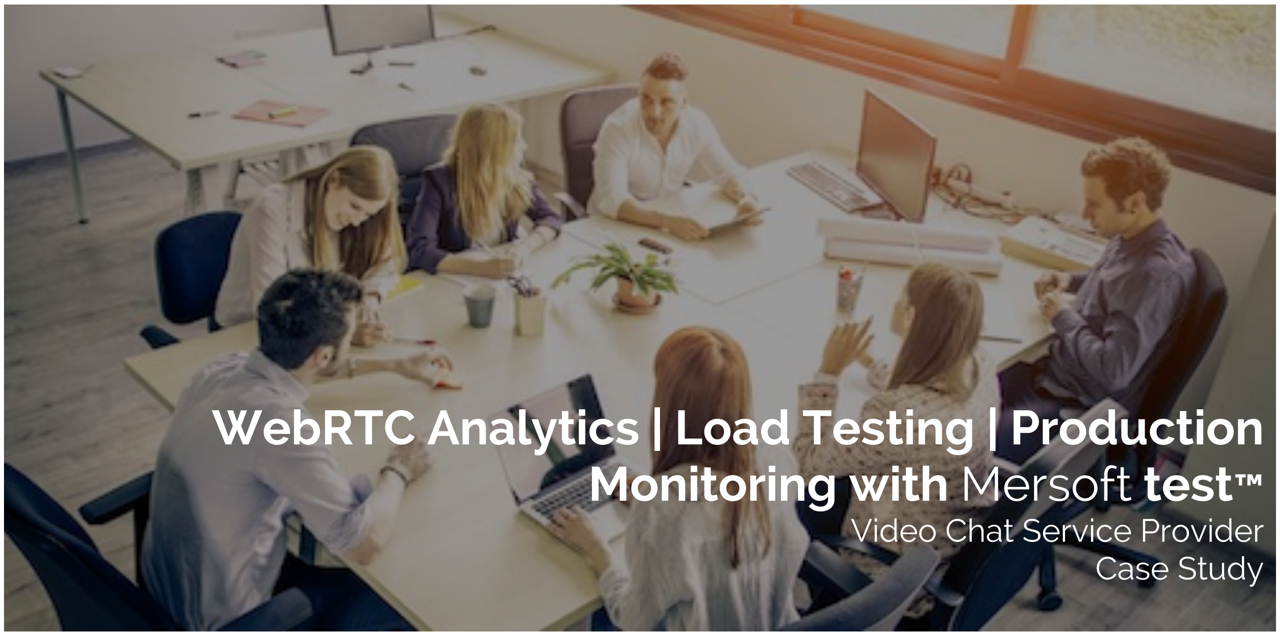


WebRTC Analytics | Load Testing | Production Monitoring with Mersoft **test™**

Video Chat Service Provider
Case Study

Executive Summary

A major U.S. communications provider worked with Mersoft on load testing and production monitoring of their video chat services that use WebRTC. The services include 2-way video, audio, text messaging and screen sharing. It is a fairly new product, but is gaining in popularity so they wanted to make sure it was ready for large-scale use.

Mersoft and the client collaborated to set up Mersoft test™ for production monitoring and several load tests to identify improvement areas. Several issues were found and resolved by the engineering team from node.js issues on the client, web server capacity issues, SDP exchange issues, ICE candidate issues, and problems when a selective forwarding unit was utilized versus a peer-to-peer connection. Each problem was resolved quickly and improved the customer experience.

All these tests were done by emulating the client experience so as new versions are released, the scripts can be easily rerun. The client now has the capability to manage the unique complexities of a large-scale WebRTC product and can save both time and labor in addition to getting insights unavailable from other products. In addition, all the graphs in this case study were taken directly from Mersoft test™. More are available to help your teams visualize the data effectively.

Case Study

The Mersoft test™ POC was executed over 90 days with the goal of determining if the product would serve as a valuable tool for production monitoring and load testing of a large WebRTC live chat platform. The project started focusing on production verification, later shifting to load testing.

Production monitoring , started with scripts first for browser-based testing. Mersoft test™ launches a VM with a browser and remotely controls the browser to perform the test and collect stats. The browser script starts a minimum of two virtual clients, one in each of the different regions (east and west U.S.). The script designates a VM as the caller and the other as a callee. The caller would proceed to use the web client to create a call to the callee.

During the period, Mersoft test™ performed production monitoring and load testing for a total of 4,000 test hours, running 5,600 total tests with a maximum of 1,000 clients on a single load test run.

After a test is performed, Mersoft test™ provides analysis of each endpoint and connection, and determine the status of each item. It provides:

Endpoint Analysis:
- Packet Loss
- High Jitter
- Not receiving video
- Not sending video

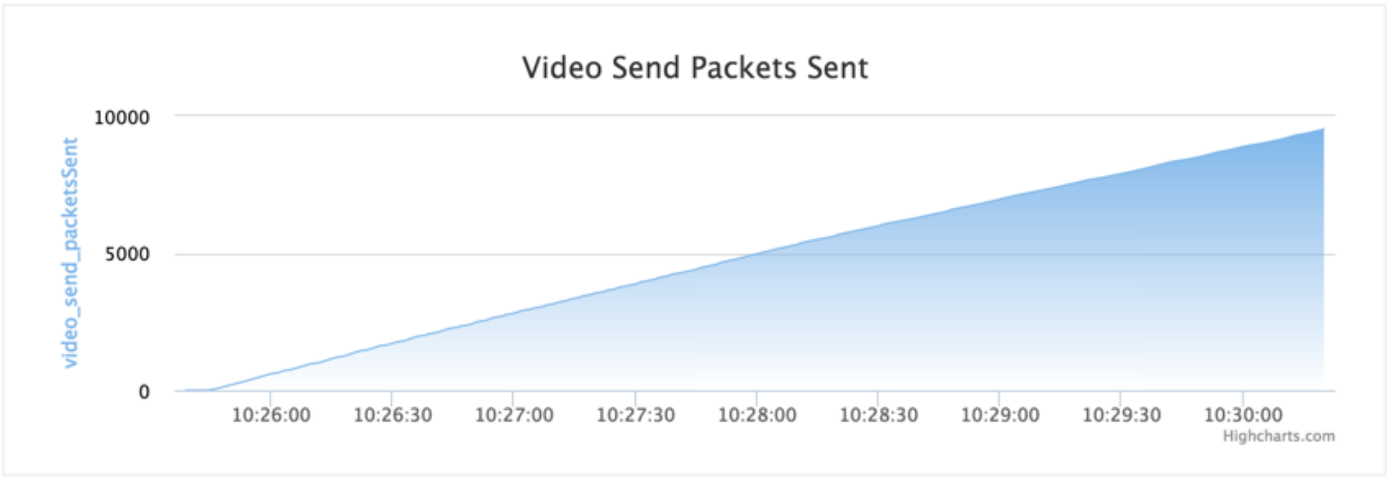
Test Analysis:
- Number of successful connections

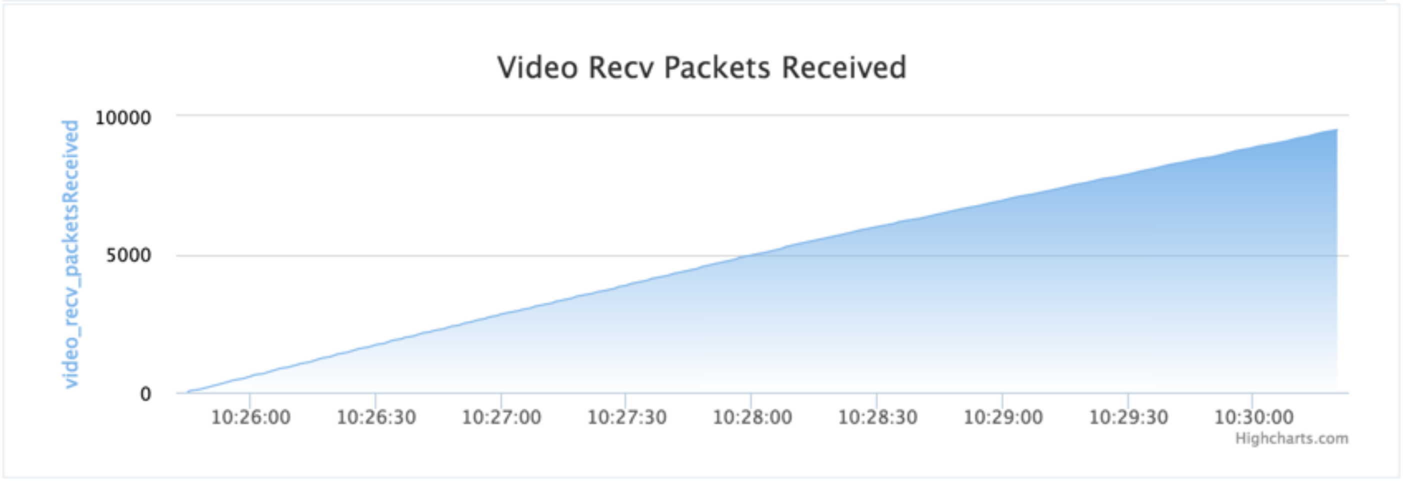
Connection Analysis:
- Send/receive ratio between endpoints on a connection
- Endpoint failures on the connection
- Call request sent
- Call request accepted
- Connection Status
- Post-dial delays

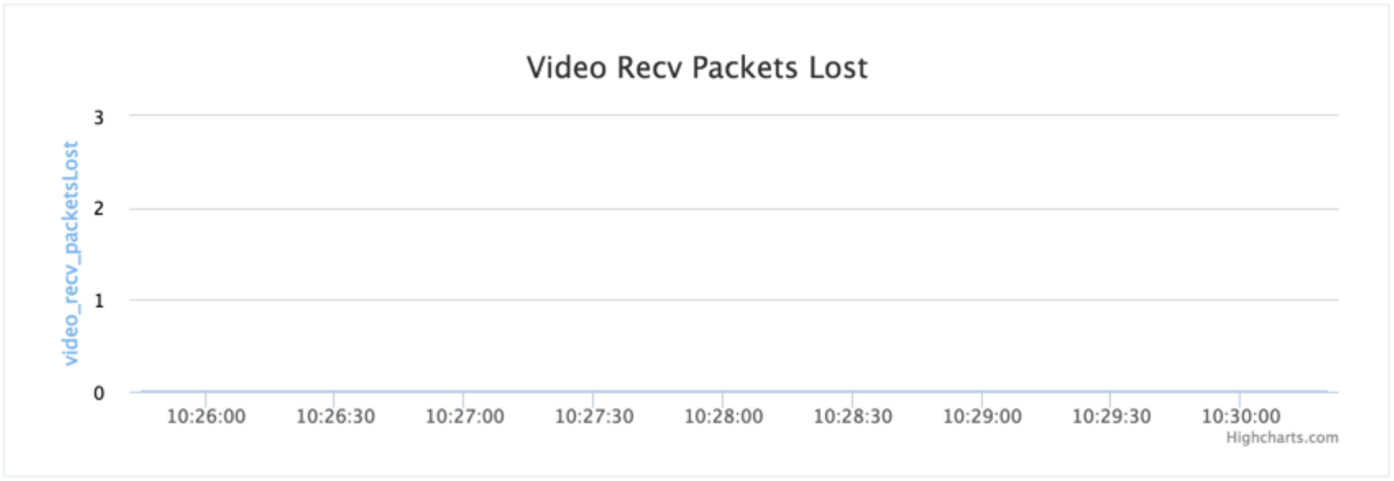
© 2017 MERSOFT CORPORATION ALL RIGHTS RESERVED

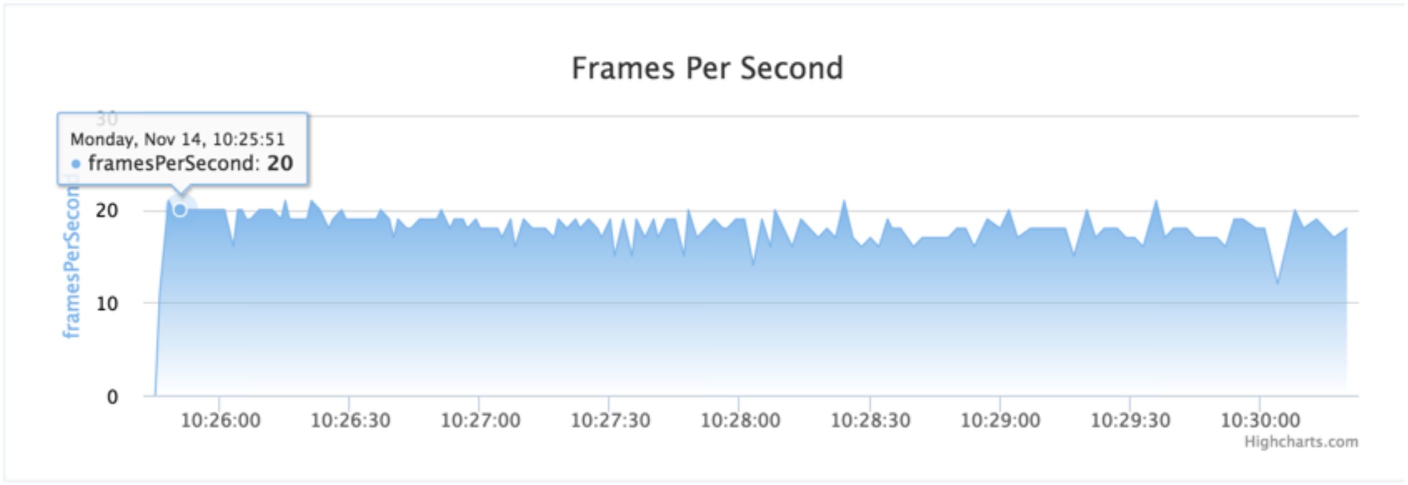
Endpoint stats:

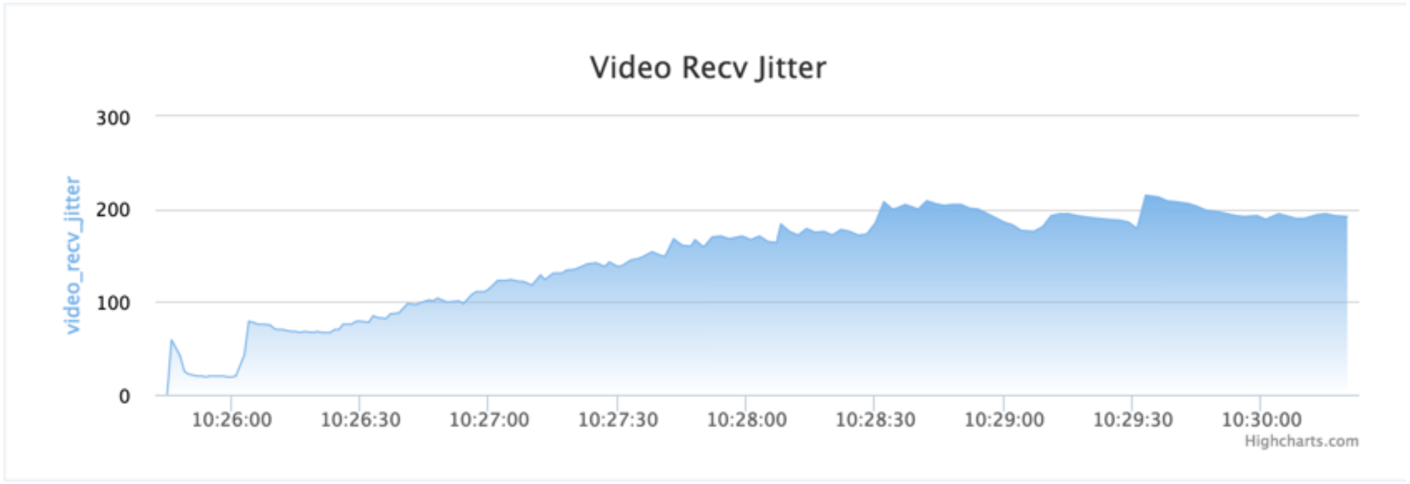
Mersoft test™ can see the stats in a test either by averaging across all clients in a test or for a specific client. The statics are stored and graphed. The graphics can be viewed in real time or after it's finished. The following are sample graphs for the endpoints.

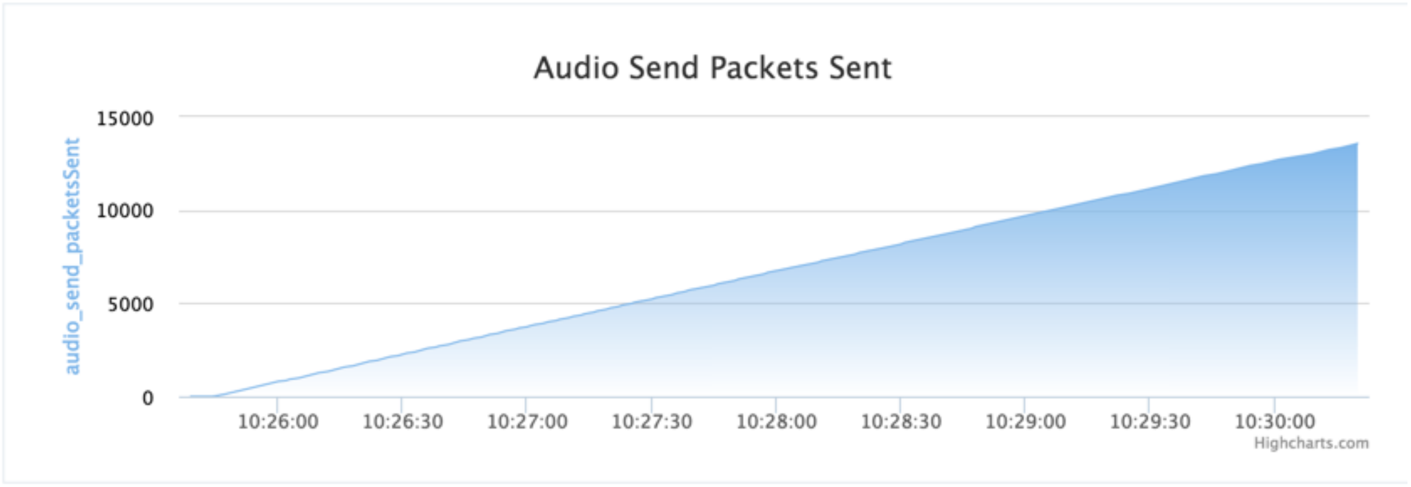


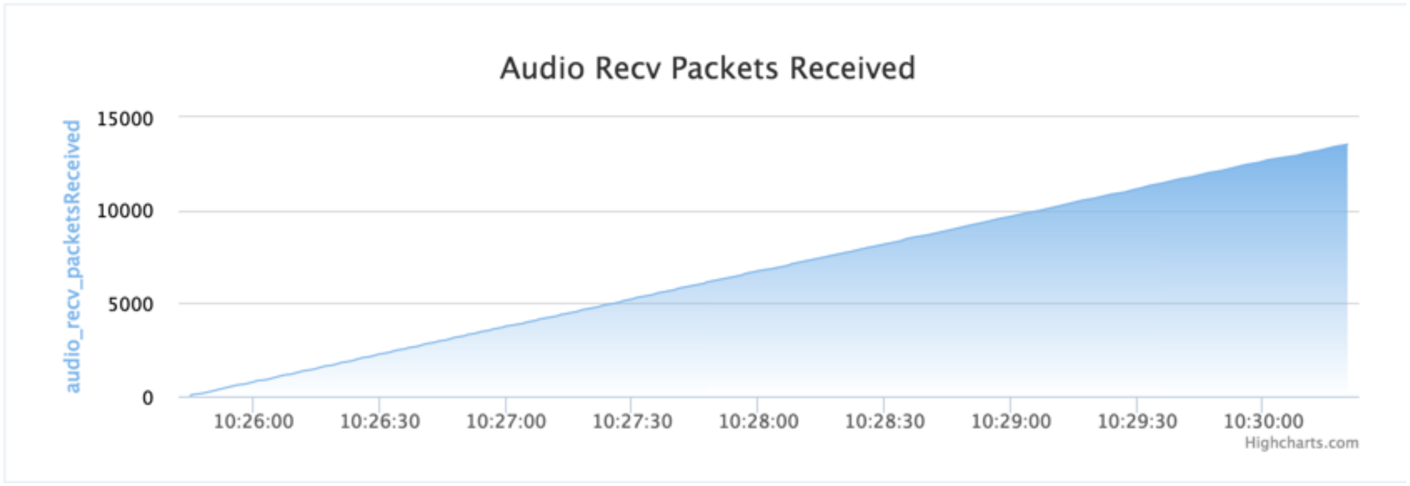


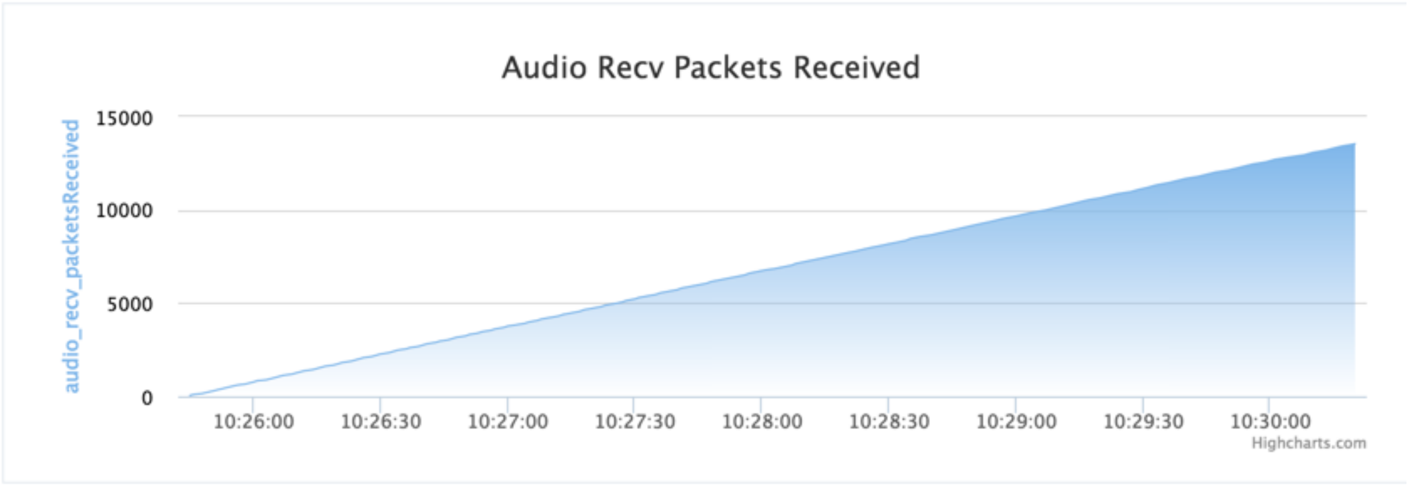


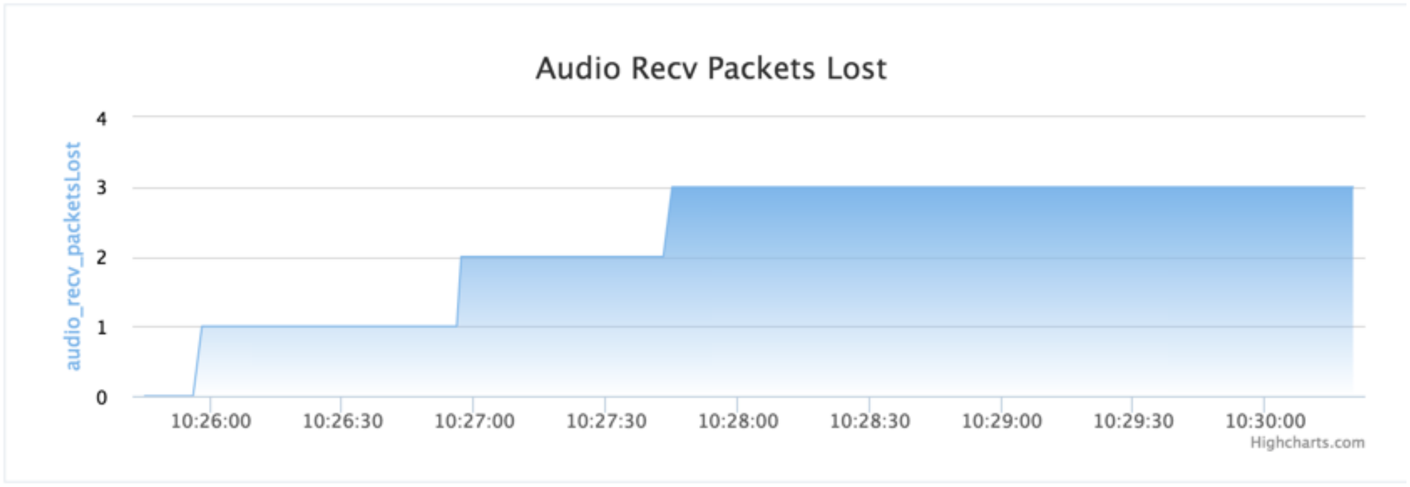

Frames Per Second
Monday, Nov 14, 10:25:51
● framesPerSecond: 20
framesPerSecond
30
20
10
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com

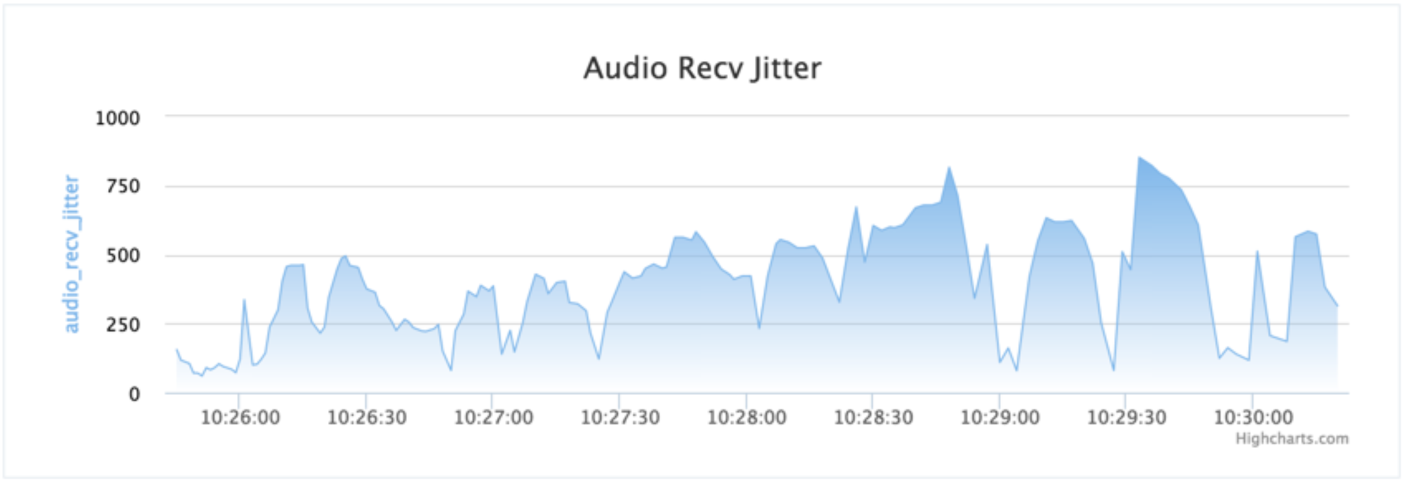

Video Recv Jitter
video_recv_jitter
300
200
100
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com


Audio Send Packets Sent
audio_send_packetsSent
15000
10000
5000
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com


Audio Recv Packets Received
audio_recv_packetsReceived
15000
10000
5000
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com

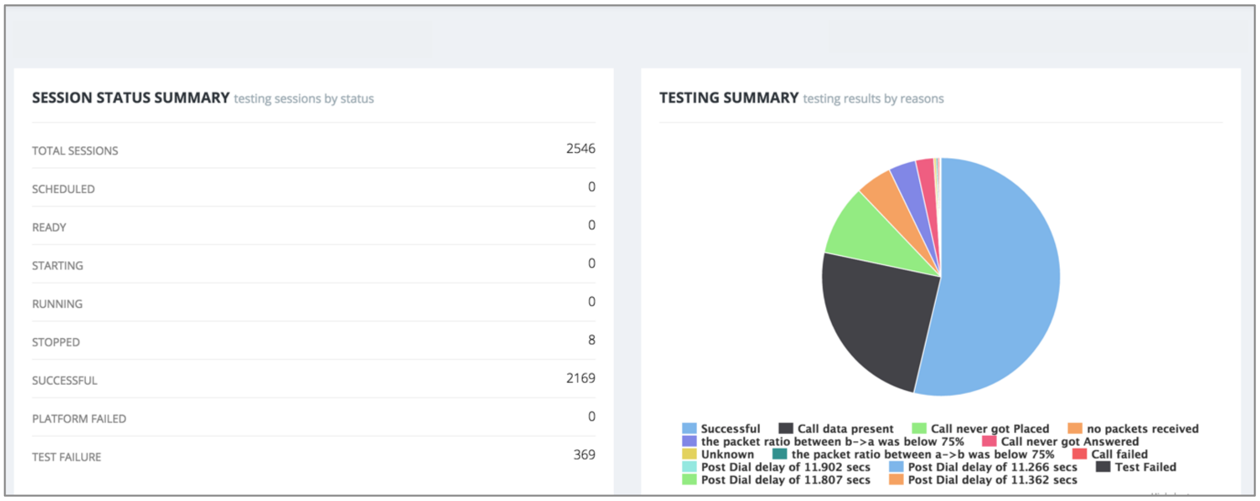

Audio Recv Packets Received
audio_recv_packetsReceived
15000
10000
5000
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com


Audio Recv Packets Lost
audio_recv_packetsLost
4
3
2
1
0
10:26:00
10:26:30
10:27:00
10:27:30
10:28:00
10:28:30
10:29:00
10:29:30
10:30:00
Highcharts.com



Along with the graphs, other information was gathered on each call. Such as call request time, call accept time and video start time. Custom information can be stored with the status, a customer example was the “Trace Id” that helped them troubleshoot.

Test Stats can be displayed by type, environment. They can also be filtered by a time range.



SESSION STATUS SUMMARY testing sessions by status

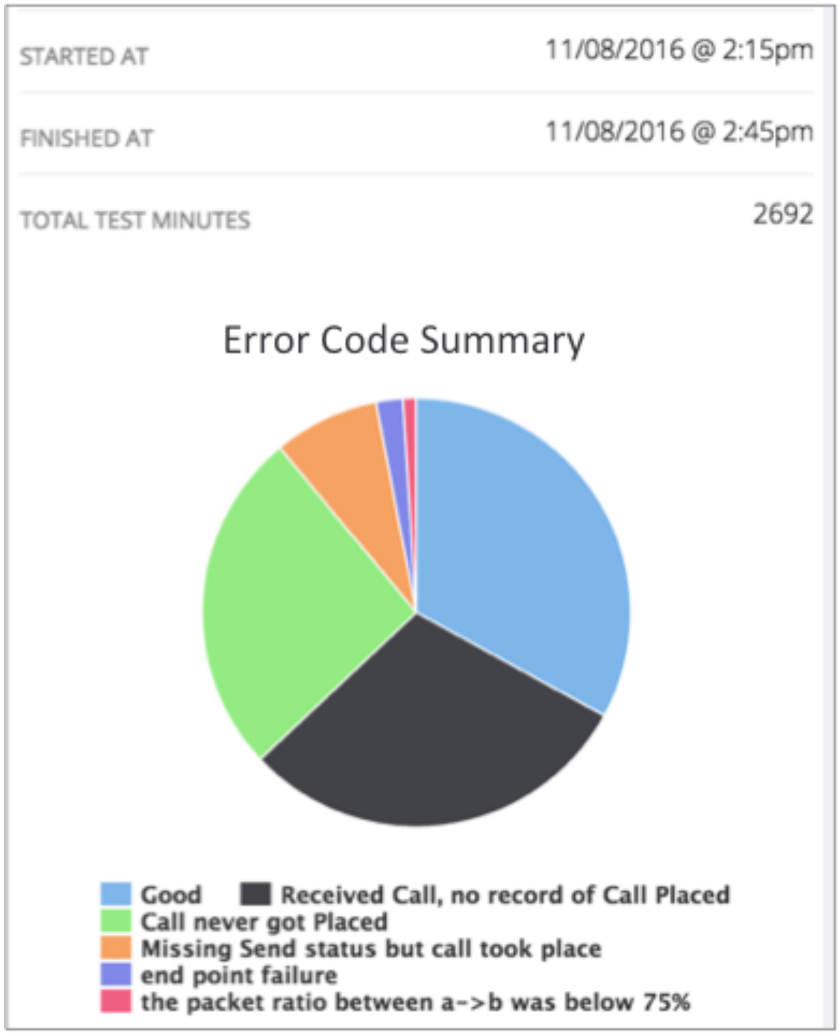
TOTAL SESSIONS	2546
SCHEDULED	0
READY	0
STARTING	0
RUNNING	0
STOPPED	8
SUCCESSFUL	2169
PLATFORM FAILED	0
TEST FAILURE	369

TESTING SUMMARY testing results by reasons

Successful
Call data present
Call never got Placed
no packets received
the packet ratio between b->a was below 75%
Call never got Answered
Unknown
the packet ratio between a->b was below 75%
Call failed
Post Dial delay of 11.902 secs
Post Dial delay of 11.266 secs
Test Failed
Post Dial delay of 11.807 secs
Post Dial delay of 11.362 secs

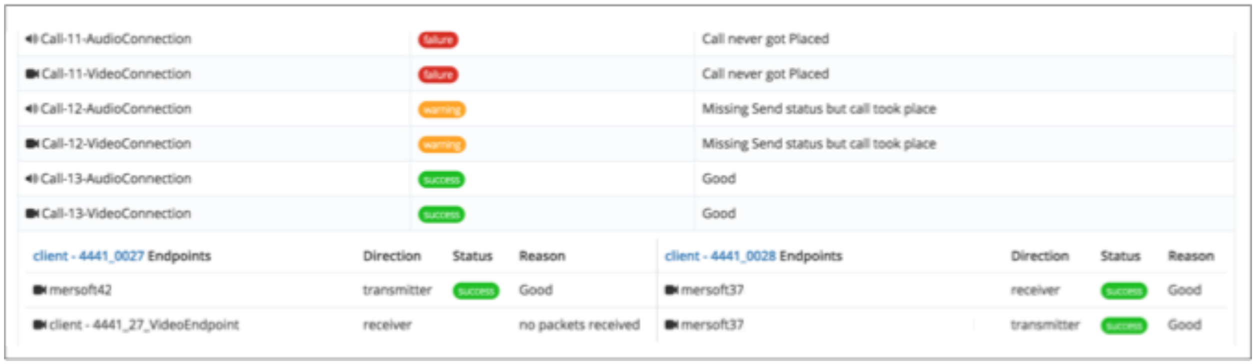
This is good for watching production. The filtering allows us to drill down on the graph for test results. For load testing, the test itself contains the multiple connections representing calls. The connection stats are analyzed inside the test itself. Mersoft `test`™ uploads a log file for each vClient at the end of a test containing the entire JavaScript console from the test for debug and further analysis.

When analyzing a load test, the connections are viewed within the structure of the test. A graph displays the resulting status for a connection.



STARTED AT	11/08/2016 @ 2:15pm
FINISHED AT	11/08/2016 @ 2:45pm
TOTAL TEST MINUTES	2692

Users can also view details on each connection with a drill down to the endpoints:



Call-11-AudioConnection	failure	Call never got Placed
Call-11-VideoConnection	failure	Call never got Placed
Call-12-AudioConnection	warning	Missing Send status but call took place
Call-12-VideoConnection	warning	Missing Send status but call took place
Call-13-AudioConnection	success	Good
Call-13-VideoConnection	success	Good

client - 4441_0027 Endpoints	Direction	Status	Reason
mersoft42	transmitter	success	Good
client - 4441_27_VideoEndpoint	receiver		no packets received

client - 4441_0028 Endpoints	Direction	Status	Reason
mersoft37	receiver	success	Good
mersoft37	transmitter	success	Good

Goals

Mersoft **test**™ had two goals for this client: production verification testing (synthetic monitoring) and load testing. The project started focusing on production verification, later shifting to load testing.

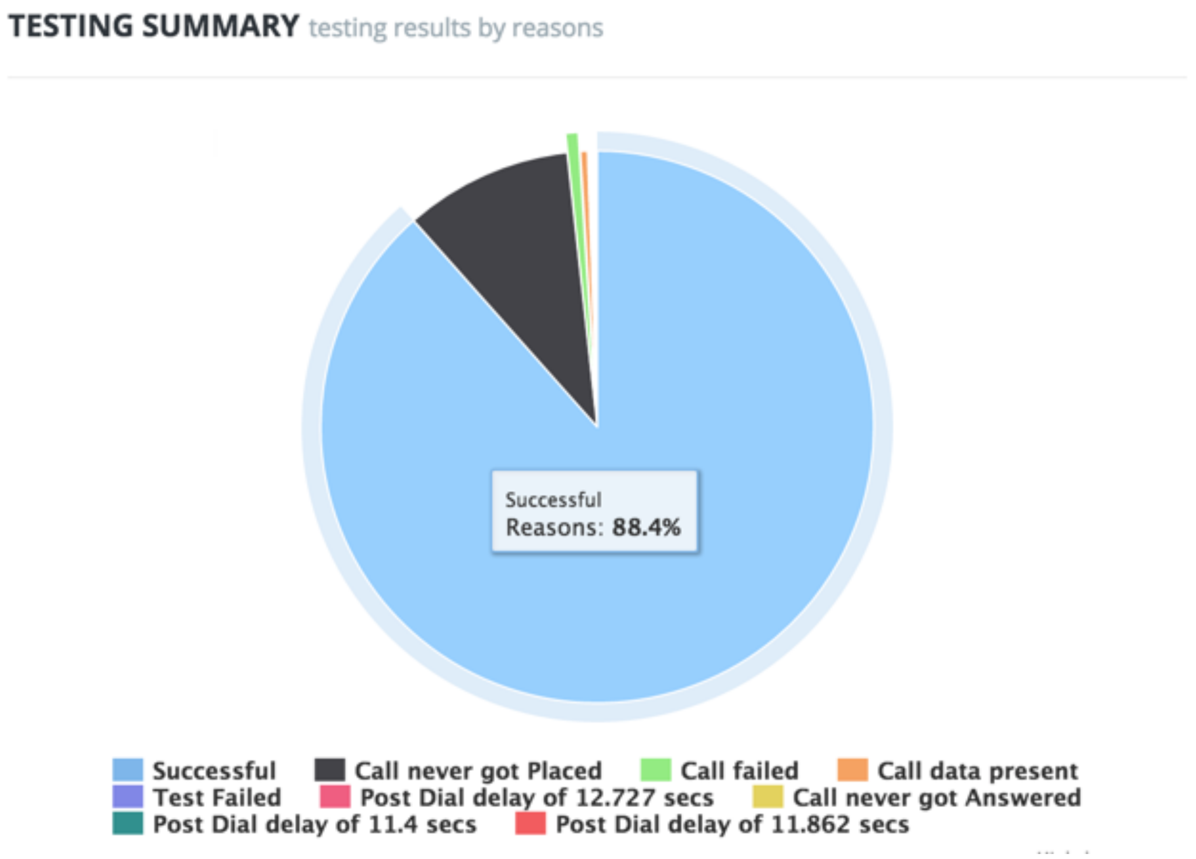
Production Monitoring

Most of the time, Mersoft **test**™ was running two monitoring tests, one for peer-to-peer connections and one using the client's SFU (Selective forwarding unit). The system was configured to send an alert to a Slack channel if there were 3 consecutive failures. The system proved to be stable and this alert only triggered at the beginning.

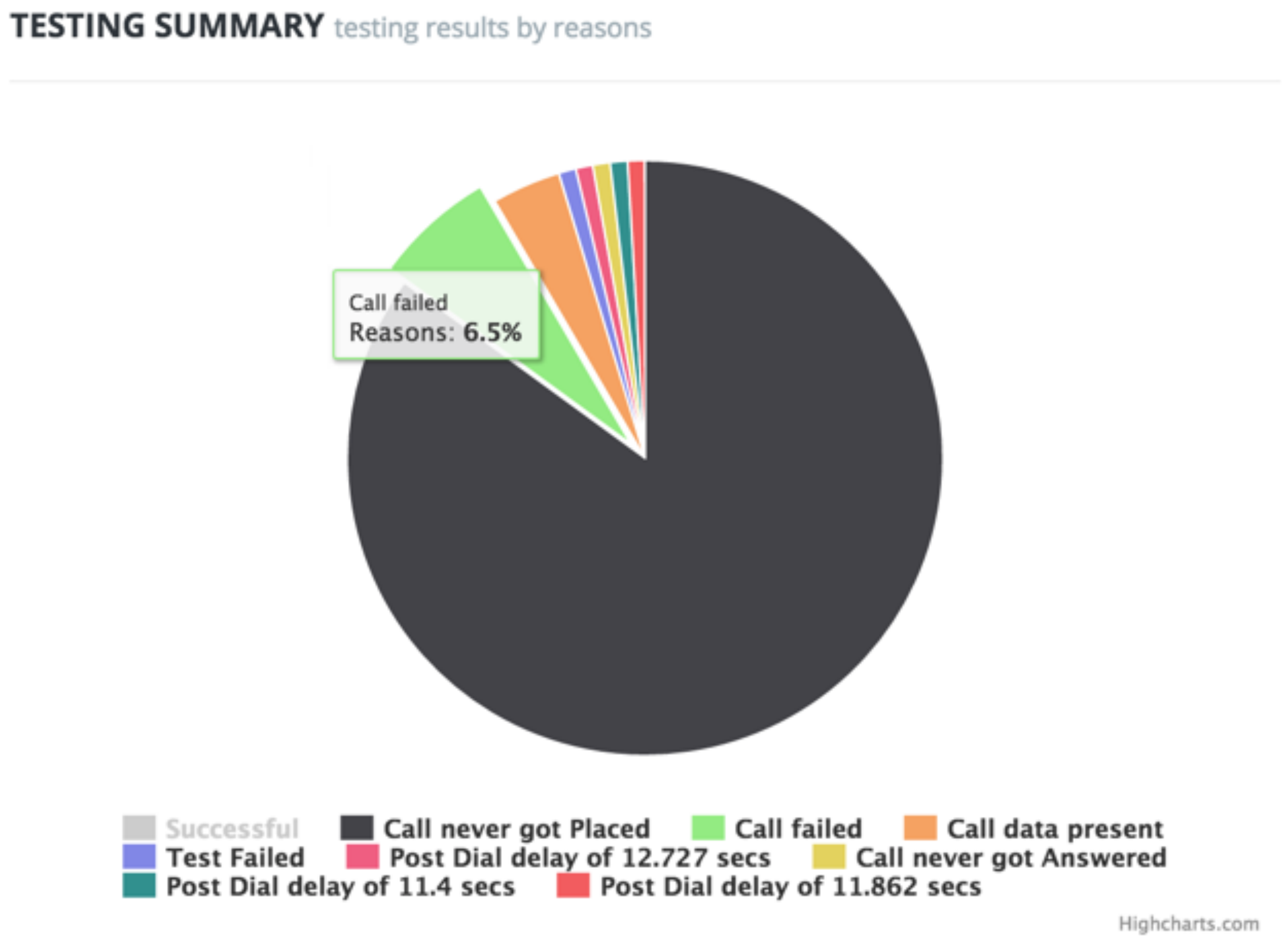
The production verification test repeated every 15 minutes, performing 5 minutes of WebRTC testing each time. It took 10 minutes to create the VMs and prepare the test, along with uploading the results. Loading VMs in different regions had a significant effect on start times. The west region startup time was 3 to 5 minutes whereas the east was 6 to 9 minutes. Mersoft **test**™ will sync the two clients before starting the test so these startup time differences did not affect the test performance.

Peer-to-Peer Tests

Mersoft **test**™ executed about 1,300 tests with an 88.4% success rate.



Drilling down on the 11.6% that failed shows the following graph:



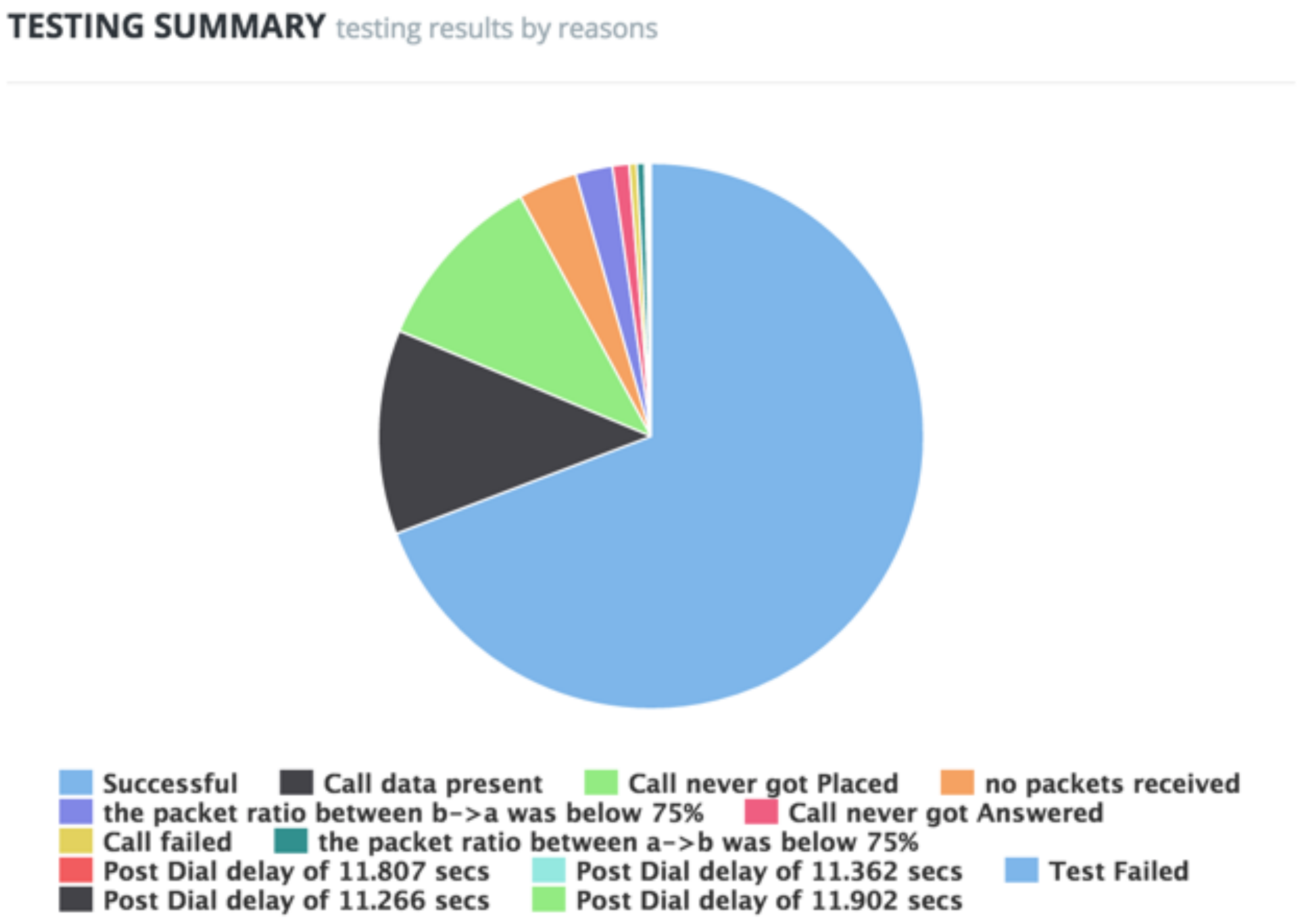
The "calls never got placed" results were usually when the web page did not load correctly and the test failed at login. "Call Failed" usually occurred when both clients loaded the webpage but the call was never placed or received. Other causes could be the WebRTC or ICE candidates never fully negotiated. NOTE: The 6.5% represents a percentage of the failures, not of total calls. "Call Data Present" represents the endpoints all having successful call data, so the call was successful, just the call offer and accept were either missing or didn't compare correctly, meaning these calls were successful, however the analysis could not verify if the signaling was correct.

We can conclude that a large majority of issues noticed as failures were caused by components for delivering the calls as a web page. Errors in the logs point to the page loads.

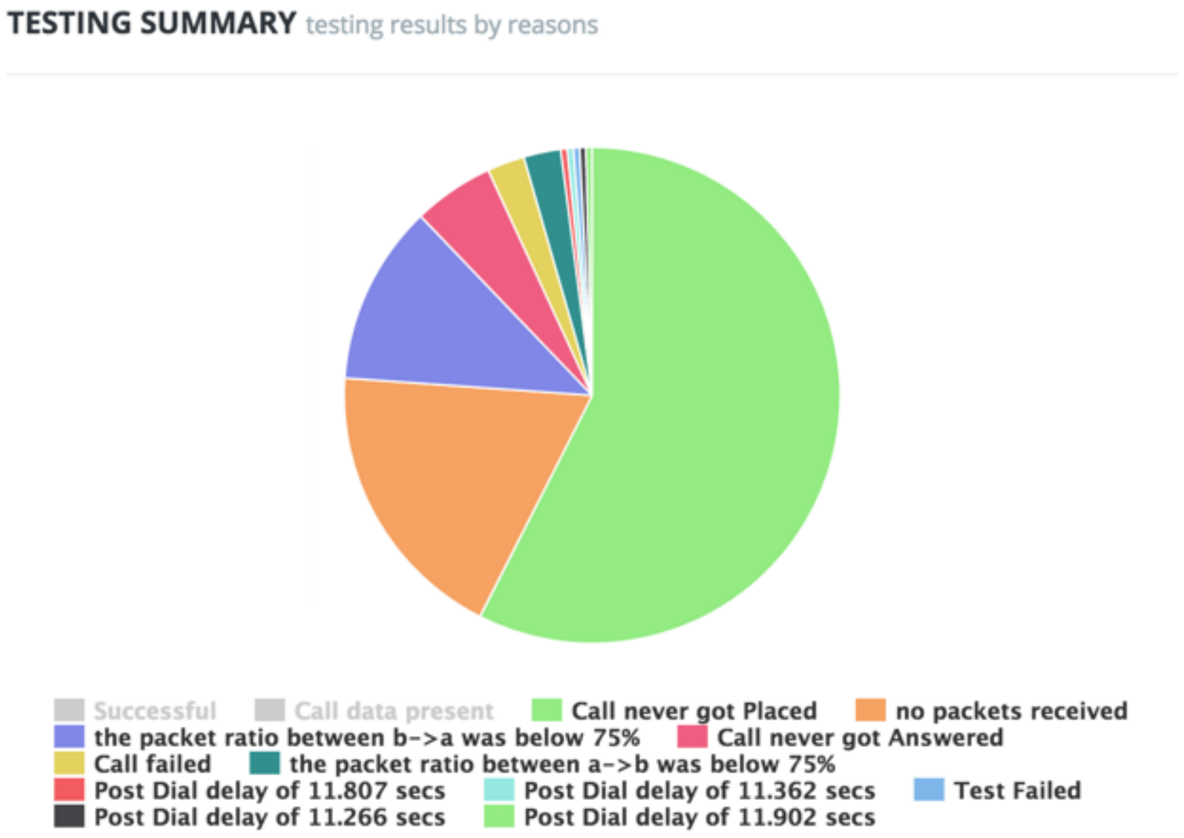
Selective Forwarding Unit (SFU) tests:

For the same time frame, tests had a 69.2% success rate when monitoring production with the SFU. One could conclude that those errors in the "Call data present" category were actually successful. Even though Mersoft test™ found errors in the signaling, there was data sent meaning that the signaling was ultimately successful. Including those calls would raise the success rate to 79.9%. Errors in time stamps or perhaps lost statistics could be drivers of this number.

The SFU call success rate differs from the peer-to-peer calls because there is a bridge that is relaying all the packets between the two clients. For the same timeframe with a similar number of tests, the success rate drops from 88.4% for the P2P vs. 79.9% for the SFU.



The largest issue in both P2P and SFU was "Call Never Got Placed." Both were a similar magnitude: SFU= 9.8%, P2P=10.2%. The SFU test revealed additional issues as well as seen in the graph below.

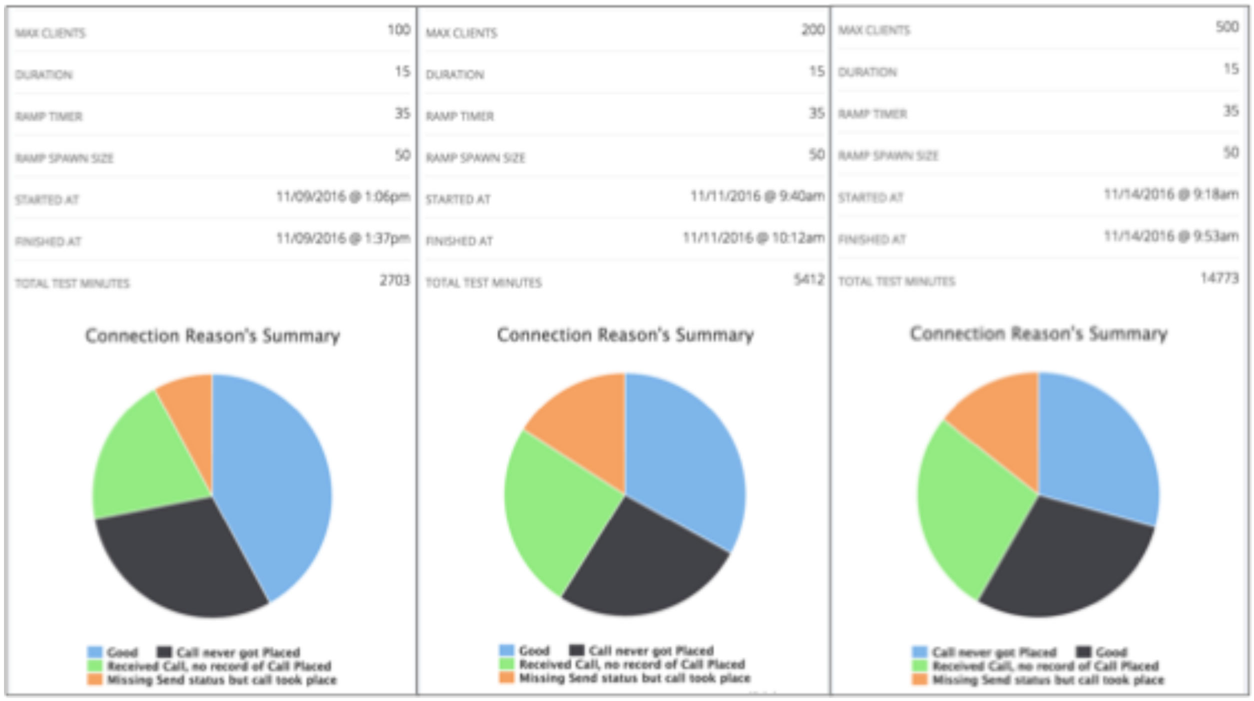


Categories not noticeable in the Peer-to-Peer but more prevalent here are "no packets received" and "packet ratio". The "no packets received" test are where the signaling offered a call and it was accepted, video was sent, however no packets were received. The "packet ratio" is where more that 25% of the send packets were not received.

Load Testing analysis:

For load testing, the project scope called for using the same script used for production verification as a way to see how versatile the scripting could be.

First, we placed multiple calls in the same second which overloaded the web server supporting the client app. So as the load increased with successive tests up to 1,000 clients, the rate of connection failures remained generally consistent because the web server maxed-out.



MAX CLIENTS	100
DURATION	15
RAMP TIMER	35
RAMP SPAWN SIZE	50
STARTED AT	11/09/2016 @ 1:06pm
FINISHED AT	11/09/2016 @ 1:37pm
TOTAL TEST MINUTES	2703

MAX CLIENTS	200
DURATION	15
RAMP TIMER	35
RAMP SPAWN SIZE	50
STARTED AT	11/11/2016 @ 9:40am
FINISHED AT	11/11/2016 @ 10:12am
TOTAL TEST MINUTES	5412

MAX CLIENTS	500
DURATION	15
RAMP TIMER	35
RAMP SPAWN SIZE	50
STARTED AT	11/14/2016 @ 9:18am
FINISHED AT	11/14/2016 @ 9:53am
TOTAL TEST MINUTES	14773

Connection Reason's Summary
Good
Call never got Placed
Received Call, no record of Call Placed
Missing Send status but call took place

Therefore, to generate a constant load, we created 3-4 times more calls to properly load the WebRTC live chat platform but it prevented the team from identifying the maximum load the WebRTC live chat platform could support. After the capacity of the web server is expanded, tests can be rerun to fully load it and determine its maximum capacity.

Results

The goal of this exercise was not only to establish production monitoring and load testing process for the client, but to also enable the DevOps team to modify the scripts as their needs change and for use with other products. The team identified several valuable insights that their previous tool set did not uncover. They identified real problems and implemented improvements. Since they are a quick team, several enhancements were made during the project and more followed.

This type of deep-dive insight is available with Mersoft **test**™ for all kinds of WebRTC solutions. Mersoft is helping teams solve problems in video conferencing, home security, enterprise security, live media broadcasting and more.

Contact Mersoft today to get more information about how we could help you. Talk to one of our engineers or see a demo. Increase customer satisfaction with your WebRTC products with Mersoft **test**™. Email: info@mersoft.com or call 913-871-6210.