of Strategic Planning and Policy Analysis

Contents

LIST OF TABLES AND FIGURES

I.   Introduction

This Technical Description to the Measuring Broadband America Program ("MBA Program") 2016 Mobile Measurement Open Data Release provides detailed background technical information on the process by which the Mobile MBA Program's FCC Speed Test App ("the App" or "client")[1] collects crowd-sourced data from volunteers' mobile broadband Internet access services (BIAS). This Description also describes analysis methodology and reporting techniques relevant for Mobile MBA Program 2016 Open Data Release.

II.   Broadband Performance Testing Methodology

This section describes the technical features of the supporting system architecture and the FCC Speed Test App, as well as other technical aspects of the methods employed to measure broadband performance during this study.

A. FCC Speed Test Methodology Overview

Participants

The Measuring Broadband America program technologies and methodologies are developed collaboratively with a technical solutions contractor, broadband providers, academic researchers, independent researchers, consultants, and consumer organizations. In both open meetings with all parties, and more directed meetings with specific parties, feedback is collected and proposed changes to the measurement methodology are periodically discussed. The contractor supports the FCC's efforts to develop and maintain technical, policy and legal exchange among interested stakeholders and the FCC. This collaboration of stakeholders provides important feedback to help the Program evolve based on changes in broadband technology, the evolving marketplace, and consumer behavior.[2]

Measurement Process

The measurements that provide the underlying data rely both on the measurement client and measurement servers. The measurement client consists of the FCC Speed Test App that has been

---

[1] The FCC Speed Test App is available for both Android devices from Google Play, https://play.google.com/store/apps/details?id=com.samknows.fcc&hl=en, and for iOS devices from the Apple App Store, https://itunes.apple.com/us/app/fcc-speed-test/id794322383?mt=8.

[2] Past collaborative meeting attendees are listed in ex parte filings for meetings in the General Docket No. 12-264 , and for prior meeting beginning in 2010 in CG Docket No. 09-158, CC Docket No. 98-170, and WC Docket No. 04-36.

downloaded by hundreds of thousands of Android and iPhone users in locations across the nation.[3] Measurements are collected not only for the four largest mobile broadband providers, but for all mobile broadband providers subscribed to by consumers running the App.

Currently, the measurement servers are hosted by Level 3 Communications, and are located in nine cities across the United States near a point of interconnection between the broadband provider's network and the network on which the measurement server resides.

This methodology focuses on the performance of each mobile broadband provider's network. The metrics are derived from traffic flowing between a measurement client (located within the FCC Speed Test App) and a measurement server. The tests use the measurement server for which the latency between the measurement client and server is the lowest value. As a result, the metrics measure performance along a specific path within each mobile broadband provider's network, through a point of interconnection between the mobile broadband provider's network and the network on which the chosen measurement server resides.

However, the service performance that a consumer experiences in practice may differ from our measured values for several reasons. Wireless performance naturally varies with the communications technology used by the consumer's device, with the strength of the signal between the consumer's device and the mobile broadband provider's base station through which the consumer is communicating, with the aggregate Internet usage by all subscribers to that same mobile broadband provider that are currently using the same base station, with the amount of spectrum available to the mobile broadband provider, and with the make and model of the consumer's device. In addition, as noted, this App measures performance metrics based on a specific path from the user device to a chosen test server. On balance and in aggregate, this is a sound approach and is a common method to measure network speeds. However, specific paths or interconnection points within a broadband provider's network may be congested at times, which can affect a specific consumer's service. In addition, congestion beyond a broadband provider's network, not measured in our study, can affect the overall performance a consumer sees in their service.

Measurement Tests and Performance Metrics

The App performs the following active tests of mobile broadband performance:

- Download speed: Measures the average download speed in megabits per second over a maximum 15-second time interval.

- Upload speed: Measures the average upload speed in megabits per second over a maximum 15-second time interval.

- Latency: Measures the average round-trip time in milliseconds of up to 60 UDP data packets that are acknowledged as received individually within 2 seconds or are recorded as lost. The packets are sent over a maximum 30-second time interval.

---

[3] The Terms of Use and Privacy Policy for the MBA FCC Speed Test App describe the steps taken to protect each volunteer's privacy. See Mobile Broadband Performance Application Privacy Notice and Terms of Use, available at https://www.fcc.gov/general/mobile-broadband-performance-application-privacy-notice-and-terms-use. The Policy and technical measures of the test architecture, client and data processing prevent a precise identification of the number of handsets reporting results to the Measuring Broadband America Program.

- <u>Packet Loss</u>: The ratio of the number of UDP packets either not acknowledged by the measurement server, or acknowledged as received after 2 seconds, to the number of total packets sent from the client.

A more detailed discussion of the measurement test methodology is presented in Section C, below.

Availability of Data

The data set described in this technical description is available at http://www.fcc.gov/measuring-broadband-america. Previous MBA reports, along with the associated data, are also available at the same website.

Both the Commission and SamKnows, the Commission's contractor for this program, recognize that, while the methodology descriptions included in this document provide an overview of the project as a whole, there will be a number of interested parties—ranging from recognized experts to members of the general public—who would be willing to contribute to the project by reviewing the actual software used in the testing. SamKnows welcomes review of its software and technical platform, consistent with the Commission's goals of openness and transparency for this program and makes available the software source code for the FCC Speed Test App for Android and iPhone freely available under an Open Source License.[4]

The FCC takes strong measures to ensure the privacy and confidentiality of volunteers. All mobile data is collected without unique identifiers or information such as name, age, or other demographic information that could pose risks of identifying a particular volunteer. Consistent with the program's privacy policy, data that could potentially identify specific devices is processed in a way to minimize the risks to a subscribers' privacy interests. Sample analysis and processing to identify and mitigate risks is described in detail below.

B. Measurement and Testing Architecture

As illustrated below, the App executes tests as part of the measurement system that comprises a distributed network of user devices of the volunteer consumer panel, used to accurately measure the performance of broadband connections based on real-world usage. The App is controlled by a hosted test scheduler and reports measurement results to the reporting database, where the data is collated on the reporting platform. The App actively measures upload and download speeds, round-trip latency, and packet loss by exchanging information with a series of speed-test servers (measurement servers), which the App contacts according to the test schedule or when a user manually initiates a test. The measurement servers that are used for mobile broadband measurement are hosted by Level 3 Networks, and are distributed nationally to enable a client to select the closest host server to minimize latency.

---

[4] The software client source code for the FCC Speed Test App for Android and iPhone is available on GitHub in the following public repositories. Core and application UI repositories for Android and iPhone respectively at https://github.com/SamKnows/skandroid-core, https://github.com/SamKnows/skandroid-fcc; and https://github.com/SamKnows/skios-core, https://github.com/SamKnows/skios-fcc. Other software that was used for the testing is made available for researchers and the general public for noncommercial purposes. To apply for noncommercial review of the code, interested parties may contact SamKnows directly at team@samknows.com, with the subject heading "Academic Code Review."

Figure 1: Testing Architecture I
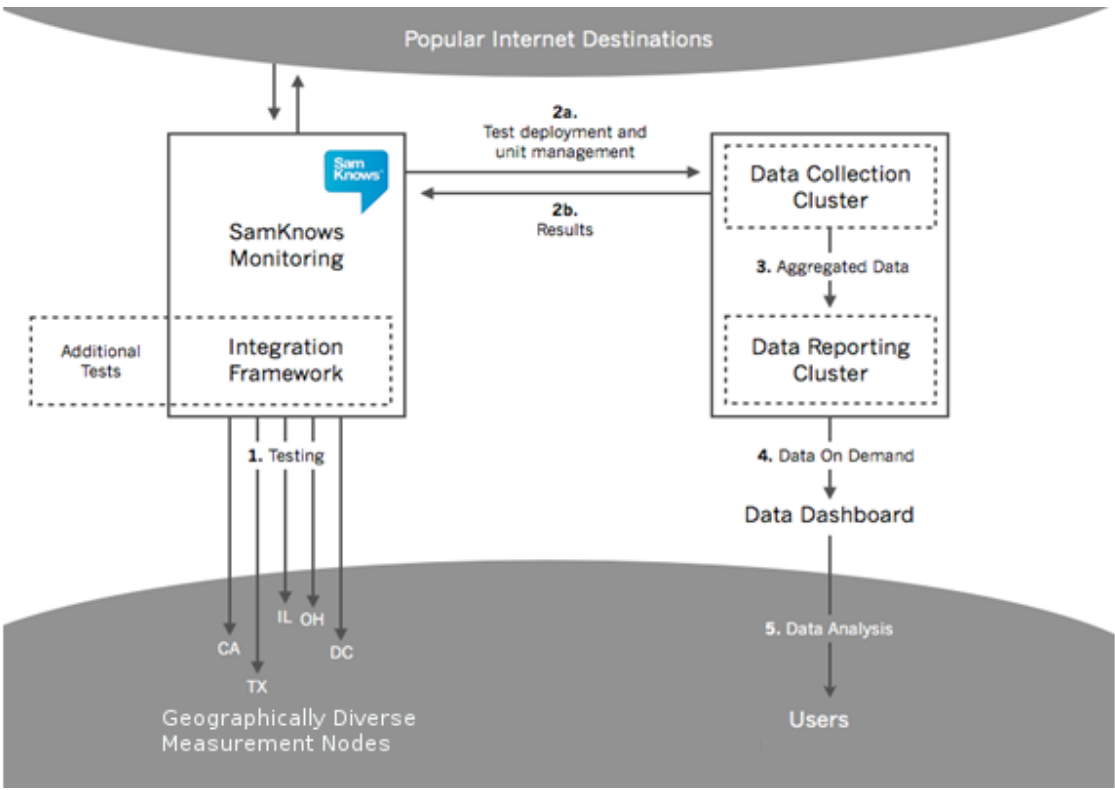
The Commission's MBA Program focuses on those elements of the Internet pathway managed by a service provider to enable a consumer's access to the Internet. The App resides on the user's device and tests these elements of the Internet path indicated in Figure 2 by the items 1-5, as the connection of the device to the provider's mobile network, through a carrier's managed networks, to a nearby major Internet exchange point where a measurement server is hosted. This focus aligns with the wireless broadband service advertised to consumers and allows a direct comparison across wireless broadband providers of actual performance delivered to the user. The App can perform tests using any connection available, including a device's WiFi connection, but only wireless information about a test's mobile cellular connection is included in the current dataset.

Figure 2: Testing Architecture II
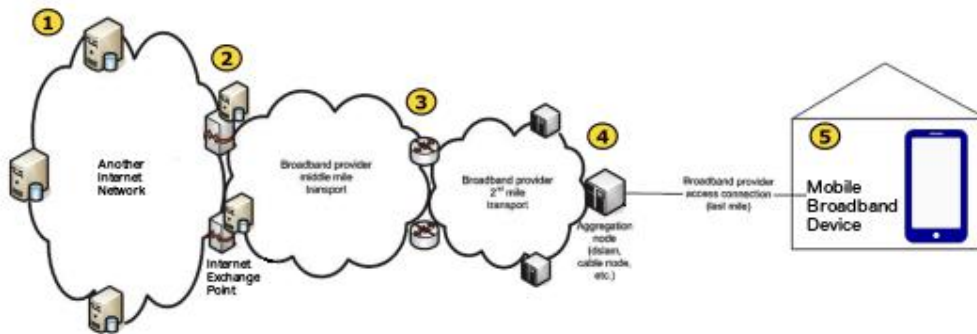
DEFINITIONS

① **Another Internet Network:** Internet content that is hosted by multiple service providers, content providers and other entities in a geographically diverse (worldwide) manner

② **Internet Exchange Point:** A point at which traffic is exchanged between the BIAS provider's network and another network

③ **Link between 2nd mile and middle mile:** broadband provider managed interconnection between middle and last mile

④ **Aggregation node:** First aggregation point for broadband provider

⑤ **Mobile Broadband Device:** User's smartphone, tablet or other mobile device used to connect broadband provider

C. TEST DESCRIPTIONS

The following sub-sections detail the methodologies used in the Download speed, Upload speed, Latency, and Packet Loss tests.

1. Download speed and upload speed

Two separate tests measure the download and upload speed in megabits per second by performing multi-connection TCP HTTP GET and POST requests to a target test node. [5] Binary non-zero content, herein referred to as the payload, is hosted on a web server on the target test node. Each test cycle begins with a maximum 5-second warmup, followed by the download or upload speed test. Each speed test concludes either after 20 MB of payload is transferred, or after a maximum elapsed time of 15 seconds. The total time for the test cycle is thus no more than 20 seconds, and the client will attempt to download as much of the payload as possible for the duration of the test. The payload and all other testing parameters are configurable and may be subject to change in the future.

Tests are executed using three concurrent TCP connections. Each connection used in the test counts the numbers of bytes of the target payload transferred between two points in time, and calculates the

---

[5] Note that round-trip latency and packet loss tests are executed using UDP packets.

speed of each thread as the number of bits transferred over the number of seconds in the active test window.

Factors such as TCP slow start and congestion are taken into account by repeatedly downloading small chunks (default 256 KB) of the target payload before the real testing begins. This "warm up" period will complete when three consecutive chunks are downloaded at the same speed or within a small tolerance (10% by default) of one another. Three individual connections are established, each on its own thread, and are confirmed as all having completed the warm up period before timing begins.

Content downloaded is copied to /dev/null or equivalent (i.e. it is discarded). Payload content for upload tests is generated and streamed on the fly from /dev/urandom (or equivalent).

The following is an example of the calculation performed for a multi-connection test using three concurrent connections.

- S = Speed (Bytes per second)
- B = Bytes (Bytes transferred)
- T = Time (Seconds) (between start time point and end time point)
- S1 = B1 / T1 (speed for Thread 1 calculation)
- S2 = B2 / T2 (speed for Thread 2 calculation)
- S3 = B3 / T3 (speed for Thread 3 calculation)
- Speed = S1 + S2 + S3
- Example values from a 3 MB payload:
- B1 = 3077360   T1 = 15.583963
- B2 = 2426200   T2 = 15.535768
- B3 = 2502120   T3 = 15.536826
- S1 = B1/T1 = 197469.668017
- S2 = B2/T2 = 156168.655454
- S3 = B3/T3 = 161044.475879
- S1 + S2 + S3 = Total Throughput of the line = 197469.668017 + 156168.655454 + 161044.475879 = 514682 (Bps) * 0.000008 = 4.12 Mbps

The following pseudo-code describes the algorithm of the test in more detail:

```
// Call the test
download_test( target, 8080, '/1000MB.bin', 5 sec, 2MB, 15 sec, 20 MB,
3 )


// Define the test
function download_test( target, port, uri, warmupmaxtime,
warmupmaxbytes, transfermaxtime, transfermaxbytes, numberofthreads ) {

   threads = array()
   for i=0; i<numberofthreads; i++ {

      t = new Thread( speedtest_thread( target, port, uri ) )
      t.start()
```

```
        threads.put(t)
    }


    warmup_complete = false
    warmup_bytes_transferred = 0
    warmup_time_elapsed = 0
    bytes_transferred = 0
    time_elapsed = 0


    while( 1 ) {

        // Get latest statistics for each thread
        foreach t in threads {

            // check to see if there's an error on the thread
            if t.status = ERROR
                fail and exit with error status

            // Check to see if the warmup was not complete but now is
            if (!warmup_complete) {
                warmup_bytes_transferred += t.bytes_transferred
                warmup_time_elapsed = now() - t.time_start
            }

            // Check to see if the test has completed
            if (warmup_complete) {
                bytes_transferred += t.bytes_transferred
                time_elapsed = now() - t.time_start
            }
        }

        // If warmup is now complete, mark it as such and begin testing
        if (!warmup_complete &&
            ( warmup_bytes_transferred > warmupmaxbytes ||
              warmup_time_elapsed > warmupmaxtime)
            ) {

        }

        // Check to see if we can finish
        if (warmup_complete) {

            // deduct warmup
            bytes_transferred -= warmup_bytes_transferred
            time_elapsed -= warmup_time_elapsed

            // Check to see if we're done
            if ( bytes_transferred > transfermaxbytes || time_elapsed >
transfermaxtime ) {

                break;
            }
```

```
        }
    }

    // calculate final result
    speed = bytes_transferred / time_elapsed
}



// Multiple threads of the below will be called
function speedtest_thread( target, port, uri ) {

    bytes_transferred = 0
    time_start = now()
    status = WAITING

    try{
        client = new http_client( target, port, uri )
        status = CONNECTED
        while( data = client.recv() ) {

            bytes_transferred += data.length
        }
    } catch {
        status = ERRROR
    }
}
```

2. Latency and Packet Loss

The UDP Latency and Packet Loss tests measure the round-trip time of UDP packets between the device and a target test site. Each packet consists of an 8-byte sequence number and an 8-byte timestamp. If an acknowledgement is not received back within two seconds of sending, the packet is treated as lost. The test sends up to 60 packets (datagrams) and records the number of packets sent, the average round-trip time of these and the total number of packets lost.

The following pseudo-code describes the algorithm of the test in more detail:

```
    call latencytest( server, 5000, 500ms, 2s, 60, 30s )


    function latencytest ( host, port, interPacketTime, delayTimeout,
    datagrams, maxTime ) {


    sent = 0

    received = 0
```

```
lost = 0
rttArray = array()
start = now()


sock = socket(host,port)
for i = 0; i < datagrams; i++ {

   t1 = now()
   send(sock, i)
   sent++

   try {
      recv(i, delayTimeout )
      t2 = now()
      rtt = t2 - t1
      insert( rttArray, rtt )
      received++

      // ensures minimum spacing between packets is interPacketTime
      sleep( max(0, interPacketTime - rtt) )

   } catch timeout {
      lost++
   }

   // exit if we've exceeded maximum time
   if start + maxTime > now() {
      break
   }


}


// calculate average RTT, all other variables
// are already available (sent, received, lost, etc)
```

```
avgRtt = avg(rttArray)
}
```

D. Testing Schedule

The FCC Speed Test App for Android and iOS both enable a user to initiate "manual" tests. In addition, and unlike many commercially available speed tests which are solely user-initiated, the FCC Speed Test App for Android also contains a default setting which causes the App to execute at automated pre-scheduled intervals.[6] The automated testing function can be disabled and the App can be configured to start a test only when manually executed. iOS devices do not have automated testing capability and can only execute the speed test manually. The scheduled testing approach minimizes extraneous factors that could degrade a statistically accurate measure of mobile broadband performance. Automatic background tests contribute valuable, high-quality data about mobile broadband performance.

The application will periodically download a predefined test schedule randomized to execute tests on the following criteria:

Table 1

Test	Target(s)	Testing Time Block	Duration	Est. Daily Volume
Download Speed	1 off-net measurement server test node	0700-0859 - 2 hours 0900-1659 - 8 hours 1700-1959 - 3 hours	20 MB fixed size, up to max 20 seconds[7]	60 MB max
Upload Speed	1 off-net measurement server test node	0700-0859 - 2 hours 0900-1659 - 8 hours 1700-1959 - 3 hours	20 MB fixed size, up to max 20 seconds	60 MB max
UDP Latency	1 off-net measurement server test node	0700-0859 - 2 hours 0900-1659 - 8 hours 1700-1959 - 3 hours	30 seconds	0.1 MB
UDP Packet Loss	1 off-net measurement server test node	0700-0859 - 2 hours 0900-1659 - 8 hours 1700-1959 - 3 hours	30 seconds	N/A (uses above)

E. Data Collection

The initial beta trials of the FCC Speed Test App and the analysis of the data collected by the various

---

[6] The Apple iOS does not support features necessary for scheduled test execution on the App for iPhone and thus App for iPhone only supports manual testing at this time.

[7] Note that TCP-based upload and download tests provide for a five-second warm up period before the 15 seconds for the test duration for a total of 20 seconds to complete a test.

beta versions of the App ensured that once the App was officially launched, all collected data met the FCC's technical privacy and engineering requirements. In addition to data collected from active download speed, upload speed, latency, and packet loss tests, data related to the radio characteristics of the device, information about the handset type and operating system version, the GPS coordinates of the handset at the time each test is run (if available), the date and time of the observation, and other metrics are recorded on the handset in JSON (JavaScript Object Notation) nested data elements within flat files. These JSON files are then transmitted to SamKnows storage servers at periodic intervals after the completion of active test measurements.[8]

Once the JSON files are received by SamKnows, they are converted into row results that are inserted into a series of SQL tables using software developed by SamKnows Ltd. Each collection of results from a "test" share a primary key, which can be combined with local timestamp or metric values to combine the active and passive metrics that are connected during a single test. The Data Dictionaries for the JSON and CSV formats for Android and iOS describe in detail the nested format of the radio, handset, active test result, and other data recorded on the handset prior to transmission to the storage infrastructure, as well as the converted CSV table structures and data elements can be found on the Measuring Broadband America website.[9]

III.   Data Processing of Test Results

This section describes the background for processing of data gathered for the 2016 Mobile MBA Program, and the methods used to collect and analyze the test results. Other supporting documentation can also be found on the Measuring Broadband America website.

A. Data Processing Temporal Features

One of the key factors that affect broadband performance is usage-based congestion. During peak use of a network there are more people attempting to use the wireless network simultaneously, giving rise to the potential for network congestion. When congestion occurs, users' broadband performance will suffer. In order to identify usage patterns, data are processed to include the hour in which a measurement occurred. The precise times collected at the time of the measurements could introduce privacy concerns, and thus dates and times of measurements are processed to remove the minutes and seconds from the date and time of measurements for inclusion as ISO 8601 formatted strings in the dataset.[10]

Where fewer than two samples exist in a given hour for the same model handset, location, and specific carrier information, the data are processed by removing features of the observations or increasing the

---

[8] The tests results are transferred depending on the available connectivity at the conclusion of the test and can be stored and forwarded when connectivity is immediately unavailable.

[9] https://www.fcc.gov/general/measuring-mobile-broadband-performance.

[10] All times are collected from calls to the device operating system and are originally recorded as Unix timestamps and a mix of ISO 8601 strings with and without corrections for the device's localtime time zone offsets.

level of aggregation of the information or otherwise redacted from the released dataset according to the privacy analysis and processing discussed below.

The speeds at which signals can traverse networks are limited at a fundamental level by the speed of light. While the speed of light is not believed to be a significant limitation in the context of the other technical factors addressed by the testing methodology, a delay of approximately 5 ms per 1,000 km of distance traveled can be attributed solely to the speed of light (assuming optical transmission media). The geographic distribution and the testing methodology's selection of the nearest test servers are believed to minimize any significant effect and can be identified in the result of a test. However, propagation delay is not explicitly identified in the results.

B. Data Processing

As described above, SamKnows collects source JSON files from volunteers. These files are then converted into a relational format, and saved as CSV files. This is done using the mmba_JSON_bulkimporter tool.[11] Finally, these CSV files are ingressed into MySQL database servers, which are managed by SamKnows. FCC staff ingressed these CSV files into Postgres databases with PostGIS extensions, as described in the Table 2 below.[12] Processing of data for the 2016 Mobile MBA Program focused on results in the following tables: curr_httpget, curr_httppost, curr_udplatency, curr_location, curr_networkdata, and curr_submission. SQL statements can be used to process results in the curr_location and curr_networkdata tables, and to produce references or variables used when processing results in the three active test tables (curr_httpget, curr_httppost, curr_udplatency). Scripts to process these tables are included on the Program's website.[13]

For example, results in the curr_networkdata table can be processed to produce 'carrier' or 'cell' variables using the following SQL statements:Carrier:

- alter table curr_httpget add column carrier text;
- update curr_networkdata set carrier = 'att' where Network_operator_code in ('310410') and network_operator_name ~* '^[\s]*at[\s]*[&]*[\s]*t[\s]*$' ;
- update curr_networkdata set carrier = 'sprint' where network_operator_code IN ('310120') and network_operator_name ~* 'sprint[\s]*$';
- update curr_networkdata set carrier = 'verizon' where network_operator_code IN ('310012', '311480') and network_operator_name ~* 'verizon';

---

[11] JSON to CSV conversion tool for FCC Speed Test JSON Result files written in PHP, https://github.com/FCC/mmba_JSON_bulkimporter.

[12] Early work was performed on versions 9.2 and 9.3 of the database but the bulk of the report processing was performed using Postgres 9.5 with extensive use of new aggregation functions for calculating percentiles, inline filters for selects and better ingressing features for \copy commands that eased the import of MySQL database extracts.

[13] See the data and supporting scripts on the Measuring Mobile Broadband Performance website at https://www.fcc.gov/general/measuring-mobile-broadband-performance.

- update curr_networkdata set carrier = 'tmobile' where network_operator_code IN ('310026', '310260') and network_operator_name ~* '^[\s]*?t[\-\s]*?mobile';
- update curr_httpget set carrier= b.carrier from curr_networkdata b where b.metric= 'httpget' and b.carrier is not null and curr_httpget.submission_id = b.submission_id;
- update curr_httppost set carrier= b.carrier from curr_networkdata b where b.metric= 'httppost' and b.carrier is not null and curr_httppost.submission_id = b.submission_id;
- update curr_udplatency set carrier= b.carrier from curr_networkdata b where b.metric= 'udplatency' and b.carrier is not null and curr_udplatency.submission_id = b.submission_id;

Cellular Test 'cell' variable:

- alter table curr_httpget add column cell boolean not null default FALSE;
- alter table curr_networkdata add column cell boolean not null default false;
- update curr_networkdata set cell = 'y' where active_network_type ~* 'mobile' or active_network_type ~* 'cell' or active_network_type ~* 'WIMAX' ;
- update curr_httpget set cell = 'T' from curr_networkdata b where b.metric= 'httpget' and b.cell = 'T' and curr_httpget.submission_id = b.submission_id;
- update curr_httppost set cell = 'T' from curr_networkdata b where b.metric= 'httppost' and b.cell = 'T' and curr_httppost.submission_id = b.submission_id;
- update curr_udplatency set cell = 'T' from curr_networkdata b where b.metric= 'udplatency' and b.cell = 'T' and curr_udplatency.submission_id = b.submission_id;

The variable 'network_type' , which defines the bearer channel, is located in the curr_networkdata table. This variable can be linked to the active test metric tables using the following statement for download tests, and using similar statements for upload tests and for the UDP tables:

- alter table curr_httpget add column lte boolean not null default FALSE;
- update curr_httpget set lte = 'Y' from curr_networkdata b where b.metric='httpget' and b.network_type ~* 'lte' and curr_httpget.submission_id = b.submission_id ;

The curr_location table can be linked to the data to produce the geocoded results used in the 2016 Mobile MBA Program. During the execution of each FCC Speed Test, the operating system of the device provides the current GPS latitude and longitude if available, although not every test measurement has a GPS location. Each GPS location observation was deleted and replaced with the cellular market area identity (CMA). GPS locations were processed to provide the CMA where tests were executed. No additional geographic information was processed for measurements that did not record a valid GPS

longitude and latitude.[14] The following SQL statements were processed in PostGIS to geocode locations to CMA:

- UPDATE curr_location SET geom = ST_SetSRID(ST_MakePoint(longitude,latitude),4326) where extract (year from localtime) = 2016 and geom is NULL;
- Update curr_location set cma_id = as c.id from cma_2010 c where curr_location.cma_id is not null and st_within (curr_location.geom, c.geom) ;

The scripts used to produce temporal and other cross-reference variables are available on the MBA Program website.[15]

C. Data Processing of Raw and Validated Data

The data collected and analyzed in the 2016 Mobile MBA Program is made available as OpenData for review and use by the public. Raw and processed data sets, testing software, and the methodologies used to process and analyze data are freely and publicly available. A developer FAQ is available online to assist with database configuration using mySQL and postgreSQL.[16]

The process flow below describes how the collected data was processed for analysis in the 2016 Mobile MBA Program. Researchers and developers are encouraged to review the flow chart and supporting files.

Table 2

Raw Data:	Raw data for the chosen period is collected from the measurement database. All data is imported into a database or other analytic environment.
Clean and Process Data:	Data is processed to identify samples used for analysis. In particular network operator names and codes and results from cellular modem connections are processed with easy to identify variables to flatten and minimize cross-table queries.
Statistical Processing:	Subsetted values are processed to produce statistical values to review features of the data useful for data exports for this release.
Excel Tables & Charts:	Summary data tables and charts are produced from targeted analysis in response to aspects of the privacy analysis.

---

[14] Staff did explore techniques to utilize base station and cell tower IDs to infer CMA's for some measurements sharing features with other measurements with valid GPS locations. These and other techniques are expected to improve the geographic resolution in the future.

[15] https://www.fcc.gov/general/measuring-mobile-broadband-performance.

[16] http://www.fcc.gov/measuring-broadband-america/2012/database-setup-and-importing-measuring-broadband-america-data-april-2012

D. Crowdsourced Volunteer Panel, Data Quality and Privacy Protection

The FCC Speed Test App is a smartphone application for Android and iOS devices which is available for download in the Google Play and Apple iTunes software marketplaces. Volunteers may choose to download the App and share information about their device and operating systems, as well as the download speed, upload speed, latency, and packet loss of their mobile broadband connection as measured during each test. The broadband performance data collected by the App is limited to information used to measure the mobile broadband service of the volunteer at the time a test is executed. The users' location and time when a test is run, and the IP address when data is uploaded to our servers, is collected and used in processing results. The App does not collect other personally identifiable information such as the name, phone number, or other identifiers associated with a device. The data are processed and aggregated to minimize risks to privacy.

This section includes the background of the study, a technical summary of the App, a discussion of the crowdsourced dataset that results from the volunteer panel, and a discussion of the ongoing management of the App which is necessary to maintain the statistical and operational goals of the MBA program. Our methodology is focused on the measurement of performance within the broadband provider's network as delivered to the consumer.

The 2016 Mobile MBA Program data release includes data that ranges from November 2013 to June 2016. The amount of data in our sample varies by geography, service provider, and technology. Technical privacy policies for the program are implemented by additional sample analysis and processing to identify sample data to aggregate results from a single device or a small and identifiable group of devices, in order to mitigate risks of identifying individual measurements. In addition to manual tests, the FCC Speed Test App for Android can perform scheduled tests at random times within pre-determined blocks of time. The randomized scheduling of these tests removes some of the spatial and temporal biases that are inherent in completely manual tests.in the publicly available data, detailed GPS coordinates were replaced by the CMA within which a given test was performed.

1. USE OF AN ALL VOLUNTEER, CROWDSOURCED PANEL

The FCC submitted an open call for voluntary participation in this measurement effort. Users were encouraged to familiarize themselves with the goal of the program, and to install and run tests using the App. Calls for recruitment and information regarding the study were predominantly disseminated in public meetings and through a website that explains the techniques used to measure mobile broadband performance and provide a means for volunteers to contribute measurement data.[17] As a result of the

---

[17] The Program and the App are described on the FCC Web Site at https://www.fcc.gov/general/measuring-mobile-broadband-performance and at SamKnows Ltd., the FCC's contractor supporting the program at https://www.measuringbroadbandamerica.com/.

open call, hundreds of thousands of volunteers across the country downloaded the FCC Speed Test App. Volunteers also provided valuable comments and feedback. The initial collected measurements were analyzed to produce statistically valid sets of volunteers based on mobile broadband provider, connection technology type, and region.

Because this is a crowdsourced data collection process, the FCC is able to collect data in both rural and urban areas, and across a wider geographic range, than would be possible using a drive test or a structured sampling scheme.

2. Other Client and Server Infrastructure Quality Measures:

The App utilizes a series of servers, software, and other measurement infrastructure to complete each download speed, upload speed, latency, and packet loss measurement in a manner that protects the quality of the data collected. The measurement infrastructure is continuously tested and monitored so that the functionality and operation of any individual component of the test infrastructure functions as defined and necessary. In particular, information about the environment of the device the App uses and the measurement servers both provide important checks on the quality of the data and are discussed below.

The App was designed to minimize burdens on the testing device and maximize the ability of the test to accurately measure the performance of the tested network connection. For that reason, the App avoids imposing high processing loads or high volumes of data usage in the testing and result delivery process. Prior to the deployment of the software, SamKnows provided test suites and results for all versions of the App to collaborative stakeholders in order to safeguard software compatibility, functionality, and operability on user devices expected to be used during the MBA Program.[18] Testing and validation further established the readiness of the measurement platform. Initial results of infrastructure use were correlated with the results from the preceding test as a measure of validating the correctness of operation of the system.

Each test records information about the client environment including checks to ensure tests do not execute if there is undue use of the network connectivity or processing power of the device. Understanding the use of the device's CPU resources and network capacity helps to ensure that when a test is run, the test results are accurately capturing the capability of the measured network connection, and not a secondary impact of the client's ability to efficiently process and transfer information to the measurement server.

SamKnows performs continuous monitoring and maintenance of the broadband measurement network and equipment used in the collection of mobile and fixed data to ensure data collection and measurement quality. These continuous efforts include monitoring, maintaining, and ensuring the health and quality of the measurement infrastructure and all data collected for reporting purposes; collecting and auditing data on the health and quality of the measurement infrastructure on a continual basis; and developing and maintaining all system and network administration staff and resources ensure

---

[18] This group included major broadband providers comprising roughly 90 percent of the U.S. consumer wireless broadband market, academics and independent researchers, consultants, and consumer organizations.

the quality of the collection process and address issues that may arise. SamKnows monitors the health of the server infrastructure using data collected on the measurement server connectivity statistics including intra-server speed and performance measurements; measurement server virtual and physical host process and utilization statistics; and other data supporting monitoring and auditing of health of the entire measurement server infrastructure as directed by the FCC. SamKnows makes available processed statistics presented in a "server dashboard" that is available for broadband providers participating in the MBA program.

Server infrastructure, including measurement servers provided by Level 3, used for mobile measurement are also utilized for fixed measurement. While not employed for collecting measurements used in reporting, SamKnows also manages a private server architecture as a continuous means to audit and diagnose data supporting the checking, maintenance and testing of any problems in the server architecture. While mobile measurements are not collected on every element of the MBA Program's infrastructure, the ability to identify trends in fixed statistics across all measurement infrastructure provide a further check on potential infrastructure problems that could influence mobile data collection efforts.

3. PROTECTION OF VOLUNTEERS' PRIVACY

The MBA Program is based on principles of openness, transparency, and commitment to releasing the Open Data used to produce each report coincident with the Report's release.[19] The trust that volunteers place in the MBA Program is critical to the mobile crowdsourcing approach and collaborative process. The privacy policies were collaboratively designed with participants from academia, industry and manufacturing, public policy and government, and wireless carriers. A technical model and privacy policy were adopted to balance measures to protect privacy for participants while maintaining the utility for important data and analysis for the public.[20] This effort focused the App's collection of information used to measure and describe subscribers' mobile broadband performance. Collaborative participants discussed the importance of collecting the location and time when a test is run[21] and determined that a

---

[19] Measuring Broadband America Policy on Openness and Transparency, available at
https://www.fcc.gov/general/measuring-broadband-america-policy-openness-and-transparency.

[20] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Nov. 30, 2012) available at
https://www.fcc.gov/ecfs/filing/6017145746/document/7022069986 (announcing the FCC would lead a review with staff from FCC and FTC over privacy policy, mobile terms and conditions, a Code of Conduct, specifications defining the nature of the data collected, aggregation policies, and consumer notification and data release policies). See Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Jan. 10, 2013) available at
https://www.fcc.gov/ecfs/filing/6017157489/; Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Jan. 17, 2013) available at https://www.fcc.gov/ecfs/filing/6017159364; Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Feb. 15, 2013) available at https://www.fcc.gov/ecfs/filing/6017165528; Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Feb. 25, 2013) available at https://www.fcc.gov/ecfs/filing/6017165532/document/7022125492. See also Mobile Collaboration Ex Parte Letter, GN No. 12-264 (Mar. 4, 2013) available at
https://www.fcc.gov/ecfs/filing/6017166912/document/7022127970 (finalizing the Terms and Conditions review draft).

[21] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 13, 2012) available at
https://www.fcc.gov/ecfs/filing/6017152922 (discussions that accessing a handset's GPS would be necessary to

post-collection review could allow for the greatest flexibility and relevance in releasing data. The approach permits the data to be released in a manner most responsive to the data that is actually collected and kinds of risks that may be identified in an analysis.[22] Discussions highlighted the risks to collecting other personally identifiable information, such as name, phone number, or identifiers associated with a device[23] and ultimately directed the narrow scope of data to be collected by the App.[24] The Program's contractor, SamKnows Ltd., contributed another significant feature that reinforces the openness and transparency of the process by making publicly available the source code for the App under an Open Source license. Allowing all interested parties to review the functions of the client software helps further ensure consistency with the information model and privacy approach.

The Privacy Policy's adoption of a post-collection analysis and processing of results prior to release helps ensure that subscribers' privacy interests are protected in a manner that is responsive to the risks presented by the collected data.

The crowdsourcing nature of the trial ensured that volunteers knowingly and explicitly opted-in to the program.[25] Access to data collected in a study such as this one is often managed through user access to a web-hosted data set, which requires data to be identified with identifiers to facilitate a user's access of their individual data. The ability to identify individual devices from such a data set is accomplished through a process known as re-identification, wherein a third party uses comparisons of features or commonalities with other dataset to reconnect de-identified "anonymous" information back to data that could potentially identify specific individuals in what should otherwise be an anonymous dataset. De-identification attacks have been successful for browsing habits, health records and other commercial data. The MBA Program adopted a different approach that does not tag data with such identifiers that could also pose risks to privacy.[26] Instead the App enables users to manage and maintain their data independently running on their device, or by exporting the data for use in other programs.

---

mapping purposes); Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 28, 2012) available at https://www.fcc.gov/ecfs/filing/6017155673. See also Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 28, 2012) available at https://www.fcc.gov/ecfs/filing/6017155672 (emphasizing that time and location data is most sensitive).

[22] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 6, 2012) available at https://www.fcc.gov/ecfs/filing/6017149663 (discussing that data for public release be processed to preserve the value of the data for the specific use case and maintain privacy goals and disclosure controls by attendees).

[23] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 28, 2012) available at https://www.fcc.gov/ecfs/filing/6017155672 (calling release of client ID "highly problematic and not recommended"); Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Dec. 28, 2012) available at https://www.fcc.gov/ecfs/filing/6017155674.

[24] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Jan. 7, 2013) available at https://www.fcc.gov/ecfs/filing/6017157193 (indicating that data with identifiers would have value in certain industries, "potential concerns weighed in favor of adopting an anonymized approach for mobile performance data collected").

[25] Microstrategy Ex Parte Letter, GN No. 12-264 (Dec. 13, 2012) available at https://www.fcc.gov/ecfs/filing/6017152703 (emphasizing that program efforts depend on volunteer consent and "on an opt-in basis").

[26] Mobile Collaborative Ex Parte Letter, GN No. 12-264 (Feb. 25, 2013) available at https://www.fcc.gov/ecfs/filing/6017165566/document/7022125587 (discussing how a re-identification review could protect volunteer privacy).

a.   Administrative Features Protecting Privacy

SamKnows ensures that all technical mechanisms and methodologies conform to the most current privacy policies for each measurement effort. SamKnows takes all reasonable development steps necessary to conform the contractor-provided solution to the privacy policy of the respective measurement effort. SamKnows also maintains the privacy of consumer data through appropriate administrative, technical, and physical safeguards to ensure the security and confidentiality of data collected from volunteers against any hazards to the security or integrity, and conforms to the specific terms of the mobile privacy policies and terms and conditions developed for this effort.[27]

b. Technical Privacy Review and Data Processing

Past discussions with the collaborative and privacy experts identified three aspects of collected data meriting attention in the technical privacy review. High precision date and time information collected in the execution of the tests provide the opportunity to understand how broadband performance changes in time, but can present privacy concerns. High precision location information derived from the device's GPS provide the opportunity to understand how broadband performance changes across the nation, but presents potential for concerns that a specific volunteer could be identified with a particular location. In addition to high-precision GPS location information, information about the cellular base station a device is connected to is also collected and could pose privacy concerns. A variety of character strings are derived from the operating system that can vary subtly in case, punctuation, or other features that could present concerns that a specific volunteer could be identified with a particular set of tests.

The program values the privacy of its volunteers and has adopted a conservative approach to mitigating privacy concerns.   High precision time and location information were deleted before the initial Open Data release. The time and location measurement that was recorded is a crucial feature to understanding mobile broadband performance. However, the FCC employed a conservative policy of coarsening the level of detail of any time and location information in measurement results prior to release, in order to ensure that our policies of protecting sensitive data were achieved.

In order to minimize risks while preserving the utility of the Open Data, date and location variables were processed to preserve temporal and spatial variance among observations in any single batch of tests. Time elapsed and distances between discrete measurements were calculated and saved as new variables to replace the time and location data that was deleted from released data sets. The year, month, day, hour and in the case of some samples the year, quarter, and period replace the high-precision time of any observation.

Not all measurements collected by the App include a high-precision GPS location, but active measurement is unaffected and a location result is simply not recorded if unavailable. GPS locations results may be unavailable for a variety of reasons. For example a user may have disabled location services, a device may be unable to acquire a location during a test, or a device may simply not include necessary features to provide location services. When a GPS location is available, a valid latitude and longitude values are replaced in the open data with the Cellular Market Areas (CMAs) that comprise

---

[27] Mobile Collaborative Ex Parte Letter, Gn 12-264 (Dec. 28, 2012), available athttps://www.fcc.gov/ecfs/filing/6017155672.

Metropolitan Statistical Areas (MSAs) and Rural Service Areas (RSAs).[28] A map of CMAs can be found on the FCC web site.[29] While other geographies including a variety of other Census statistical areas in the figure below were considered, CMA level of geography balances the importance of minimizing potential risks or concerns for volunteers' privacy and value of maximizing the utility and flexibility of the geography for policy and academic research.[30]

Figure 3. Standard Hierarchy of Census Geographic Entities
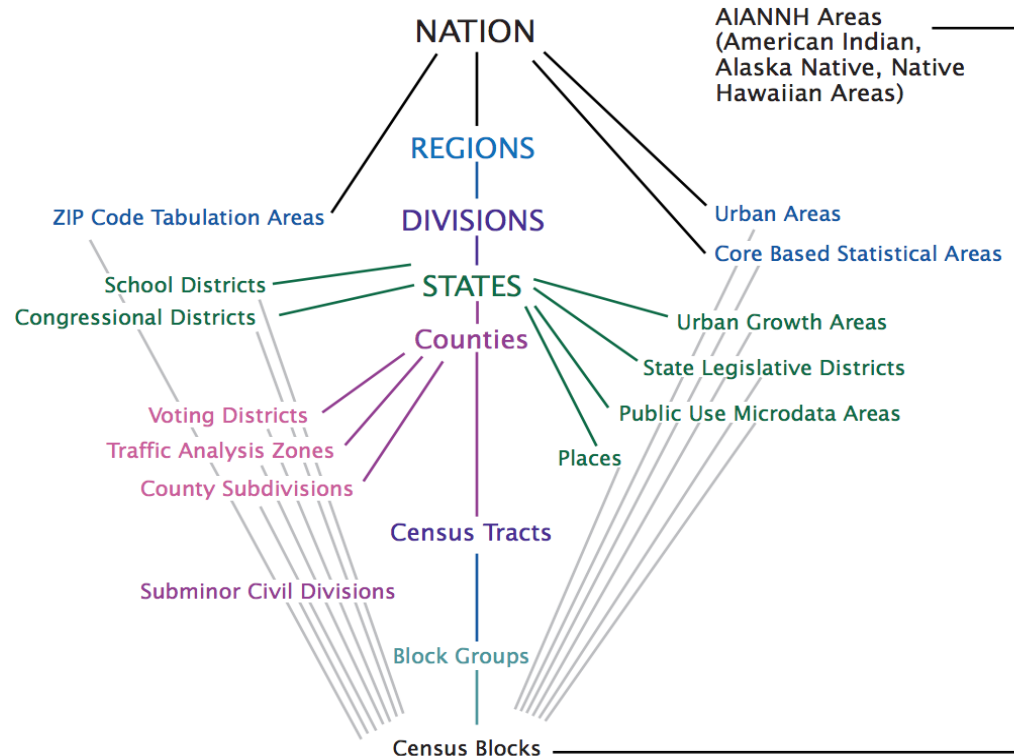
Source: U.S. Census Bureau, Hierarchy Diagrams, <www.census.gov/geo/reference/hierarchy.html>.

---

[28] 47 CFR 22.909. The Cellular Market Areas (CMAs) were created from the Metropolitan Statistical Areas (MSAs) defined by the Office of Management and Budget (1-305), the Gulf of Mexico (306), and Rural Service Areas (RSAs) established by the FCC which do not cross state borders (307-734). These RSAs include parts of Puerto Rico not already in an MSA (723-729), U.S. Virgin Islands (730-731), Guam (732), American Samoa (733), and Northern Mariana Islands (734). "Common Carrier Public Mobile Services Information, Cellular MSA/RSA Markets and Counties," Public Notice, Rep. No. CL-92-40, 7 FCC Rcd 742 (1992).

[29] See FCC WTB Geographic Licensing Schemes, "Cellular Market Areas (CMAs) Metropolitan Statistical Areas (MSAs) and Rural Service Areas (RSAs)", available at http://wireless.fcc.gov/auctions/data/maps/CMA.pdf.

[30] The US Census Bureau provides a variety of material describing the analytic value of a variety of geographies. *See generally*, Michael Ratcliffe, Charlynn Burd, Kelly Holder, and Alison Fields, Defining Rural at the U.S. Census Bureau (rel. Dec. 2016), available at http://www2.census.gov/geo/pdfs/reference/ua/Defining_Rural.pdf.

The number of measurements in each CMA depends on the number of volunteers running the FCC Speed Test App in that CMA. Since the population of CMAs varies widely, the number of measurements per CMA also varies widely. As the program matures in the future and the density of measurements increases, other geographies may be considered.

Other information about the source of the GPS observation, the accuracy in meters of the location and other information that poses minimal risk was not deleted. Cellular base station information, such as the cell tower identifier code that could provide a rough location was deleted, although other network information was included. Other features that inform the risk associated with sensitive aspects of data were also collected including the number of households, businesses, and total population of CMA's in order to help inform the level of diversity of potential locations that might be associated with any particular measurement in a CMA.[31]

Data were reviewed prior to release to define and ensure a minimum number of devices associated with any grouping of categories in order to avoid revealing patterns that would allow a third-party to filter data gathered from an individual device. This step also strengthens the data set against a third-party who might try to use a narrow group of results from the data to link results with other data sets. Coarsened time and location data was also reviewed in order to protect de-identified data.

The following data processing techniques were employed to perform the privacy analysis and to prepare the source data for release.[32]

1. **Time Related Aspects of Data**

Data collected during testing records two precise time and date variables that record the device's "*localtime*". These date and time values are removed and replaced with processed variables

- The year, month, day and hour are extracted from the *localtime* and saved as an ISO 8601 date string with the hour and minutes replaced with the value zero, e.g. '2013-11-13 07:00:00'.
  - For samples with low samples in a given aggregation, the *localtime* is processed to identify the date's year, quarter and saved as an integer.
- In some cases, multiple rows of data share the same *submission*_id from the same batch of tests, for example where multiple locations or bearer channel changes occur during a test. In order to preserve the ordinality of the rows, the order of all rows of data sharing the same *submission*_id from the same batch of tests is processed and saved as an integer.
  - row_number() over (partition by *submission*_id order by localtime) as metric_order
- In order to preserve the amount of time passing between the multiple rows of data sharing the same *submission_id* from the same batch of tests, the *localtime* is converted into seconds and subtracted from the preceding row to determine the number of seconds elapsed between the two rows and saved as an integer. This is done via the following SQL statements:
  - retr_seconds (l.localtime) - lag (retr_seconds (l.localtime)) over (partition by l.submission_id order by metric_order) as lead
  - CREATE FUNCTION retr_seconds (timestamp) RETURNS integer AS $$ select ((extract (hour from $1) * 3600)::integer + (extract (minutes from $1) * 60)::integer + (extract

---

[31] The calculation of the l-diversity

[32] The SQL code executing these table statements is available on the Program website.

(seconds from $1)::integer)) $$ LANGUAGE SQL IMMUTABLE RETURNS NULL ON NULL INPUT;

For download speed, upload speed and UDP test data appearing in the curr_httpget, curr_httpost, and curr_udplatency tables, a *QUARTER* and *HOUR* variable were processed and appended for each row result to facilitate use of the data. These values are extracted from the *localdtime* and appended in integer variables in the table. This is done via the following SQL statements:

- SET HOUR
    - update curr_httpget set hour = extract (hour from localdtime) where hour is null;

2. Location Related Aspects of Data

As tests execute, the operating system can provide the current GPS latitude and longitude, although for a variety of reasons not every test measurement may have a GPS location. Each GPS longitude and latitude was deleted and replaced with a CMA when available for export in the dataset. A separation distance between each individual location update in a given test is calculated to identify the distance traveled during a test.

GPS locations can be processed to provide the cellular market area where tests were executed. No additional geographic information will be available for measurements that did not record a valid GPS longitude and latitude. The following SQL statements identified the CMA in which a GPS location is located:

- UPDATE curr_location SET geom = ST_SetSRID(ST_MakePoint(longitude,latitude),4326) where extract (year from localdtime) = 2016 and geom is NULL;
- Update curr_location set cma_id = as c.id from cma_2010 c where curr_location.cma_id is not null and st_within (curr_location.geom, c.geom) ;
- (((ST_Distance(l.geom , lag (l.geom) over (partition by l.submission_id order by   metric_order))))) as separation_distance
- Cell Tower ID. Base station ID's and Cell Tower IDs may be used to derive a rough location and will be deleted, although other network information may be included.

3. Series of characters with identifiable features

Test data that is derived from operating system APIs can expose particular features of a local area network or version of device or operating system. Because such features could potentially be used to associate test data with a particular device, data was processed to remove variations in case or punctuation if variations are associated with low numbers of samples. Specifically case, punctuation and other white space of the network_operator_name, network_operator_code, sim_operator_name, and sim_operator_code for certain samples were processed to eliminate variations in the strings. Out of an abundance of caution the "submission_id" processed by the SamKnows backend is replaced in the dataset with an artificially generated random list of numbers to serve as a primary key across all measurement tables for a give unique suite of upload, download, latency, and packet loss tests.[33]

---

[33] The submission_id is hashed against a variety of information including the IP address of a client uploading data to the storage backend. The artificially generated key is included to ameliorate the risk of attacks on the hashed

4. Table joins and aggregations of sets identifying law sample counts

The features of a suite of tests share certain characteristics such as the model type, test submission type, CMA location, year/month/day/hour time information, and carrier network features that could potentially be used to identify a device producing measurements. Other features of data, such as the results of active test results or the instantaneous radio characteristics during a test naturally could vary even during a suite of tests and were not deemed to pose a concern. All test samples and features of test data of concern were aggregated to identify any combination by hour with fewer than 3 samples or a k-anonymity of 2. A set of submission_id's requiring analysis and potential processing were identified that composed roughly 25% of the total sample of measurements.

4. Processing of data in stratifications with sample counts to raise k-anonymity

For all samples that were unique in a given combination, a combination of approaches discussed above were tested to raise the k-anonymity to 2, including by removing fields such as the model or CMA location or reducing the specificity of the time reported by removing the month and replacing with a Quarterly value. For scheduled test submission_types processing attempted to preserve the period of the measurement, a fundamental feature of the test and the CMA location and thus the model type was excluded and carrier network information was processed. For init test submission_types the potential research value of identifying whether a datacap was reached limited a scheduled test from occurring was preserved for most samples by providing the day of the month and period but not the month in a quarter nor the CMA location of the test. Because manual tests are user initiated and could pose a higher risk of exposing a user behavior, less precision in the time and location of the test variables was preserved. Tables describing the processing applied and separately exported datasets are included in the Appendix below.

---

primary key. Future versions of backend processing software will employ a different hashing technique that does not rely on client IP addresses.

APPENDIX A: PRIVACY ANALYSIS OF LOW FREQUENCY DATA

Row level results for this data export were selected by a cross-table primary key, *submission_id*, that included:

- A combination of SIM and network operator code and names to denote a connection to a US carrier by
  - A sim operator code matching a properly formatted mobile country code for the United States and no network_operator_code; or
  - A properly formatted network operator code (NOC) matching SIM mobile country code (MCC) and NULL NOC; a partial NOC matching a MCC and a match for a major four carrier name or a valid MCC SIM; and
- Time between '2013-11-01 00:00:00' and '2016-07-01 00:00:00'

In addition any submission_id with a valid cma_id location is included in the scope of data to be exported.

Three methods are used to identify low-frequency combinations of data in tables and across tables that will be targets for analysis and processing. First all submission_ids found in the curr_submission table that are unique by hour, model, os_version, sim_operator_code and submission_type are flagged. Both the same and different models may have different os_versions so checks for os_version must be done separately from model. In addition, while init and scheduled or manual tests may occur together or apart they must be treated in the strictest sense as a single low-frequency batch of such tests could imply a single handset produced the handsets. Second all submission_ids identified by a cross-table query of all potentially sensitive values collated to identify any unique instances of a combination of the following fields: submission_type, date truncated by hour, model, os_version,cma_id,network_operator_name, network_operator_code, sim_operator_name, sim_operator_code. Finally, unique occurrences of a test by hour in the active metric tables is identified and added to the list of submission_ids to be processed. These low-frequency samples are excluded from the set of "clean" and subject to analysis and processing.

Low-frequency samples are successively aggregated at levels that preserve the most temporal, location or other features. Coarsening samples that may have passed earlier aggregations in order to pool larger numbers of samples in time or space was also considered. However, a large portion of low-frequency samples were able to be aggregated in the first pass with processing of network operator strings and temporal variables that preserved much of the research value of the data. Successive aggregations all preserved research value of the underlying active test measurements and other supporting data while sacrificing more sensitive features such as the device model.

Unrelated to low-frequency sample analysis, the data set was also analyzed for risks implied by l-diversity. Location aspects of the data are a potential quasi-identifier and in each CMA geography the population, population density, number of households and businesses were reviewed. In addition, the cross-table primary key, *submission_id* created by the SamKnows ingress software was replaced with a randomized arbitrary bigint list.

Table Name	Table References	

exportids		• Table to define the scope of measurements for the export identifies a combination of sim and network operator code and names to denote a connection to a US carrier by
		  ○ A sim operator code matching a properly formatted mobile country code for the United States and no network_operator_code; or
		  ○ A properly formated network operator code matching SIM MCC and NULL NOC; a partial NOC matching a MCC and a match for a major four carrier name or a valid MCC SIM; and
		• Time between '2013-11-01 00:00:00' and '2016-07-01 00:00:00'
		
		```create table exportids as (select submission_id from curr_networkdata a where a.localdtime between '2013-11-01 00:00:00' and '2016-07-01 00:00:00' and ((sim_operator_code ~ '(31[106])(...)' and network_operator_code ~ '(^$)\|(null)') or (network_operator_code ~ '(31[016])(...)' or (network_operator_code ~ '(31[016])' and (sim_operator_code ~ '(31[016])(...)' or (network_operator_name ~ '(^[\s]*at[\s]*[&]*[\s]*t[\s]*)\|(^[^\s]*t[\s\-]*[\s]*mobile)\|(verizon)\|(sprint[\s]*$)')))))) ; --SELECT 71172918```

exportids2	curr_location, exportids	Set of all measurements within scope of the data export ```create table exportids2 as (select (submission_id) from ((select distinct submission_id from curr_location where cma_id is not null) union (select distinct submission_id from exportids)) as c);```
exportids_clean	exportids2, pr_exclude	Set of all measurements within scope of the data export that in combination did not yield

		records that were identified in the privacy analysis for review
		```create table exportids_clean as (select distinct e.submission_id from exportids2 e left pr_exclude p on e.submission_id = p.submission_id where p.submission_id is NULL);```
		```--SELECT 20487097```

pr_exclude	`Prsubmissionuniq_v2ids,`
	`pr9v2ids`
		Set of low frequency measurements the focus of the privacy analysis.
		 Measurements in combination of Device Model, Operating System version, SIM and Network Operator Code and Name, Test Submission Type and Date and Time to the Hour yield low frequency of test execution.
		 Base table for defining the submission_ids to be processed before export.
		
		```create table pr_exclude as (select submission_id, date from ((select s.submission_id, date_trunc as date from prsubmissionuniq_v2ids s) union (select s.submission_id , date from pr9v2ids s)) as foo);```
		```SELECT 6295696```

`prsubmissionuniq_v2ids`	`prsubmissionuniq_v2`	Set of measurements that are unique by all relevant features in the submission table for the measurements within scope of the data export
		
		```create table prsubmissionuniq_v2ids as (select s.submission_id , p.* from prsubmissionuniq_v2 p inner join curr_submission s on p.date_trunc = date_trunc('hour',s.localtime) and p.model = s.model and p.sim_operator_code =s.sim_operator_code and p.os_version = s.os_version and p.submission_type = s.submission_type);```
		```--SELECT 4434774```

`prsubmissionuniq_v2`		Group by aggregation query to identify what features yield unique test executions by all relevant features in the submission table for the measurements within scope of the data export, including Device Model, Operating System version, SIM Operator Code, Test Submission Type and Date and Time to the Hour ```create table prsubmissionuniq_v2 as (select count (*) as tally , date_trunc('hour' , s.localtime) , s.model ,```

		s.sim_operator_code , os_version, submission_type from **exportids2** e inner join **curr_submission** s on e.submission_id = s.submission_id group by date_trunc('hour' , s.localdtime) , s.model , s.sim_operator_code , os_version , submission_type having count(*) =1) ; --SELECT 4082542
pr9v2ids	pr9v2, curr_submission, curr_networkdata, curr_location	Set of measurements that are unique by all relevant features in the submission, networkdata, and location tables that are most relevant for privacy review for the measurements within scope of the data export including including Device Model, Operating System version, SIM and Network Operator Code and Name, Test Submission Type, CMA location ID, and Date and Time to the Hour `create table `**`pr9v2ids`**` as (select s.submission_ID, s.submission_type, p.date , p.model, p.os_version, p.cma_id, p.network_operator_name, p.network_operator_code, p.sim_operator_name, p.sim_operator_code `**`from pr9v2 p inner join curr_submission`**` s on p.date = date_trunc ('hour', s.localdtime) and p.submission_type = s.submission_type and p.model = s.model and p.os_version=s.os_version and p.sim_operator_code = s.sim_operator_code inner join `**`curr_networkdata`**` n on s.submission_id = n.submission_id and p.network_operator_code = n.network_operator_code and p.network_operator_name = n.network_operator_name and p.sim_operator_name = n.sim_operator_name inner join `**`curr_location`**` l on s.submission_id = l.submission_id and p.cma_id = l.cma_id ); --SELECT 4988846`
pr9v2	exportids2, curr_submission	Group by aggregation query to identify the features yielding a low frequency test execution by all relevant features in the submission table for the measurements within scope of the data export `     table pr9v2 as (select count(distinct s.submission_id)  , submission_type, date_trunc( 'hour', s.localdtime) as date ,  s.model, s.os_version, l.cma_id, n.network_operator_name, n.network_operator_code, n.sim_operator_name, n.sim_operator_code `**`from exportids2 e inner join curr_submission`**` s on s.submission_id = e.submission_id inner join  curr_location l`

		on e.submission_id = l.submission_id inner join curr_networkdata n on e.submission_id = n.submission_id group by date_trunc('hour', s.localdtime), s.model, l.cma_id, n.network_operator_name, n.network_operator_code, n.sim_operator_name, n.sim_operator_code, submission_type , os_version having count (*) < 3) ; --SELECT 4171795
`pr9v2_exclude_ids_DISTINCT`	`PR9V2_EXCLUDE_IDS`	Table of aggregation of features for set measurements that are unique by all relevant features in the submission, networkdata, and location tables. Source table for analysis of the low frequency measurements to identify new aggregations on Device Model, Operating System version, SIM and Network Operator Code and Name, Test Submission Type, CMA location ID, and Date and Time to produce aggregations of k-anonymity of two or greater. This table is a "distinct" operation to remove redundant submission_ids.This table is the base for the first round of aggregations and the list of all pr_exclude IDs. create table **pr9v2_exclude_ids_DISTINCT** as ((SELECT * FROM **PR9V2_EXCLUDE_IDS**) UNION (SELECT * FROM **PR9V2_EXCLUDE_IDS**)); --SELECT 2947142

`pr9v2_exclude_ids`	`pr9v2_exclude,`
	`curr_submission,`
	`curr_location,`
	`curr_networkdata`
		Source table for pr9v2_exclude_ids_DISTINCT identifying aggregation characteristics in **pr9v2_exclude** that match the relevant features in the cross-table join for the relevant tables.
		
		create table **pr9v2_exclude_ids** as (select s.submission_ID, s.submission_type, p.date , p.model, p.os_version, p.cma_id, p.network_operator_name, p.network_operator_code, p.sim_operator_name, p.sim_operator_code from **pr9v2_exclude** p inner join **curr_networkdata** s on p.date = date_trunc ('hour', s.localdtime) and p.submission_type = s.submission_type and p.model = s.model and p.os_version=s.os_version and p.sim_operator_code = s.sim_operator_code inner join **curr_networkdata** n on s.submission_id = n.submission_id and p.network_operator_code =

		n.network_operator_code and p.network_operator_name = n.network_operator_name and p.sim_operator_name = n.sim_operator_name inner join **curr_location** l on s.submission_id = l.submission_id and p.cma_id = l.cma_id); --SELECT 17622902
pr9v2_exclude	**Pr_exclude, curr_submission, curr_locationcurr_netwo rkdata**	Table of aggregation of features for set measurements that are unique by all relevant features in the submission, networkdata, and location tables. Source table for analysis of the low frequency measurements to identify new aggregations on Device Model, Operating System version, SIM and Network Operator Code and Name, Test Submission Type, CMA location ID, and Date and Time create table **pr9v2_exclude** as (select count(distinct s.submission_id) , submission_type, date_trunc('hour', s.localdtime) as date , s.model, s.os_version, l.cma_id, n.network_operator_name, n.network_operator_code, n.sim_operator_name, n.sim_operator_code **from pr_exclude e inner join curr_submission** s on s.submission_id = e.submission_id inner join **curr_location** l on e.submission_id = l.submission_id inner join **curr_networkdata** n on e.submission_id = n.submission_id group by date_trunc('hour', s.localdtime), s.model, l.cma_id, n.network_operator_name, n.network_operator_code, n.sim_operator_name, n.sim_operator_code, submission_type , os_version ;r SELECT 4616355
pr10v2ids_exclude	**Pr10v2_exclude, pr9v2_exclude_ids_DISTI NCT**	This table holds 2.2 million aggregated low-frequency scheduled tests samples grouped by Year/Month/Period, model, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes. 2.2 million test samples achieve a k-anonymity of two or greater with this aggregation. This aggregation does not preserve either the hour or day of the month but does provide the period, which is valuable for scheduled test analysis. create table **pr10v2ids_exclude** as (select s.submission_ID, s.submission_type, p.date , p.period, p.model, p.os_version, p.cma_id, p.network_operator_name, p.network_operator_code, p.sim_operator_name, p.sim_operator_code from **pr10v2_exclude** p inner join **pr9v2_exclude_ids_DISTINCT** s on p.date = date_trunc ('month', s.date) and p.period = retr_period(s.date) and p.submission_type = s.submission_type and

| | | ```
p.model = s.model and p.os_version =
s.os_version and p.sim_operator_code =
 upper(regexp_replace(s.sim_operator_code,
'\s*', '', 'g'))   and
  p.network_operator_code =
upper(regexp_replace(s.network_operator_code,
'\s*', '', 'g'))    and
p.network_operator_name =
upper(regexp_replace(s.network_operator_name,
'\s*', '', 'g'))   and p.sim_operator_name =
upper(regexp_replace(s.sim_operator_name,
'\s*', '', 'g'))       and p.cma_id =
s.cma_id );
--
mmba=# select count(distinct submission_id )
from  pr10v2ids_exclude;;
 count
---------
2233546
(1 row)
``` |
| **pr10v2_exclude** | **pr9v2_exclude_ids_DISTINCT** | This table aggregates low-frequency scheduled tests samples by Year/Month/Period, model, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes.  2.5 million test samples achieve a k-anonymity of two or greater with this aggregation.<br><br>```
create table pr10v2_exclude as (SELECT
count(submission_id) as tally ,
submission_type , date_trunc('month', date)
as date , retr_period(date) as period ,
model, os_version, cma_id ,
upper(regexp_replace(network_operator_name,
'\s*', '', 'g'))  as network_operator_name,
upper(regexp_replace(sim_operator_name,
'\s*', '', 'g')) as sim_operator_name ,
upper(regexp_replace(network_operator_code,
'\s*', '', 'g')) as network_operator_code,
upper(regexp_replace(sim_operator_code,
'\s*', '', 'g')) as sim_operator_code from
pr9v2_exclude_ids_DISTINCT where
submission_type = 'scheduled_tests' group by
upper(regexp_replace(network_operator_name,
'\s*', '', 'g')) ,
upper(regexp_replace(sim_operator_name,
'\s*', '', 'g')),
upper(regexp_replace(network_operator_code,
'\s*', '', 'g')),
upper(regexp_replace(sim_operator_code,
'\s*', '', 'g'))  , cma_id , model,
os_version , date_trunc('month', date),
retr_period(date) , submission_type having
count(*) > 1) ;
-- (257009 rows)
``` |
| pr10v2ids_exclude_man | **Pr10v2_exclude_man, pr9v2_exclude_ids_DISTINCT** | This table holds 32605 aggregated low-frequency manual test samples grouped |

| | | by Year/Quarter/Hour, model, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes.  32605 test samples achieve a k-anonymity of two or greater with this aggregation.  This aggregation does not preserve either the year or day of the month but does provide the hour, which is valuable for manual test analysis.<br><br>```
create table pr10v2ids_exclude_man  as
(select s.submission_ID, s.submission_type,
p.year , p.quarter, p.hour, p.model,
p.os_version, p.cma_id,
p.network_operator_name,
p.network_operator_code, p.sim_operator_name,
p.sim_operator_code   from pr10v2_exclude_man
p inner join pr9v2_exclude_ids_DISTINCT  s on
p.year = extract (year from s.date) and
 p.quarter =  retr_quarter (s.date) and
p.hour =  extract(hour from s.date)   and
p.submission_type = s.submission_type and
p.model = s.model and p.os_version =
s.os_version and p.sim_operator_code =
 upper(regexp_replace(s.sim_operator_code,
'\s*', '', 'g'))   and
  p.network_operator_code =
upper(regexp_replace(s.network_operator_code,
'\s*', '', 'g'))   and
p.network_operator_name =
upper(regexp_replace(s.network_operator_name,
'\s*', '', 'g'))   and p.sim_operator_name =
upper(regexp_replace(s.sim_operator_name,
'\s*', '', 'g'))        and p.cma_id =
s.cma_id );
--SELECT 32605
``` |
| **pr10v2_exclude_man** | **pr9v2_exclude_ids_DISTINCT** | This table aggregates low-frequency manual tests samples by Year/Quarter/Period, model, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes.  2.5 million test samples achieve a k-anonymity of two or greater with this aggregation.<br><br>```
create table pr10v2_exclude_man as (SELECT
count(submission_id) as tally ,
submission_type , extract (year from date) as
year, retr_quarter(date) as quarter, extract
(hour from date) as hour, model, os_version,
``` |

		<pre>cma_id ,
upper(regexp_replace(network_operator_name,
'\s*', '', 'g')) as network_operator_name,
upper(regexp_replace(sim_operator_name,
'\s*', '', 'g')) as sim_operator_name ,
upper(regexp_replace(network_operator_code,
'\s*', '', 'g')) as network_operator_code,
upper(regexp_replace(sim_operator_code,
'\s*', '', 'g')) as sim_operator_code from
pr9v2_exclude_ids_DISTINCT where
submission_type ~* 'manual' group by
upper(regexp_replace(network_operator_name,
'\s*', '', 'g')) ,
upper(regexp_replace(sim_operator_name,
'\s*', '', 'g')),
upper(regexp_replace(network_operator_code,
'\s*', '', 'g')),
upper(regexp_replace(sim_operator_code,
'\s*', '', 'g')) , cma_id , model,
os_version , extract (year from date),
retr_quarter(date) , extract(hour from date)
, submission_type having count(*) > 1) ;
sum

32605
(1 row)</pre>
pr11v2ids_EXCLUDE	**pr11v2ids_EXCLUDE_sch**, **pr10v2ids_exclude_man**	This table holds the remaining samples after the first aggregation of manual and scheduled tests. <pre>create table pr11v2ids_EXCLUDE as (select
a.* from pr11v2ids_EXCLUDE_sch a left join
pr10v2ids_exclude_man b on a.submission_id =
b.submission_id where b.submission_id is NULL
)
mmba-# ;
SELECT 675585</pre>
pr11v2ids_EXCLUDE_sch	pr9v2_exclude_ids_DISTINCT, **pr10v2ids_exclude**	<pre>create table pr11v2ids_EXCLUDE_sch as (
select a.* from pr9v2_exclude_ids_DISTINCT a
left join pr10v2ids_exclude b on
a.submission_id = b.submission_id where
b.submission_id is NULL)
-- count 706088</pre>
pr11v2ids_exclude_init_YQDP_nomodel	**pr11v2_exclude_init_YQDP_nomodel**, **pr11v2ids_EXCLUDE**	This table holds 164919 aggregated low-frequency init test samples grouped by Year/Quarter/Day/Period, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes. Test samples achieve a k-anonymity of two or greater with this aggregation. This aggregation does not preserve the month or device model or manufacturer but does provide the operating system version and day and

		hour, which are valuable temporal variables for init test analysis. ``` create table pr11v2ids_exclude_init_YQDP_nomodel as (select s.submission_ID, s.submission_type, p.year , p.quarter, p.day, p.period, p.os_version, p.cma_id, p.network_operator_name, p.network_operator_code, p.sim_operator_name, p.sim_operator_code from pr11v2_exclude_init_YQDP_nomodel p inner join pr11v2ids_EXCLUDE s on p.year = extract (year from s.date) and p.quarter = retr_quarter(s.date) and p.day = extract (day from s.date) and p.period = retr_period(s.date) and p.submission_type = s.submission_type and p.os_version = s.os_version and p.sim_operator_code = upper(regexp_replace(s.sim_operator_code, '\s*', '', 'g')) and p.network_operator_code = upper(regexp_replace(s.network_operator_code, '\s*', '', 'g')) and p.network_operator_name = upper(regexp_replace(s.network_operator_name, '\s*', '', 'g')) and p.sim_operator_name = upper(regexp_replace(s.sim_operator_name, '\s*', '', 'g')) and p.cma_id = s.cma_id); SELECT 164919 ```
pr11v2_exclude_init_YQDP _nomodel	**pr11v2ids_EXCLUDE**	This table aggregates low-frequency init tests samples by Year/Quarter/Day/Period, os version, CMA and a whitespace and case insensitive matching of SIM and network operator names and codes.. ``` create table pr11v2_exclude_init_YQDP_nomodel as (SELECT count(submission_id) as tally , submission_type , extract (year from date) as year, retr_quarter(date) as quarter, extract (day from date) as day , retr_period (date) as period, os_version, cma_id , upper(regexp_replace(network_operator_name, '\s*', '', 'g')) as network_operator_name, upper(regexp_replace(sim_operator_name, '\s*', '', 'g')) as sim_operator_name , upper(regexp_replace(network_operator_code, '\s*', '', 'g')) as network_operator_code, upper(regexp_replace(sim_operator_code, '\s*', '', 'g')) as sim_operator_code from pr11v2ids_EXCLUDE where submission_type ~* 'init_test' group by upper(regexp_replace(network_operator_name, '\s*', '', 'g')) , upper(regexp_replace(sim_operator_name, '\s*', '', 'g')), upper(regexp_replace(network_operator_code, ```

| | | ```
'\s*', '', 'g')),
upper(regexp_replace(sim_operator_code,
'\s*', '', 'g'))  , cma_id , os_version ,
extract (year from date), retr_quarter(date)
, extract (day from date), retr_period (date)
, submission_type having count(*) > 1) ;
SELECT 63152
``` |
| `pr12v2ids_EXCLUDE` | `pr11v2ids_EXCLUDE_sch,`<br>`pr10v2ids_exclude_man` | The remaining 510440 samples after the first aggregations of init, scheduled and manual tests are further aggregated to increase the k value in various aggregations.<br><br>```
create table pr12v2ids_EXCLUDE as ( select
a.* from pr11v2ids_EXCLUDE a left join
pr11v2ids_exclude_init_YQDP_nomodel b on
a.submission_id = b.submission_id where
b.submission_id is NULL )
--SELECT 510440
``` |
| `pr13ids` | `Pr13_exclude,`<br>`pr12v2ids_EXCLUDE` | 488372 remaining samples are aggregated to include Year/Quarter, CMA location, and the network operator code and name.  While the temporal and device related information is unavailable for analysis, the remaining samples location and network operator information by year and quarter are valuable for analysis of the active metric test results as well as network related signal strength, bearer channel and other features.<br><br>```
create table pr13ids as (select
s.submission_ID, s.submission_type, p.year ,
p.quarter,   p.cma_id,
p.network_operator_name,
p.network_operator_code    from pr13_exclude p
inner join pr12v2ids_EXCLUDE  s on p.year =
extract (year from  s.date) and p.quarter =
retr_quarter(s.date) and p.submission_type =
s.submission_type and p.network_operator_code
=
upper(regexp_replace(s.network_operator_code,
'\s*', '', 'g'))   and
p.network_operator_name =
upper(regexp_replace(s.network_operator_name,
'\s*', '', 'g'))    and p.cma_id = s.cma_id
);
SELECT 488372
``` |

pr13	pr12v2ids_EXCLUDE	create table **pr13** as (SELECT count(submission_id) as tally , submission_type , extract (year from date) as year, retr_quarter(date) as quarter , cma_id , upper(regexp_replace(network_operator_name, '\s*', '', 'g')) as network_operator_name, upper(regexp_replace(network_operator_code, '\s*', '', 'g')) as network_operator_code from **pr12v2ids_EXCLUDE** group by upper(regexp_replace(network_operator_name, '\s*', '', 'g')) , upper(regexp_replace(network_operator_code, '\s*', '', 'g')), cma_id , extract (year from date), retr_quarter(date) , submission_type having count(*) > 1) ; SELECT 41655
pr14	pr13ids, pr12v2ids_EXCLUDE	21457 remaining low-frequency samples not subject to aggregation are excluded from the exported data set. create table pr14 as (select a.* from pr12v2ids_EXCLUDE a left join pr13ids b on a.submission_id = b.submission_id where b.submission_id is NULL) ; SELECT 21457

Summary of Analysis of
Potential Low-Frequency Outliers in Active Test Results by Hour Block

Summarized below are results of two sets of analysis to identify whether any active metric results appear alone in any hour block in either the export cleared or export processing sets of submission_ids. No records among those flagged for export processing were discovered in isolation by hour. However, some small number of records were discovered to appear in the set of results cleared for export. Those records are removed from the general export and included in the set of results flagged for privacy processing.

Export Filters applied:
- where submission_id not in (select submission_id from pr_exclude_active_get2)
- where submission_id not in (select submission_id from pr_exclude_active_post2)
- where submission_id not in (select submission_id from pr_exclude_active_udp2)

Analysis:

```
create table pr_exclude_active_get as (select distinct submission_id from (
select e.submission_id from  pr_exclude e inner join  curr_httpget c  on
c.submission_id = e.submission_id  where date_trunc('hour', localdtime) in
(select date from (select count( distinct e.submission_id) ,
```

```
 date_trunc('hour', localdtime) as date from pr_exclude e inner join
 curr_httpget c  on c.submission_id = e.submission_id  group by
date_trunc('hour', localdtime) having count( distinct e.submission_id) =1 )
as foo) ) as foo);
--SELECT 176
mmba=# select count(distinct E.submission_id) from pr_exclude_active_get e
inner join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
  176
(1 row)


create table pr_exclude_active_get2 as (select distinct submission_id from (
select e.submission_id from  curr_submission_primarykey_clean e inner join
 curr_httpget c  on c.submission_id = e.submission_id  where
date_trunc('hour', localdtime) in (select date from (select count( distinct
e.submission_id) ,  date_trunc('hour', localdtime) as date from pr_exclude e
inner join  curr_httpget c  on c.submission_id = e.submission_id  group by
date_trunc('hour', localdtime) having count( distinct e.submission_id) =1 )
as foo) ) as foo);
SELECT 798
select count(distinct E.submission_id) from pr_exclude_active_get e inner
join pr_EXCLUDE s on e.submission_id = s.submission_id;
mmba=# select count(distinct E.submission_id) from pr_exclude_active_get2 e
inner join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
     0
(1 row)




create table pr_exclude_active_post as (select distinct submission_id from (
select e.submission_id from  pr_exclude e inner join  curr_httppost c  on
c.submission_id = e.submission_id  where date_trunc('hour', localdtime) in
(select date from (select count( distinct e.submission_id) ,
 date_trunc('hour', localdtime) as date from pr_exclude e inner join
 curr_httppost c  on c.submission_id = e.submission_id  group by
date_trunc('hour', localdtime) having count( distinct e.submission_id) =1 )
as foo) ) as foo);
--SELECT 348
mmba=# select count(distinct E.submission_id) from pr_exclude_active_POST e
inner join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
  348
(1 row)


create table pr_exclude_active_post2 as (select distinct submission_id from (
select e.submission_id from  curr_submission_primarykey_clean e inner join
 curr_httppost c  on c.submission_id = e.submission_id  where
date_trunc('hour', localdtime) in (select date from (select count( distinct
```

```
e.submission_id) ,  date_trunc('hour', localdtime) as date from pr_exclude e
inner join  curr_httppost c  on c.submission_id = e.submission_id  group by
date_trunc('hour', localdtime) having count( distinct e.submission_id) =1 )
as foo) ) as foo);
SELECT 1719
select count(distinct E.submission_id) from pr_exclude_active_POST2 e inner
join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
      0
(1 row)


create table pr_exclude_active_udp as (select distinct submission_id from (
select e.submission_id from  pr_exclude e inner join  curr_udplatency c  on
c.submission_id = e.submission_id  where date_trunc('hour', localdtime) in
(select date from (select count( distinct e.submission_id) ,
 date_trunc('hour', localdtime) as date from pr_exclude e inner join
 curr_udplatency  c  on c.submission_id = e.submission_id  group by
date_trunc('hour', localdtime) having count( distinct e.submission_id) =1 )
as foo) ) as foo);
--SELECT 335
mmba=# select count(distinct E.submission_id) from pr_exclude_active_UDP e
inner join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
  335
(1 row)

create table pr_exclude_active_udp2 as (select distinct submission_id from (
select e.submission_id from  curr_submission_primarykey_clean e inner join
 curr_udplatency c  on c.submission_id = e.submission_id  where
date_trunc('hour', localdtime) in (select date from (select count( distinct
e.submission_id) ,  date_trunc('hour', localdtime) as date from pr_exclude e
inner join  curr_udplatency  c  on c.submission_id = e.submission_id  group
by date_trunc('hour', localdtime) having count( distinct e.submission_id) =1
) as foo) ) as foo);
SELECT 1629
select count(distinct E.submission_id) from pr_exclude_active_UDP2 e inner
join pr_EXCLUDE s on e.submission_id = s.submission_id;
count
-------
      0
(1 row)
```

APPENDIX B: REFERENCE DOCUMENT

Mobile Broadband Performance Application Privacy Notice and Terms of Use

(Version 2.0 Updated October 2016)

What Information We Collect

The information the FCC Speed Test App ("Application") collects is limited to information used to measure your mobile broadband service. Your location and time when a test is run, and IP address when data is uploaded to our servers is collected and used in processing your results. We do not collect other personally identifiable information, such as name, phone number, or identifiers associated with your device. We collect five kinds of information to measure the performance of your mobile broadband service.

- Location
  - When we measure your mobile broadband performance, we identify the location and unique ID of the cell tower that is providing that service, and the GPS location of your device when the measurements were made.
- Time of Data Collection
  - We record the times when the Application performs measurements of your mobile broadband performance. This is necessary because some measurements function by recording a start and end time.
- Device Type and Operating System Version
  - When the Application starts, we record the manufacturer and model of device you are using, and the version of the operating system installed on your device.
- Cellular Performance and Characteristics
  - We collect information about your cellular device performance, such as your service provider, the strength of the radio signal, and what type of connection and service you have.
- Broadband Performance
  - The Application tests your mobile broadband upload and download rate, round trip latency, and packet loss measured between your device and our servers managed by the FCC, the FCC contractor, and other project partners.

This data is stored on your device in the Application's sandbox (a protected private area for the Application) for you to review, and periodically is uploaded to our servers. You may delete older data periodically from your device. You can find more detail in the FCC's technical summary of this program. https://www.fcc.gov/measuring-broadband-america/mobile/technical-summary.html

Data Sharing

The data we collect is helpful for consumers, the FCC, and others interested in understanding mobile broadband in the United States. We share your mobile broadband performance information in four ways.

- **Internally**

We may share your mobile broadband performance information within the FCC to write our Measuring Broadband America Report on Mobile Performance and to support our development of broadband policy.

- **Third-Party Partners Who Help Us Run Our Test and Support Our Program**

We share your mobile broadband performance information with our contractor who manages the measurement program, with partners who support the program and architecture, and with partners supporting our analysis or contributing peer reviews. We share data subject to agreements that include strict requirements to safeguard the data, not disclose the data, and use data only in furtherance of the goals of the program.

The FCC has contracted with a broadband measurement company, SamKnows, to help manage the program and conduct its technical functions. SamKnows performs functions such as writing and maintaining the software that performs the test measurements, responding to customer service requests, managing the Application and the execution of tests, collecting and protecting data gathered in the trial, and providing analysis and support for the preparation of FCC reports. The terms of our contract with SamKnows include obligations to protect the data and your privacy.

Research platforms and your carrier may support various aspects of the testing and architecture for the program. We may share your detailed mobile performance data with these partners supporting our data processing and analysis for the specific purpose of identifying patterns in the data and recommending ways of addressing any concerns for your privacy and anonymity. The FCC directs researchers, its contractor, and your carrier working under non-disclosure agreements (NDA) to analyze the collected data for privacy risks or concerns and to plan for future measurement efforts.

Researchers and other partners supporting this analysis are skilled in the field of statistics and computer science work to identify patterns that a third party might exploit in ways to compromise your privacy, for example, by comparing the location information with other sources of information. If we find concerns, we will take steps to address the risks by making the time, location, or other information less specific; limiting the combinations of information we release; or deleting information. These partners are required to sign an NDA and must agree to data-handling practices that safeguard the data. The NDA prohibits any public disclosure of the data except in the form of a report to the FCC, limits both the access duration and specific research purposes of data, and requires disclosures if the data is ever compromised in any way.

- **Pursuant to legitimate requests by law enforcement or where otherwise required by law or regulation**

  ○ We may share mobile broadband performance data collected by this Application subject to legitimate requests by law enforcement or where otherwise required by law or regulation.

- **De-Identified Data Available to the Public**

We will publish de-identified data in summary statistics, coarsened data, and other minimally necessary datasets that support our review of the statistics only if we determine that the data poses very low risks to your privacy. These include:

○ **Summary Data**

We will publish averages and other summary statistics that describe important features of mobile broadband performance. These summary statistics pose very low risks to your privacy.

○ **Coarsened Data**

We will publish only de-identified data in a coarsened form that supports the averages and other statistics that we publish in our reports. Prior to release of this data, the FCC and researchers skilled in the field of statistics and computer science will identify patterns in the raw data that could be exploited to compromise your privacy. If we find concerns, we will take steps to address the risks by making the time, location or other information less specific; limiting the combinations of information we release; or deleting information.

For example, the analysis we perform would identify and address risks from a third party combining the location information in the data we collect with other sources of information to try to identify your device in our database. We will publicly release the results of this analysis to ensure the quality of our review, and will also describe the measures we took to protect your anonymity. We will release information about the time or location of measurements only after reviewing the risks and making any changes to the precise data necessary to minimize the risk of individual volunteers being identified from the de-identified data.

Data Consumption

By default, the Application is configured to use no more than 100 megabytes (MB) of data each month for automated testing. Any additional tests you initiate will also consume between 100 kilobytes (kB) for the smallest packet loss test and 40 MB for a complete suite of tests. At any time, you can change the amount of data the Application uses by increasing or reducing the "monthly data cap" in the settings menu. You should ensure that your monthly broadband usage, including the data used by the Application and Service during the Program, does not exceed the data allowance included in your mobile broadband package. Your carrier may charge you for any data used in excess of your package's data allowance.

Stopping Your Participation

If at any time you want to stop participating in this program, simply delete the Application and the accompanying data, and the Application will not collect further data. Any data collected prior to the deletion of the Application will remain in the data sets subject to the above terms and protections.

● **Other terms of use of the Application**
  ○ Detailed Testing Architecture and Technical Summary (https://www.fcc.gov/measuring-broadband-america/mobile/technical-summary.html)
  ○ Program Stakeholders' Code of Conduct (https://www.fcc.gov/measuring-broadband-america/mobile/code-of-conduct.html)
  ○ GNU Licensing Agreement (http://www.gnu.org/licenses/gpl-2.0.html)
  ○ This PRA Collection Authorized By OMB Control No. 3060-1139 (Expires 05/31/2017). Paperwork Reduction Act (PRA) of 1995 Notice (https://www.fcc.gov/measuring-broadband-america/mobile/pra-privacy-act-notice.html)
  ○ Mobile Provider Disclaimers (https://www.fcc.gov/measuring-broadband-america/mobile/mobile-provider-disclaimers.html)

- ○ Application Terms and Conditions and Privacy Notice (https://www.fcc.gov/measuring-broadband-america/mobile/mobile-terms-privacy-notice.html)
- ○ Mobile broadband providers endorse some applications certified for use on their network. This application may not have be formally certified by your provider or carry a formal endorsement from your mobile broadband network

[DOCUMENT ENDS]