

**Video Streaming Airtime Utilization Analysis**

**Test Plan**

*June 5, 2018*

**CHANGE REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Section Revised** | **Revision Level** | **Revision**  **Date** |
| HPE Aruba | Initial draft | 1.0 | 06/05/18 |

**TABLE OF CONTENTS**

[1 OBJECTIVE 3](#_Toc516003857)

[1.1 Summary 3](#_Toc516003858)

[1.2 Procedure summary 3](#_Toc516003859)

[2 TESTBED SETUP 3](#_Toc516003860)

[2.1 Baselining test environment and validating test procedures 3](#_Toc516003861)

[2.1.1 Verifying clean channel and high MCS rates 3](#_Toc516003862)

[2.1.2 Validation of packet capture quality 4](#_Toc516003863)

[2.2 Network Topology 5](#_Toc516003864)

[2.3 Equipment 5](#_Toc516003865)

[2.4 BSS Configuration 6](#_Toc516003866)

[2.5 Video Source and Server Software 6](#_Toc516003867)

[3 TEST PROCEDURE 7](#_Toc516003868)

[3.1 Test Cases 8](#_Toc516003869)

[3.2 Sniffer Capture Method 8](#_Toc516003870)

[3.3 Test Procedure 9](#_Toc516003871)

[3.4 Wireshark Data Extraction Method 11](#_Toc516003872)

[3.5 Airtime utilization calculation 12](#_Toc516003873)

[4 TEST Results 14](#_Toc516003874)

[5 considerations for greenfield 802.11AX in 6 ghz 14](#_Toc516003875)

**INDEX OF FIGURES**

[Figure 1: Omnipeek custom plugin to show the MCS histogram for the data 4](#_Toc516003876)

[Figure 2: Network Topology - known file on local server 5](#_Toc516003877)

[Figure 3: Properties of video file including bitrate and resolution 7](#_Toc516003878)

[Figure 4: Snapshot of a video file used in the test 7](#_Toc516003879)

[Figure 5: Wireless Diagnostics user interface 8](#_Toc516003880)

[Figure 6: VLC Streaming Menu Screenshot 9](#_Toc516003881)

[Figure 7: VLC Streaming Screenshot with configurations 10](#_Toc516003882)

[Figure 8: VLC Screenshots to show VLC Client setup 10](#_Toc516003883)

[Figure 9: Wireshark Column Preferences 11](#_Toc516003884)

[Figure 10: Wireshark columns 12](#_Toc516003885)

[Figure 11: CSV export 12](#_Toc516003886)

[Figure 12: Corrected CSV file 13](#_Toc516003887)

[Figure 13: MPDU duration 13](#_Toc516003888)

[Figure 14: Airtime utilization computation 13](#_Toc516003889)

# OBJECTIVE

## Summary

This contribution provides over-the-air (OTA) test procedure of a major video display resolution (1080P) from a local video source with an encoding bitrate of 4.5 Mbps.

## Procedure summary

Video data is streamed to the clients over an 80MHz clean channel and the packet capture is used to calculate the airtime utilized for different test cases. Details on the test methodology, capture procedure and data analysis procedure will be provided. We ran tests for one client and three clients in order to validate that calculated airtime scales as expected.

The test procedure can be summarized as follows:

1. Prepare video source for streaming
2. Start capture on packet capture machine
3. Start streaming on all the client laptops simultaneously
4. Stop capture and collect data
5. Export data in table form and run airtime utilization calculations

# TESTBED SETUP

The following sections describe the testbed setup along with details on the network topology, devices and software applications used.

## Baselining test environment and validating test procedures

### Verifying clean channel and high MCS rates

Below are the steps taken to ensure that the channel used is clean.

1. Use an isolated location or a RF shielded room to run the tests
2. Run spectrum analyzer on the channel and check for signals in the chosen test channel
3. Ensure all APs apart from the test AP are on a different channel with sufficient spatial separation or switched off.

Before beginning testing we ran a series of tests for a few clients and selected ones which showed a high MCS rate. Below is a histogram of the MCS rates for a sample video stream. We note that the client is capable of consistently using MCS 9 for streaming data.

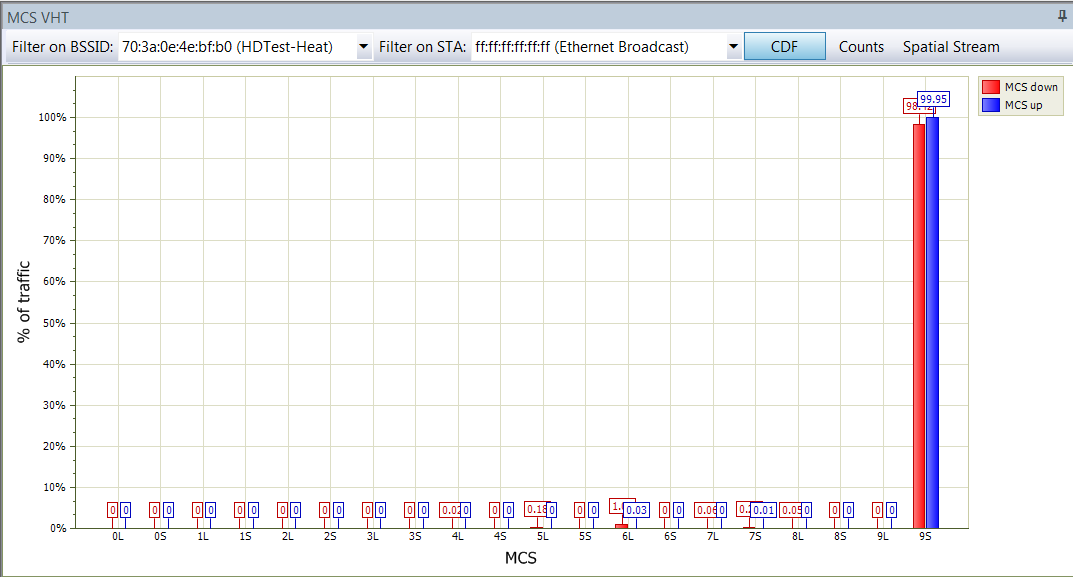


Figure 1: Omnipeek custom plugin to show the MCS histogram for the data

### Validation of packet capture quality

It is important to validate that the sniffer has captured close to 100% of the packets on air. The following steps provide a method to validate the packet capture. Because the 802.11 data packets are transmitted at much higher data rate than control and management frames, it is possible that some could be missed by the sniffer. Therefore the steps below validate the integrity of the data packet capture.

1. Load the packet capture file in Wireshark, and export all the packets to CSV format. Make sure the following fields are selected: “Source”, “Destination”, “Transmitter address”, “Receiver address”, “Type/Subtype”, and “Sequence number”.
2. To inspect the capture quality for downstream data packets, open the CSV file in Excel and do the below filtering:
   * “Source”: Select the IP address of the video server
   * “Destination”: Select the IP address of the wireless client
   * “Transmitter address”: Select the wireless MAC address of the AP
   * “Receiver address”: Select the wireless MAC address of the client
   * “Type/Subtype”: Select “QoS Data”
   * “Sequence number”: Deselect “Blanks”
3. For upstream data packets, just swap the “Source” and “Destination”, and swap the “Transmitter address” and “Receiver address” when filtering.
4. After filtering, copy the column “Sequence number” to a new spreadsheet. Then count the gaps in 802.11 sequence number, and add up all the gaps.

Here is an example, for packet number N the sequence number is X. And for packet number N+1 the sequence number is X+Y. There are a few cases:

* Y= -4095, this means the sequence number of packet N is 4095, and that of packet N+1 is 0. This is not a gap since the 802.11 sequence numbers are reused every 4096 frames
* -4095 < Y <= 0, this means there is a retransmission, and this is not a gap
* Y = 1, there is no gap
* Y > 1, there is a gap in sequence number
* |Y| is a very large value, e.g. if |Y|>4000, check the packet capture, this could be retransmission or malformed packet with inaccurate information

1. Then add up all (Y-1)’s, and this is the total number of missed packets in the packet capture. The quality of the packet capture is verified through the percentage of captured packets, e.g.:

is the percentage of captured data packets, and is the count of captured data packets (802.11 MPDU’s).

## Network Topology

The following is the network topology for the test.

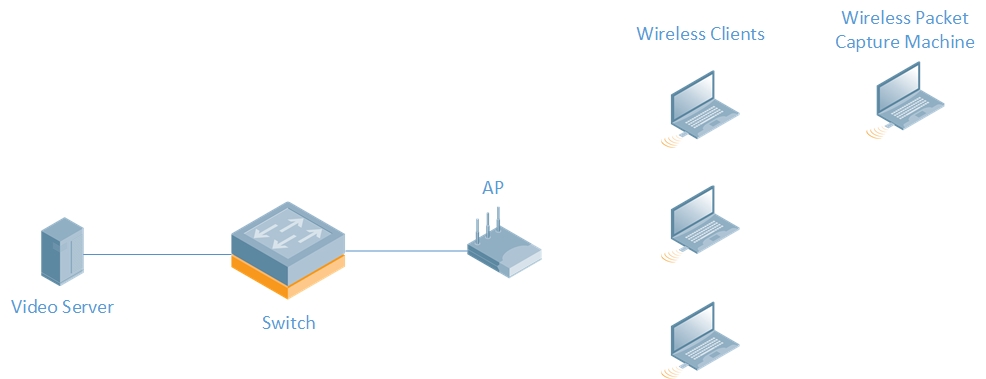


Figure 2: Network Topology - known file on local server

Ensure that the client laptops are blocked from the internet in order to avoid any other traffic being generated apart from the streaming application.

## Equipment

The following table details out the hardware equipment used for the test. Going forward in this document the “Device Label” will be used to refer to a particular device.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Device Label | Function | Model | Number of Streams | OS/Firmware |
| Client | Receives streaming data | Vendor A Laptop | 2x2 11ac | Windows 10 |
| AP | Wi-Fi access point | Aruba AP 335 | 4x4 11ac | Aruba OS 6.5.4.0 |
| Sniffer | Wireless Packet Capture Machine | MacBook Pro 2013 | 3x3 11ac | OSX – Mavericks – 10.9.5 |

Table 1: Equipment

|  |  |  |
| --- | --- | --- |
| Software | Function | Version |
| VLC media player | Video Server and Client | 3.0.2 |
| Wireshark | Packet Capture Analysis | 2.4.6 |

Table 2: Software List

## BSS Configuration

|  |  |
| --- | --- |
| Channel | 80 MHz, primary channel 36 |
| EIRP | 18 dBm |
| Basic Rate | 6 Mbps |
| Beacon Rate | 6 Mbps |
| TxBF | Disabled |
| MU-MIMO | Disabled |
| Encryption | Open |
| Rate Adaptation | Automatic |

Table 3: BSS Configuration

## Video Source and Server Software

With local video files as source we use three different video files with one file for each client. The three video files were generated using different 5 minute slices of the same video source file. This was done to generate three different action video clips.

The video is a first person view of a motorcycle driving down a road. The video format is set to ***MPEG-4*** with H.264 codec and resolution of 1080P. For each of the three 5 minute videos, a video file was created, encoded at 4.5 Mbps. The audio encoding was set to 192 Kbps with ***AAC*** format.

|  |  |
| --- | --- |
| Format | MPEG-4 |
| Codec | H.264 |
| Resolution | 1920 x 1080 |
| Frame Rate | 29 fps |
| Duration | 5 minutes |
| Video Bitrate | 4.5 Mbps |
| Audio Bitrate | 192 Kbps |

Table 4: Video Source Parameters

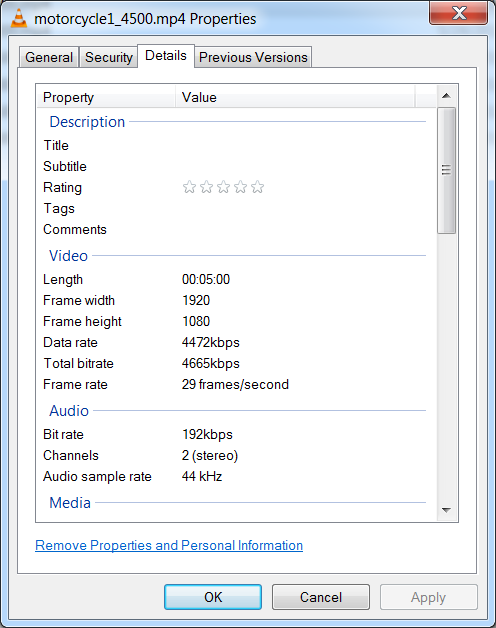


Figure 3: Properties of video file including bitrate and resolution



Figure 4: Snapshot of a video file used in the test

VLC media player is used to stream video from a local video server to the client laptops. The video server can be a Windows machine with VLC media player installed. VLC also needs to be installed on the client laptops.

# TEST PROCEDURE

The following section outlines the test cases and test procedures for different cases along with data collection methodology.

## Test Cases

The following table outlines the different test cases

|  |  |  |
| --- | --- | --- |
| Test Case ID | Video Bit Rate | Number of Clients |
| 1 | 4.5 Mbps | 1 |
| 2 | 4.5 Mbps | 3 |

Table 5: Test Cases

## Sniffer Capture Method

The following method should be used to capture packets over the air using a MacBook Pro as sniffer.

**Method**: Wireless Diagnostics Application

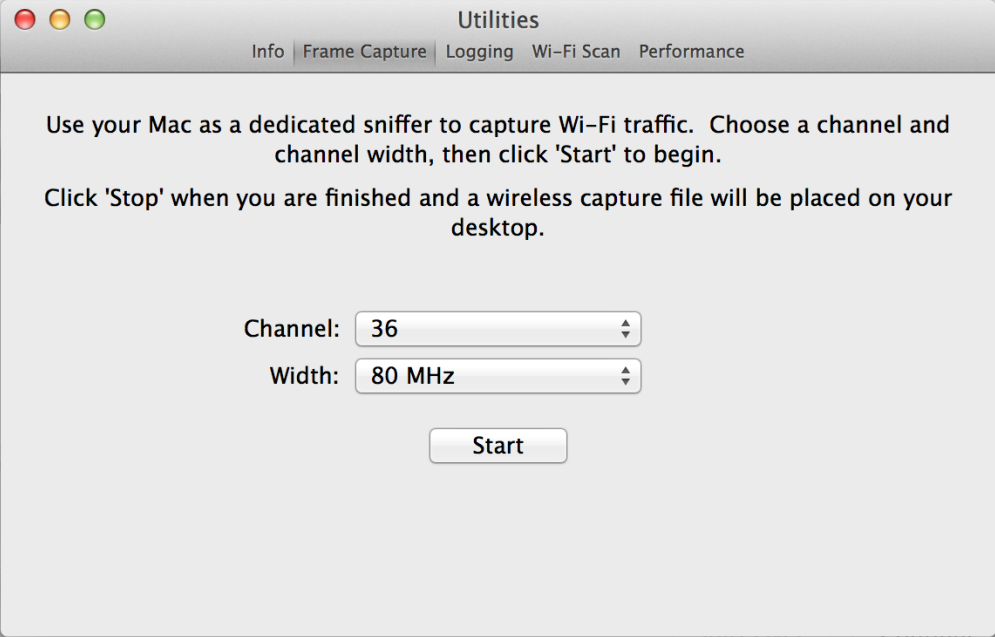


Figure 5: Wireless Diagnostics user interface

Wireless Diagnostics is an application in Mac OS with MacBook. It can capture Wi-Fi packets and generate a “\*.wcap” file in the file system. The capture file can be loaded in packet analyzing tools such as Wireshark or Omnipeek.

Refer to [this tutorial](http://osxdaily.com/2015/04/23/sniff-packet-capture-packet-trace-mac-os-x-wireless-diagnostics/)[[1]](#footnote-2) on how to capture packets with Wireless Diagnostics. If you are experiencing high packet loss with this method, try to connect the sniffer to another AP on the same channel and run the capture. This could improve the capture performance by fixing the sniffer to the targeted channel. Ensure that no other client is connected to this AP, and no traffic is going on between the sniffer and this AP, which will cause interference to the test AP and clients.

Please note that the test BSS should be an open network, as Wi-Fi encryption will impact the packet capture performance. According to [Apple Support Site](https://developer.apple.com/library/content/qa/qa1176/_index.html#//apple_ref/doc/uid/DTS10001707-CH1-SECWIFICAPTURE)[[2]](#footnote-3), “If the Wi-Fi network has a password, Wi-Fi encryption will make it much harder to examine the trace. To get around this, either temporarily turn off the Wi-Fi password on your network or use a separate test network that has no password.”

We recommend OS X Mavericks for the capture machine as the newer Mac OS versions may lose large amount of packets during the packet capture. We have verified the packet capture quality of the sniffer in OS X 10.9.5 and listed the steps in section 2.1.2.

## Test Procedure

Use the following procedure to collect the data:

1. Setup the VLC Video Server on a server machine on the network
   1. Go to VLC>Media>Stream and choose the necessary configurations

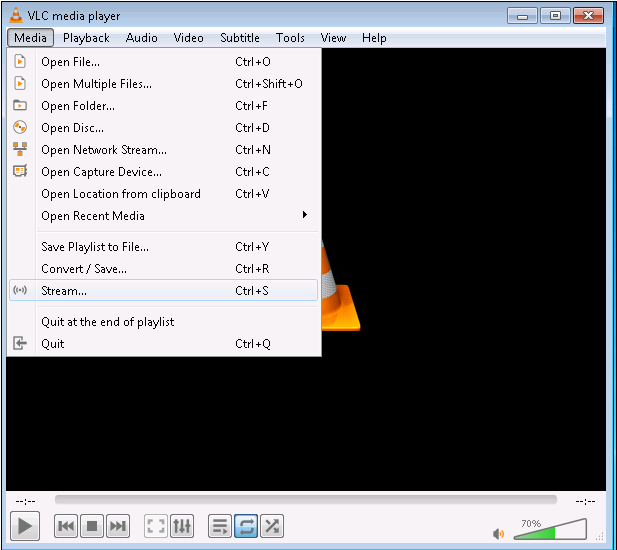


Figure 6: VLC Streaming Menu Screenshot

* 1. Below is an image of the final output after setup. Do not click on the Stream button yet. Please note, for a multi-client test, multiple VLC sessions need to be set up on the VLC server.

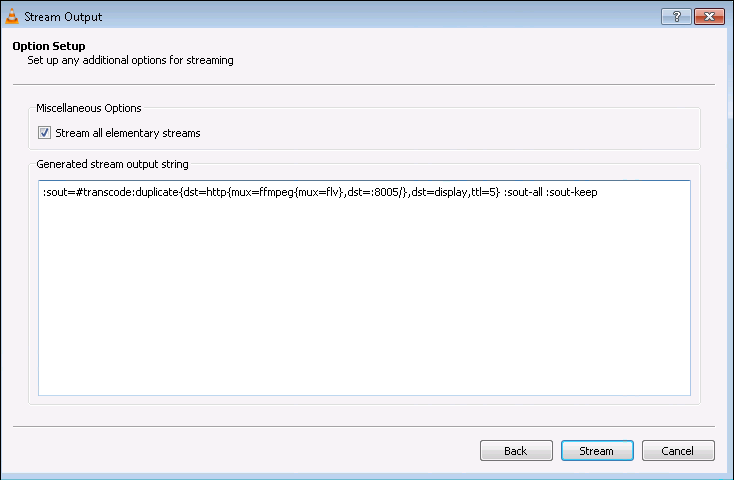


Figure 7: VLC Streaming Screenshot with configurations

**Note**: Detailed instructions can be found [here](http://letzgro.net/blog/how-to-use-vlc-as-a-live-streaming-server/)[[3]](#footnote-4)

1. Setup VLC Client on the client laptops

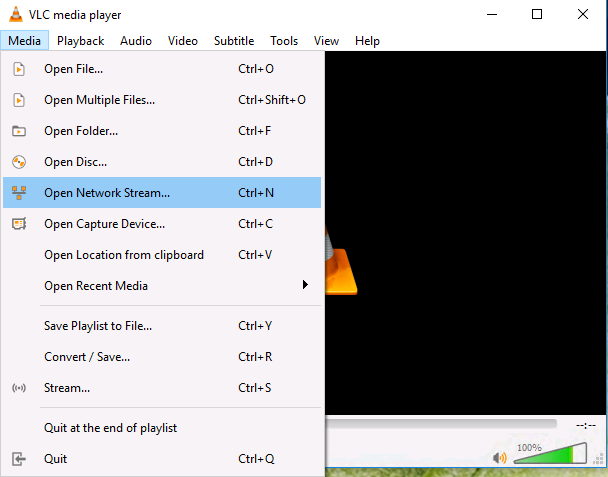
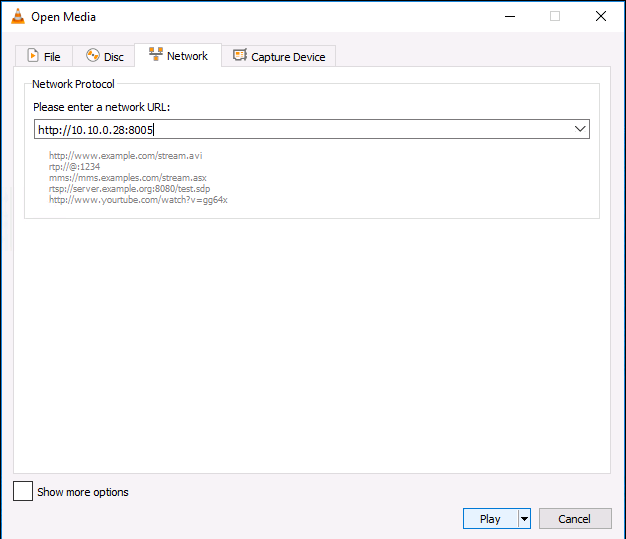
 

Figure 8: VLC Screenshots to show VLC Client setup

1. Setup the capture machine near the test devices. Start the capture.
2. Start streaming the output on the VLC video server by clicking on the Stream button shown in figure 7.
3. Start streaming on all client devices simultaneously by clicking on the Play button shown in figure 8.
4. Run test for 5 minutes and stop packet capture.
5. Open the packet capture in Wireshark and export data.
6. Analyze the data and calculate Airtime

## Wireshark Data Extraction Method

In order to calculate the airtime we need the following fields to be exported to the CSV.

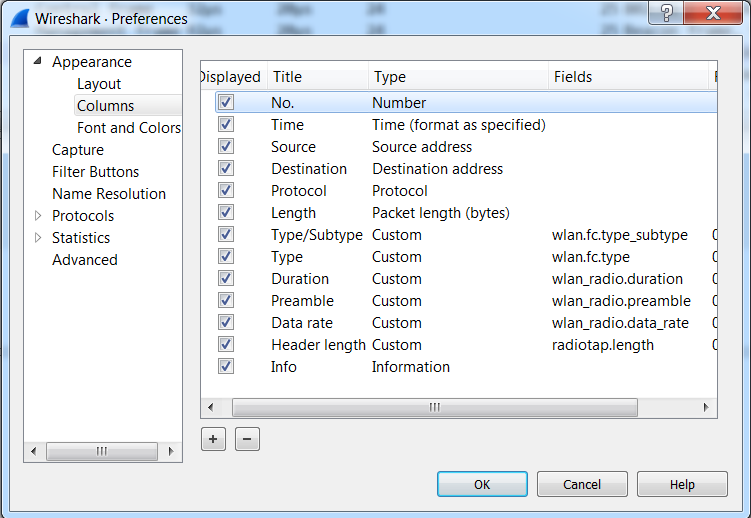


Figure 9: Wireshark Column Preferences

* The type field allows for filtering by data, management or control frames
* The Duration field can be used to cross check the final calculation
* The length field and data rate field to calculate airtime per packet

More details on the calculation procedure are provided in the next section.

Below is a screenshot of the different columns added to Wireshark before exporting the data.

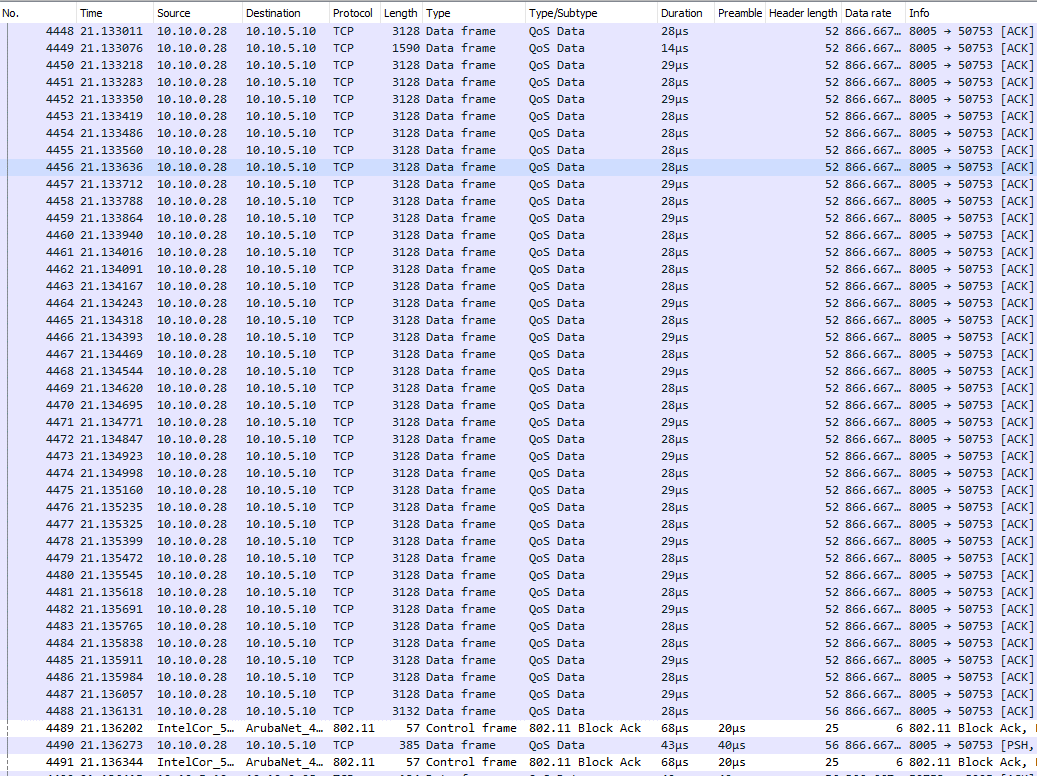


Figure 10: Wireshark columns

Once the columns are setup go to:

File > Export Packet Dissections > “As CSV”

and save the file. This will generate a csv file which can then be used for airtime utilization calculations.

## Airtime utilization calculation

An example of the Wireshark CSV file output is illustrated below.



Figure 11: CSV export

In Figure 10, the “Duration” and “Preamble” columns include the unit “µs” as part of the value field. Upon export to CSV, this gets converted to “\302\265s”. A simple way to correct for this is to open the file in Excel, select the Duration and Preamble columns, and perform a replace all operation to replace “\302\265s” with a blank space. The result is illustrated below.



Figure 12: Corrected CSV file

The simplest way to compute airtime utilization is to start with the values in the “Duration” column. The “Duration” column is Wireshark’s estimate of airtime of the MPDU. If this is the first MPDU in an A-MPDU, it includes the airtime of preamble. In the figure below, we see that packet number 408 is the first MPDU in an A-MPDU. The Duration field includes the preamble. Subsequent packets for the remaining MPDU’s in the A-MPDU, numbers 409-414, do not include the preamble duration. Then packet 415 is the 802.11 Block Ack.



Figure 13: MPDU duration

As seen in this example, the Total Duration can be computed by summing the duration from packet 1 to 221761. The “Time” column provides the time stamp for each sniffer packet in units of seconds. The time stamp of the last packet provides the total time of the video capture. Therefore Total Airtime Utilization is computed by converting Total Duration from µs to seconds, then dividing by the time stamp of the last packet. Since there are three users in this experiment, dividing by the number of users results in the airtime utilization per user.

Figure 14: Airtime utilization computation

On closer inspection, we see that in Figure 14 the preamble value for packet 221760 is 40µs. However, the data rate of the packet is 866.667 Mbps, which means it is a two stream 802.11ac transmission. Wireshark appears to be assigning 40 µs to the preamble duration of all of the two stream 802.11ac packets, but the preamble duration should be 44µs not 40µs. A simple correction is to add 4µs to the duration of each of the packets that contain an 802.11ac preamble (note that for the few 802.11ac packets in the capture that are single stream, this will erroneously add an extra 4µs). For the example in Figure 14, this results in additional total airtime of 92996µs. Since this is two orders of magnitude smaller than the Total Duration, the utilization per user increases very marginally from 0.93% to 0.94%.

# TEST Results

The test was conducted with the prescribed procedure in the Aruba-HPE CTO Lab. The lab is fully isolated from outside RF radiation which gives us a clean channel. The results are as follows for the 4.5 Mbps video file:

* **Airtime utilization when testing with one client: 1.17%**
* **Airtime utilization when testing with three clients: 0.93% per client (2.78% overall)**

# considerations for greenfield 802.11AX in 6 ghz

The video streaming measurements presented in this document were performed with devices based on IEEE 802.11ac technology. We expect IEEE 802.11ax to be the minimum technology in 6 GHz band. This section provides a conversion factor to adjust 802.11ac measurements for greenfield 6 GHz operation.

* Data Rate: One significant improvement in 802.11ax is the addition of 1024-QAM modulation and the use of longer symbols, which increases the spectrum efficiency and the available data rate. For example, physical layer data rate for 2 stream 80 MHz 802.11ac is 0.866 Gbps while it is 1.201 Gbps for 802.11ax. This is an increase of 38.5%.
* Average Bandwidth: 6 GHz 802.11ax devices are predicted to use higher average bandwidth (ref. 6 GHz SRDoc) which increases the average throughput of system and reduces the on-air time. Weighted average bandwidth for 6 GHz is 94 MHz as per the ETSI SRDoc. As such, the average data rate can be found scaling 80 MHz 11ax rate with the ratio of bandwidths, i.e. 94/80.

For purposes of SE45, the data rate and bandwidth adjustments apply only to data frames. Control and management frames are assumed to continue to operate at 6 Mbps and to be bandwidth invariant.

A scaling factor for data frames can be easily obtained from by combining the data rate and average bandwidth adjustments. This gives average data rate of 1.411 Gbps for a 2 spatial stream client in a 94 MHz channel (94 ÷ 80 \* 1.201). In order to estimate the duty cycle for 802.11ax devices under similar traffic loads, on-air time can be re-calculated by substituting the 0.866 Gbps rate with 1.411 Gbps for data transmissions.

Applying this scaling factor to the results of Section 4 produces the following adjusted values for the 4.5 Mbps video file:

* **Airtime utilization when testing with one client: 0.97%**
* **Airtime utilization when testing with three clients: 0.72% per client (2.15% overall)**

1. “How to Sniff Packets & Capture Packet Trace in Mac OS X the Easy Way”, http://osxdaily.com/2015/04/23/sniff-packet-capture-packet-trace-mac-os-x-wireless-diagnostics/ [↑](#footnote-ref-2)
2. “Getting a Packet Trace”, https://developer.apple.com/library/content/qa/qa1176/\_index.html#//apple\_ref/doc/uid/DTS10001707-CH1-SECWIFICAPTURE [↑](#footnote-ref-3)
3. “How to Use VLC as a Live Streaming Server”, http://letzgro.net/blog/how-to-use-vlc-as-a-live-streaming-server/ [↑](#footnote-ref-4)