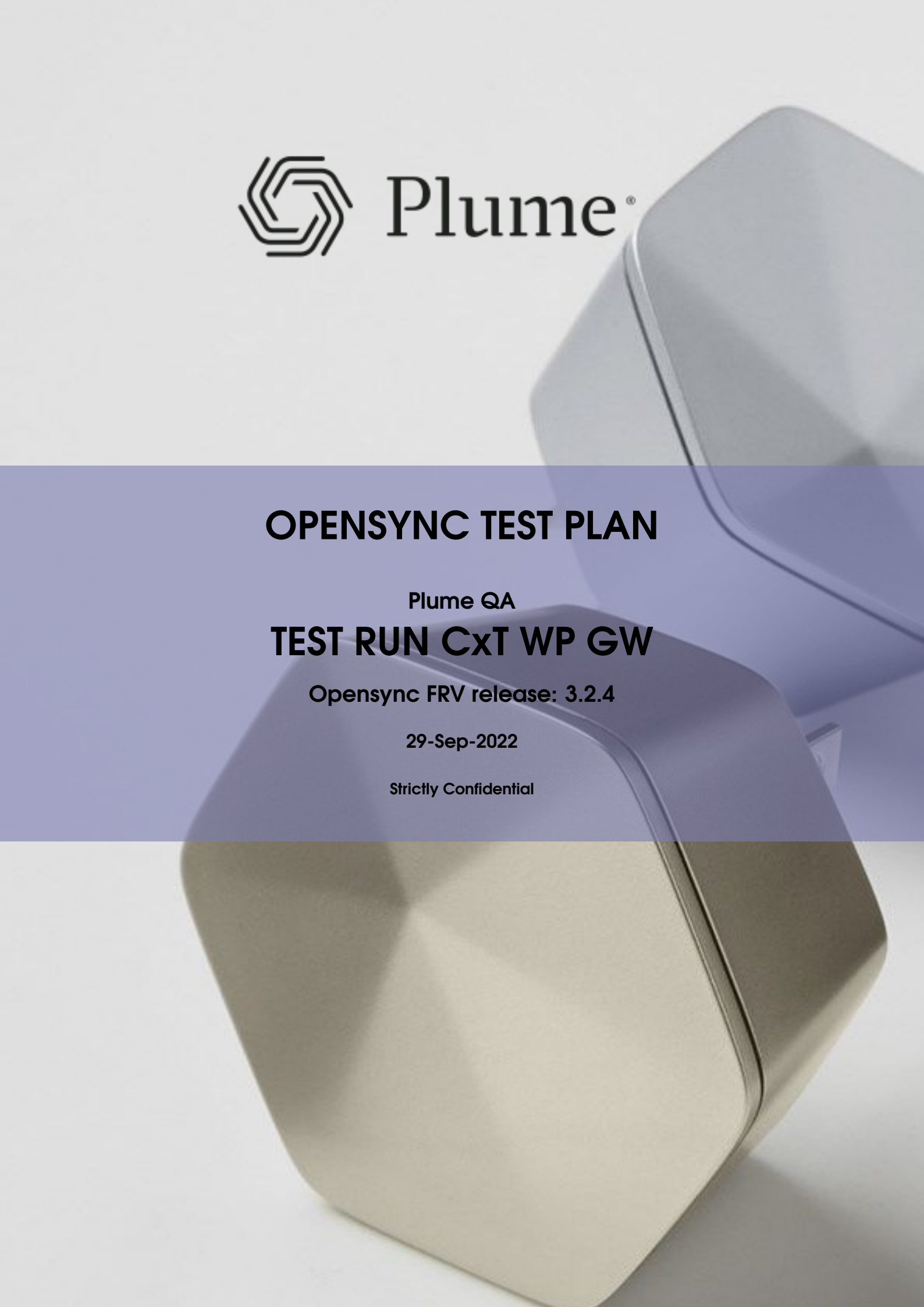


OPENSYNC TEST PLAN

Plume QA

TEST RUN CxT WP GW

Opensync FRV release: 3.2.4

29-Sep-2022

PUBLISHED BY PLUME
PLUME.COM

Contents

1. HomePass

1.1 Roaming Standards

1.1.1 iOS

#ICMP Roaming#

ICMP roaming results give us objective stats that we can compare between different clients, FRV, nodes, etc.

Preconditions:
–
– Location in tree topology (min. 3 nodes)
– Pretested position of nodes with Augustus version 3.4.1–88. We need reference measurement that works good so we can compare the results
– Computer (Ubuntu 20.04 LTS) connected to network with Ethernet to perform pinging to the roaming device:
  – Router mode: connected to GW node
  – Bridge mode: connected to switch to the same subnet (eg. CxT Unified environment – network connected to port 1 on CRS312 switch and laptop connected to port 2)
– Downloaded roaming.py and count_lost_ping.awk scripts (https://drive.google.com/drive/folders/141Iq83BtVlxP0wxuc1I_S8LIkck_1_Nv?usp=sharing)
Reference measurment:
–
For collecting any kind of measurments first pre–test the positioning of the nodes using Augustus pods on FW 3.4.1–88. You must use the same position of the nodes for all the testing later.

Do 50 roams using Augustus pods:
– 45 roams should be below 6 pings lost
– up to 5 roams from 6–8 pings lost
– 1 roam with more than 10 pings lost

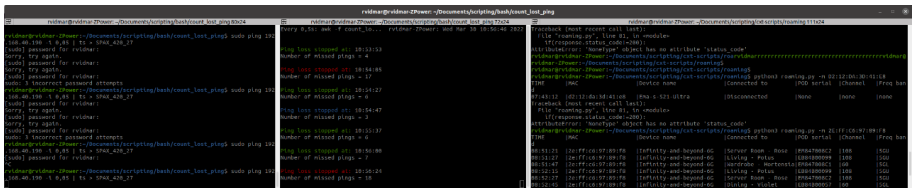
If you reach that KPI, positioning is considered OK for testing.

Instructions for ICMP roaming setup:
–
– Connect roaming device to the network
– In terminal move to the folder where you have ICMP AWK roaming script
– Start pinging device with TS and log everything into a file (depending on region/locale of the computer it wants either . or , in the interval):
  – $sudo ping 192.168.40.70 –i 0,05 | ts > ./FILENAME
  – $sudo ping 192.168.40.70 –i 0.05 | ts > ./FILENAME
– In other terminal in the same folder run AWK script (this script looks for missing sequences in ping data, and reports if sequence is missing):
  – $awk –f count_lost_ping.awk FILENAME
– You can use this command to automatically refresh the last 20 lines every 0.5 seconds:
  – $watch –c –n 0.5 "awk –f count_lost_ping.awk FILENAME | tail –n 20"
– Use python roaming script that logs each roam using cloud in third terminal
– You must open and modify script according to your location. See the image:

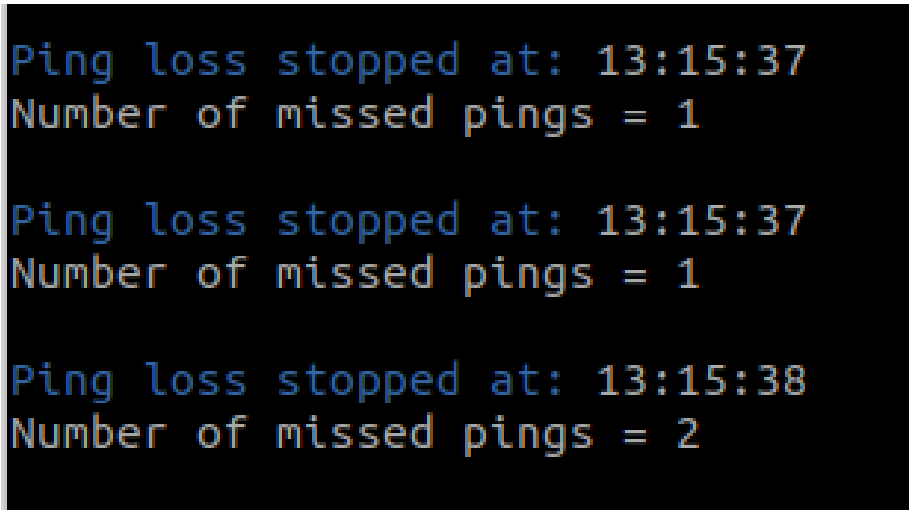
```
6
7   #url to either beta or df cloud, without /api !
8   #it should be like this https://piranha-beta.prod.us-west-2.aws.plumenet.io
9   #url="https://piranha-beta.prod.us-west-2.aws.plumenet.io" #beta cloud
10  url="https://piranha-dog1.dogfood.us-west-2.aws.plume.tech" #dogfood cloud
11
12  customerId="62377ef090a01e2a271a9204"
13  locationId="62377ef190a01e2a271a9205"
14
15  #Email and PW of the locationID
16  locationEmail="cxtxperience+cxth2510@gmail.com"
17  locationPassword="house17plume"
18
```

– Use command to run the script using MAC of the device you want to track
  – $python3 roaming.py –m 2E:FF:C6:97:89:F

– Your screen should look something like this after a few roams



Note: The time stamps of ping loss can be reported up to 3 seconds before the script that logs the roams. It can happen that there are more ping losses one after another on a roam (check the image below), note only the biggest ping loss. It can also happen that there is some ping loss when not roaming (disregard these ping losses, but note in test case resolution if there was any major ping loss not related to the roams).
Python script can also report disconnect even if there is not one, but if ping loss is still low 0–6 it is irrelevant.



Note down roams in the spreadsheet (same as performance and latency per hop): https://docs.google.com/spreadsheets/d/1LUx−rCeI7KB6K7cLkoMHKHABC4GSvYOrKmvhuvKM10c/edit#gid=534713672

KPIs:
–
In the 15 roams per client, there should be:
– no roams with more than 10 pings lost
– maximum 1 roam with 8−10 pings lost
– 15 roam average should be below 300ms (6 pings lost)

If roaming does not reach the above KPIs, proceed to do more roaming with the problematic client and see if you can reach the reference measurements.
Do 50 roams using DUT nodes:
– 45 roams should be below 6 pings lost
– up to 5 roams from 6−8 pings lost
– 1 roam with more than 10 pings lost

If these KPIs cannot be reached, open an ESW ticket for roaming issues.

1.1.1.1 ICMP Roaming performance - iOS

Case ID

C1868802

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup

2) Clients:
– Linux PC
– iOS smartphone
3) Two people to execute the test

Test Steps

1) Connect smartphone client to Wi–Fi and PC to ethernet
2) Ping smartphone with 0,05s period: sudo ping ip_address –i 0,05 –O #can be 0.05 depending on the locale of your PC
3) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Clients successfully connect
2) You are able to ping the smartphone
3) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.1.1.2 ICMP Roaming performance - bigger packet size - iOS

Case ID

C1952912

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– Linux PC
– iOS smartphone
3) Two people to execute the test

Test Steps

1) Connect smartphone client to Wi–Fi and PC to ethernet
2) Ping smartphone with 0,05s period: sudo ping ip_address –i 0,05 –s 1000 –O # can be 0.05 depending on the locale of your PC
3) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Clients successfully connect
2) You are able to ping the smartphone
3) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.1.2 Android

#ICMP Roaming#

ICMP roaming results give us objective stats that we can compare between different clients, FRV, nodes, etc.

Preconditions:
–
– Location in tree topology (min. 3 nodes)
– Pretested position of nodes with Augustus version 3.4.1–88. We need reference measurement that works good so we can compare the results
– Computer (Ubuntu 20.04 LTS) connected to network with Ethernet to perform pinging to the roaming device:
  – Router mode: connected to GW node
  – Bridge mode: connected to switch to the same subnet (eg. CxT Unified environment – network connected to port 1 on CRS312 switch and laptop connected to port 2)
– Downloaded roaming.py and count_lost_ping.awk scripts (https://drive.google.com/drive/folders/141Iq83BtVlxP0wxuc1I_S8LIkck_1_Nv?usp=sharing)
Reference measurment:
–
For collecting any kind of measurments first pre–test the positioning of the nodes using Augustus pods on FW 3.4.1–88. You must use the same position of the nodes for all the testing later.

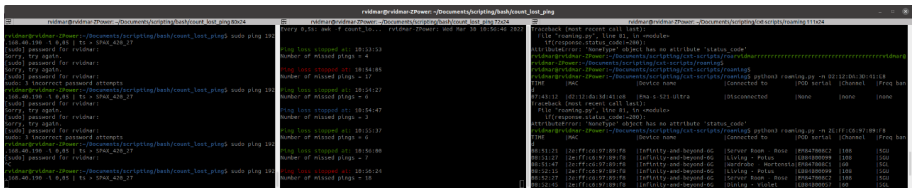
Do 50 roams using Augustus pods:
 – 45 roams should be below 6 pings lost
 – up to 5 roams from 6–8 pings lost
 – 1 roam with more than 10 pings lost
If you reach that KPI, positioning is considered OK for testing.

Instructions for ICMP roaming setup:
–
– Connect roaming device to the network
– In terminal move to the folder where you have ICMP AWK roaming script
– Start pinging device with TS and log everything into a file (depending on region/locale of the computer it wants either . or , in the interval):
  – $sudo ping 192.168.40.70 –i 0,05 | ts > ./FILENAME
  – $sudo ping 192.168.40.70 –i 0.05 | ts > ./FILENAME
– In other terminal in the same folder run AWK script (this script looks for missing sequences in ping data, and reports if sequence is missing):
  – $awk –f count_lost_ping.awk FILENAME
– You can use this command to automatically refresh the last 20 lines every 0.5 seconds:
  – $watch –c –n 0.5 "awk –f count_lost_ping.awk FILENAME | tail –n 20"
– Use python roaming script that logs each roam using cloud in third terminal
– You must open and modify script according to your location. See the image:

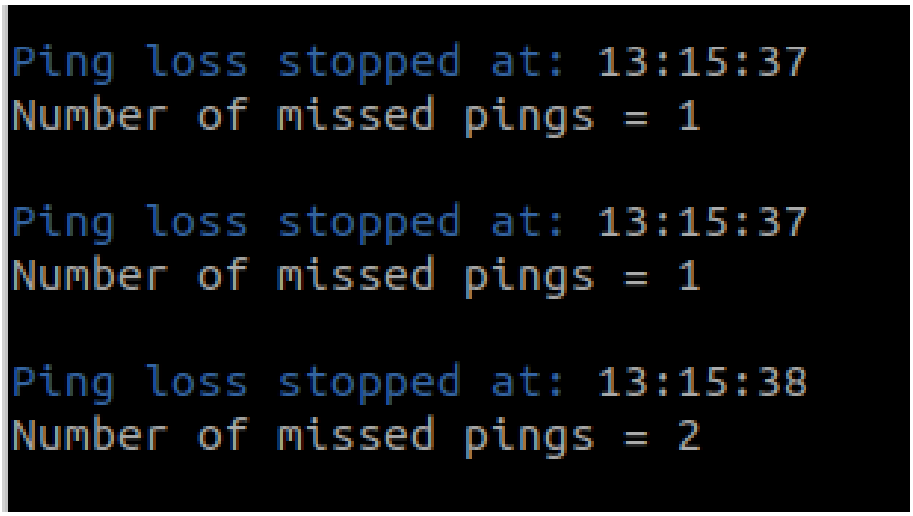
```
#url to either beta or df cloud, without /api !
#it should be like this https://piranha-beta.prod.us-west-2.aws.plumenet.io
#url="https://piranha-beta.prod.us-west-2.aws.plumenet.io" #beta cloud
url="https://piranha-dog1.dogfood.us-west-2.aws.plume.tech" #dogfood cloud

customerId="62377ef090a01e2a271a9204"
locationId="62377ef190a01e2a271a9205"

#Email and PW of the locationID
locationEmail="cxtxperience+cxth2510@gmail.com"
locationPassword="house17plume"
```

– Use command to run the script using MAC of the device you want to track
  – $python3 roaming.py –m 2E:FF:C6:97:89:F

– Your screen should look something like this after a few roams



Note: The time stamps of ping loss can be reported up to 3 seconds before the script that logs the roams. It can happen that there are more ping losses one after another on a roam (check the image below), note only the biggest ping loss. It can also happen that there is some ping loss when not roaming (disregard these ping losses, but note in test case resolution if there was any major ping loss not related to the roams).
Python script can also report disconnect even if there is not one, but if ping loss is still low 0–6 it is irrelevant.



Note down roams in the spreadsheet (same as performance and latency per hop): https://docs.google.com/spreadsheets/d/1LUx−rCeI7KB6K7cLkoMHKHABC4GSvYOrKmvhuvKM10c/edit#gid=534713672

KPIs:
−
In the 15 roams per client, there should be:
− no roams with more than 10 pings lost
− maximum 1 roam with 8−10 pings lost
− 15 roam average should be below 300ms (6 pings lost)

If roaming does not reach the above KPIs, proceed to do more roaming with the problematic client and see if you can reach the reference measurements.
Do 50 roams using DUT nodes:
− 45 roams should be below 6 pings lost
− up to 5 roams from 6−8 pings lost
− 1 roam with more than 10 pings lost

If these KPIs cannot be reached, open an ESW ticket for roaming issues.

1.1.2.1 ICMP Roaming performance - Android

Case ID

C1868803

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
− Linux PC
− Android smartphone
3) Two people to execute the test

Test Steps

1) Connect smartphone client to Wi−Fi and PC to ethernet
2) Ping smartphone with 0,05s period: sudo ping ip_address −i 0,05 −O #can be 0.05 depending on the locale of your PC
3) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Clients successfully connect
2) You are able to ping the smartphone
3) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.1.2.2 ICMP Roaming performance - bigger packet size - Android

Case ID

C1952913

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– Linux PC
– Android smartphone
3) Two people to execute the test

Test Steps

1) Connect smartphone client to Wi–Fi and PC to ethernet
2) Ping smartphone with 0,05s period: sudo ping ip_address –i 0,05 –s 1000 –O # can be 0.05 depending on the locale of your PC (OR use roaming.py script)
3) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Clients successfully connect
2) You are able to ping the smartphone
3) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.1.3 Video/Audio Services

1.1.3.1 MS Teams

Case ID

C1868808

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish MS Teams call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.2 Google Meet

Case ID

C1868809

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Google Meets call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.3 Zoom

Case ID

C1868812

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Zoom call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.4 FB Messenger

Case ID

C1868813

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Messenger call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.5 Viber

Case ID

C1868816

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Viber call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.6 WhatsApp

Case ID

C1868817

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the Android client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish WhatsApp call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.3.7 Wi-Fi Calling

Case ID

C1868818

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x Android client with Wi–Fi calling enabled
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Enable Airplane mode on the Android client and connect it to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish a call from roaming client to static client and vice versa
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe audio quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.1.4 Group Calls

1.1.4.1 MS Teams

Case ID

C1868804

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or mobile app like Aruba Utilities (Android) to track roaming
3) Clients:
– Windows PC Client
– 2 x Smartphone (Android)
– Independent static client (LTE, other ISP, different house)

Test Steps

1) Connect the Windows PC Client to wired internet OR connect a smartphone client to LTE
2) Connect the other 2 devices to the Node Wifi network
3) Call from one of the smartphones to the PC and the other smartphone (so that a minimum of 3 users are on the call)
4) Move around the house with one of the smartphones & track roaming in Frontline , Aruba Utilities/WiFi Analyzer or console
5) Start a Speedtest on one Wifi Client

Expected Results

1) Client connects to wired internet/LTE
2) Smartphone connects to Node WiFi
3) Call is established between all users
4) Call works well during roams (no bigger video or audio quality disruptions, no major delays)
5) Speedtest does not greatly affect the call

1.1.4.2 Google Meets

Case ID

C1868805

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or mobile app like Aruba Utilities (Android) to track roaming
3) Clients:
– Windows PC Client
– 2 x Smartphone (Android)
– Independent static client (LTE, other ISP, different house)

Test Steps

1) Connect the Windows PC Client to wired internet OR connect a smartphone client to LTE
2) Connect the other 2 devices to the Node Wifi network
3) Call from one of the smartphones to the PC and the other smartphone (so that a minimum of 3 users are on the call)
4) Move around the house with one of the smartphones & track roaming in Frontline , Aruba Utilities/WiFi Analyzer or console
5) Start a Speedtest on one Wifi Client

Expected Results

1) Client connects to wired internet/LTE
2) Smartphone connects to Node WiFi
3) Call is established between all users
4) Call works well during roams (no bigger video or audio quality disruptions, no major delays)
5) Speedtest does not greatly affect the call

1.1.4.3 Zoom

Case ID

C1868806

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or mobile app like Aruba Utilities (Android) to track roaming
3) Clients:
– Windows PC Client
– 2 x Smartphone (Android)
– Independent static client (LTE, other ISP, different house)

Test Steps

1) Connect the Windows PC Client to wired internet OR connect a smartphone client to LTE
2) Connect the other 2 devices to the Node Wifi network
3) Call from one of the smartphones to the PC and the other smartphone (so that a minimum of 3 users are on the call)
4) Move around the house with one of the smartphones & track roaming in Frontline , Aruba Utilities/WiFi Analyzer or console
5) Start a Speedtest on one Wifi Client

Expected Results

1) Client connects to wired internet/LTE
2) Smartphone connects to Node WiFi
3) Call is established between all users
4) Call works well during roams (no bigger video or audio quality disruptions, no major delays)
5) Speedtest does not greatly affect the call

1.2 Guard

1.2.1 Block website - URL

Case ID

C1868822

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x iOS/Android phone running HomePass app connected to the account of the test network
– 1x Android/iOS client and PC connected to Wi–Fi

Test Steps

1) Open Homepass app and navigate to Guard events tab
2) Open Block tab
3) Input a website address
4) Try to access the website from different clients connected to the network

Expected Results

3) Website is added to list of blocked websites
4) Website is not accessible on all clients

1.2.2 Block website - IP

Case ID

C1880313

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x iOS/Android phone running HomePass app connected to the account of the test network
– 1x Android/iOS client and PC connected to Wi–Fi

Test Steps

1) Open Homepass app and navigate to Guard events tab
2) Open Block tab
3) Input a website IP – use nslookup tool to find IP
4) Try to access the website from different clients connected to the network

Expected Results

3) IP is added to list of blocked IP's
4) Website on blocked IP is not accessible on all clients

1.3 Adapt

1.3.1 Platform features

1.3.1.1 Locate and name pods

Case ID

C1942609

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline.
3) Clients:
– 1x iOS client with access to HomePass
– 1x Android client with access to HomePass

Test Steps

1) Open HomePass
2) Navigate to the Adapt section on the home page
3) Tap the 3 dots in the pod list panel
4) Tap "Locate and name pods"
5) Move to a pod and tap it with your phone
6) Name the pod using a default name (Bedroom, Dining Room...)
7) Move to another pod, physically tap it with your phone and enter a custom name for the pod and tap Done
8) Name all other pods connected to the location
9) Check Frontline and ensure all pods are correctly named
10) Tap at least two already renamed pods and rename them again
11) Check Frontline and ensure all pods are correctly named

Expected Results

1) The app opens
2) The Pods panel shows all nodes, connected to the location
3) The additional functions show on the screen
4) The Locate and name pods screen opens and says "Tap to rename a pod" and " Looking for n pods...", where n is the remaining number of unnamed pods
5) The Name this pod screen shows up
6) The pod is renamed and shows on the Tap to rename a pod screen. The text is correctly adjusted
7) The pod is renamed and shows on the Tap to rename a pod screen. The text is correctly adjusted
8) The pods are renamed and show on the Tap to rename a pod screen. The text is correctly adjusted
9) All pods show up with the same names as in the App on Frontline
10) The pods are renamed and show the new names on the Tap to rename a pod screen
11) All pods show up with the same names as in the App on Frontline

1.3.1.2 Ethernet LAN stats GW node (Frontline) OS3.2+

Case ID

C1966114

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x client with Frontline access
– 1x wired Ethernet client connected to GW node

Test Steps

1) Connect client to GW node with ethernet connection and turn off Wi–Fi
2) Find connected device in Frontline and check its data consumption under "View history" drop–down and note it down
3) Download this file http://84.255.230.196/ffa/560mb
4) Upload that same file to WeTransfer (https://wetransfer.com/)
5) Wait 30min and leave Ethernet client idle (data in Frontline refreshes every 15min)
6) Check if Data consumption increased in range from 560Mb to 570Mb (560Mb + background usage) for download, and for upload

Expected Results

1) Client connects, and it shows it is connected to GW node with Ethernet in Frontline
3) File can be downloaded without issues
4) File can be uploaded on WeTransfer without issues
6) Data consumption for the client has increased in specified range for upload and download

1.3.1.3 Ethernet LAN stats Leaf node (Frontline) OS3.2+

Case ID

C1966117

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x client with Frontline access
– 1x wired Ethernet client connected to Leaf node

Test Steps

1) Connect client to Leaf node with ethernet connection and turn off Wi–Fi
2) Find connected device in Frontline and check its data consumption under "View history" drop–down and note it down
3) Download this file http://84.255.230.196/ffa/560mb
4) Upload that same file to WeTransfer (https://wetransfer.com/)
5) Wait 30min and leave Ethernet client idle (data in Frontline refreshes every 15min)
6) Check if Data consumption increased in range from 560Mb to 570Mb (560Mb + background usage) for download, and for upload

Expected Results

1) Client connects, and it shows it is connected to Leaf node with Ethernet in Frontline
3) File can be downloaded without issues
4) File can be uploaded on WeTransfer without issues
6) Data consumption for the client has increased in specified range for upload and download

1.4 Sense/Motion

1.4.1 Sense - Wi-Fi Motion Detection

Case ID

C1859128

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 3 x IoT WiFi device (video camera, motion sensor, SmartPlug) – stationary WiFi devices
– 1x smartphone (iOS or Android) – portable WiFi device

Test Steps

1) Place all WiFi sounding devices with a distance of 1–5m from the nearest gateway
2) Associate and distribute WiFi sounding devices across location tagged for motion detection testing
3) Enable Sense in HomePass app
4) Wait a few hours for Sense to start working
5) Check if all WiFi devices elected as sounding devices/sensors are able to sense motion
6) Roam around the house and check if Sense is detecting motion, icon changes to a man walking, circle around the button will fill to indicate motion intensity and text will change to a "Motion detected"
7) Check if (PubNub) control notifications are being sent from the cloud to HomePass
8) Check if user is able to tap on the WiFi motion button to enter the WiFi motion configuration screen
9) Check and test WiFi motion configuration menu and verify motion views
10) Check visibility of live and historic motion events (up to 7 days of data)

Expected Results

1) Sounding devices are placed
3) WiFi Motion Detection is enabled
5) All elected sounding sensors are able to sense motion
6, 7, 8) Icon acts as a live Motion detector and changes according to movement (stops if there is no movement, changes to walking figure if there is movement and shows intensity of the movement)
9) Notifications are being sent as needed
10) Configuration screen works
12) History of motion events works

1.4.2 Home/Away History

Case ID

C1859129

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x smartphone (iOS or Android) with HomePass, assigned to a person, Motion Detection ON for AT LEAST 3 hours & notifications ON

Test Steps

1) Have the device, that is assigned to a person, connected to the WiFi
2) Smart Activation is turned ON
3) Trigger motion events
4) Disconnect the device from the Node WiFi. Make sure, this was the last assigned device connected to the Node WiFi
5) Trigger some more motion events
6) Turn motion detection off
7) Check History of Motion Detection

Expected Results

1) Device connects to WiFi
2) Smart Activation is ON
3) Motion events are recorded in the app
4, 5) Motion events note that there is "noone home" and trigger differently colored events in the History of motion events
6) There is no more motion events after Motion detector is turned off
7) All colors (Home/Away/No motion) are visible in the history of motion detection menu

1.4.3 Smart activation

Case ID

C1859130

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 3x IoT WiFi device (video camera, motion sensor, SmartPlug) – stationary WiFi devices
– 1x smartphone (iOS/Android) – portable WiFi device assigned to a person
– 1x smartphone (iOS/Android) with HomePass installed

Test Steps

1) In the app turn Smart Activation on (go to the Sense menu, and enable it under Sense settings at the bottom)
2) Trigger movement events and check for notification (the app should not be open during this step, preferably lock the phone but keep it connected to the WiFi)
3) Remove the client from the location by leaving it and connecting to another WiFi hot spot or to LTE
4) Have another person trigger movement events at the location and check for notifications
5) Return to the location, connect to the Node WiFi and trigger movement events, then check for notifications

Expected Results

1) Smart Activation is turned on
2) Movement is triggered, noted in the History of Motion Detection but no notifications are sent to the client
3) Client is disconnected from the Node WiFi
4) Client gets notification warning about movement at Node WiFi while the client is "away"
5) Notifications are no longer happening after the client reconnects to the Node WiFi (is "home")

1.4.4 False Positives

Case ID

C1859131

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 3x IoT WiFi device (video camera, motion sensor, SmartPlug) – stationary WiFi devices
– 1x smartphone (iOS/Android) – portable WiFi device assigned to a person
– 1x smartphone (iOS/Android) with HomePass installed

Test Steps

1) Turn on motion detection and wait for it to set up
2) Choose the highest sensitivity for the Motion Detection
3) Choose or move sounding devices so that they are all in the same area (one or more rooms – half a room or an open space is not good). Sounding devices outside of the room will trigger unwanted motion events. All of the Nodes should also be in this area. The positioning of the sounding devices can have an effect in which direction false positives will be detected
4) Trigger motion events from inside of the designated area to set a benchmark
5) Close all the doors of the area and try to trigger motion events by moving close to the outside of the walls, doors, windows, ceiling and floor (be creative – bigger/ more people are better at triggering events)
6) If no motion events are triggered in step 4, move the sounding devices closer to the outer walls and doors and repeat step 4

Expected Results

1) Motion detection sets up
2) Highest sensitivity is ON
3) All sounding devices are in the same room, no devices out of the room are triggering any motion events
4) Motion events inside of the room are triggered correctly
5, 6) Motion events outside of the room should not trigger the Motion detection feature. Smaller mistakes are allowed, but please note them in the results

1.4.5 Plume Nodes

1.4.6 Same Channel

1.4.6.1 ICMP Roaming performance - bigger packet size - iOS

Case ID

C2069053

Test type

None

Test case coverage

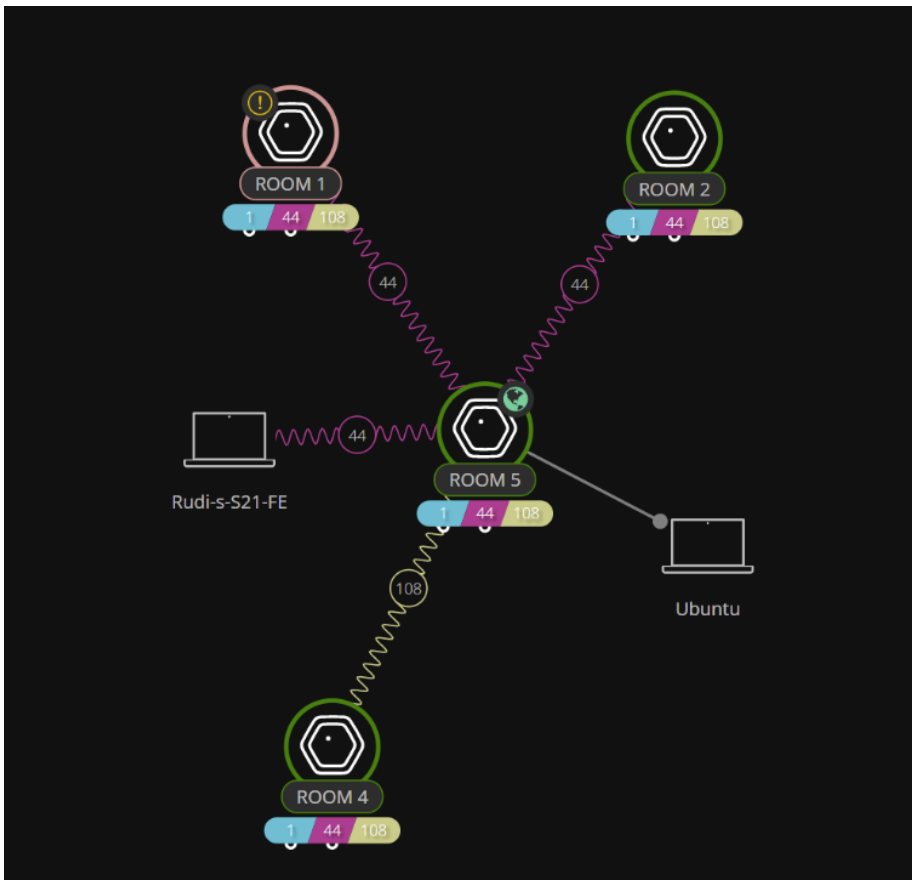
None

Preconditions

1) Initial test environment setup
2) Clients:
– Linux PC
– iOS smartphone
3) Two people to execute the test
4) Access to autotest–testrunner and Frontline

Test Steps

1) Open location in Frontline
2) Check that channels, which our clients will roam on, are the same. In case they are not, PTOPO the location accordingly. Example of appropriate topology:



3) Connect smartphone client to Wi−Fi and PC to ethernet
4) Ping smartphone with 0,05s period: sudo ping ip_address −i 0,05 −O −s 1000 # can be 0.05 depending on the locale of your PC
5) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Location opens in Frontline
2) Channels match
3) Clients successfully connect
4) You are able to ping the smartphone
5) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.4.7 Different Channel

1.4.7.1 ICMP Roaming performance - bigger packet size - iOS

Case ID

C2069054

Test type

None

Test case coverage

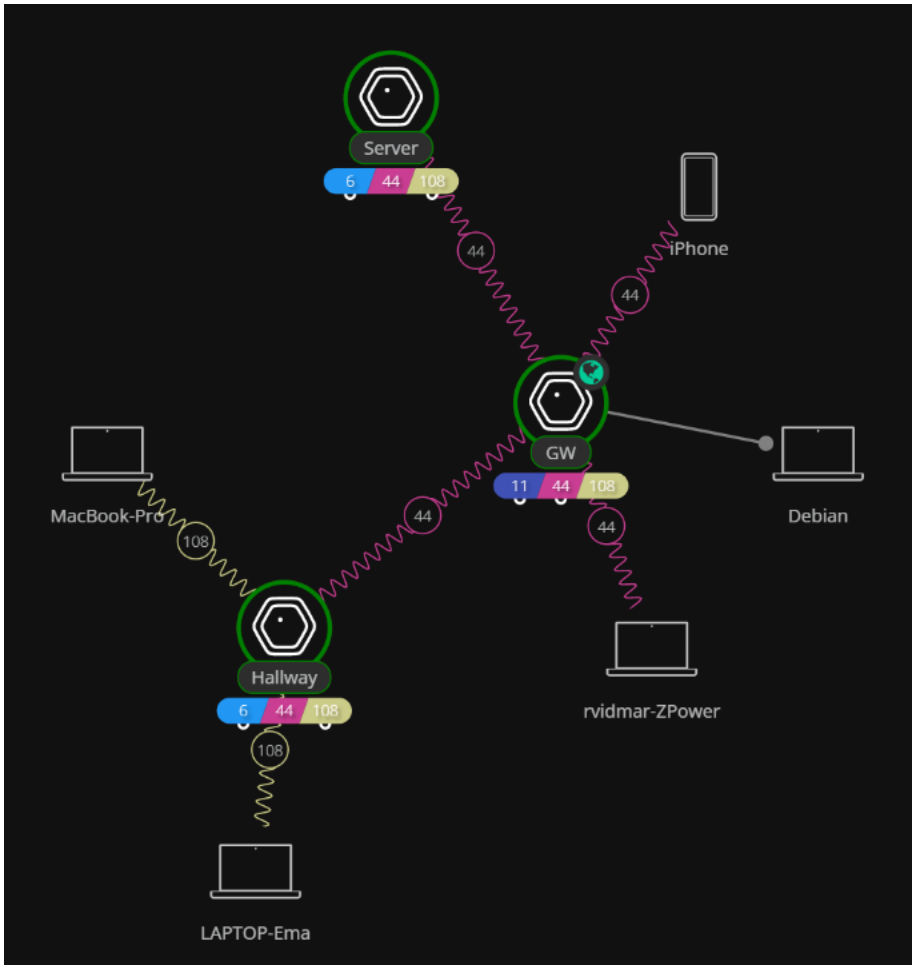
None

Preconditions

1) Initial test environment setup
2) Clients:
– Linux PC
– iOS smartphone
3) Two people to execute the test
4) Access to autotest–testrunner and Frontline

Test Steps

1) Open location in Frontline
2) Check that channels, which our clients will roam on, are NOT the same. In case they are, PTOPO the location accordingly. Example of appropriate topology:



3) Connect smartphone client to Wi−Fi and PC to ethernet
4) Ping smartphone with 0,05s period: sudo ping ip_address −i 0,05 −O −s 1000 # can be 0.05 depending on the locale of your PC
5) Walk around the house to make 15 roams between the APs, count the pings lost between roams

Expected Results

1) Location opens in Frontline
2) Channels do NOT match
3) Clients successfully connect
4) You are able to ping the smartphone
5) On average there should be less than 8 pings lost per roam, log the lost pings per roam in spreadsheet

1.4.8 Multi-PSK access control

1.4.9 WPA2

1.4.9.1 HomePass - Home zone

Case ID

C1859175

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a new password in the Home Zone
2) Connect the client with the new password
3) Check if client can access different services on the Internet and ping other devices on the same network
4) Repeat step 2–3 with a different device

Expected Results

1) Home Zone is created
2) Client connects to Home Zone and can be see under that zone in the app
3) Client can access the internet AND ping other devices on the same network

1.4.9.2 HomePass - Guest zone

Case ID

C1859176

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a two separate passwords for two guests (App settings –> Adapt –> WPA2 –> Guests)
2) Connect two clients to Guest networks with each password (4 clients altogether)
3) Connect all other clients to the Home zone – with Home zone password
4) Check connectivity between devices connected to the same guest zone (ping)
5) Check connectivity between devices connected to different guest zones (ping)
6) Check connectivity between devices connected to Guest zone and Home zone

Expected Results

1) Passwords created
2) Client connects to the Guest zone and is seen in the app under Guest zone
3) Clients connect to the Home zone and are seen in the app under Home zone
4) Connectivity between clients on the same Guest Zone (same password) is there
5) Connection between clients on the different Guest zones does not work
6) Connection between clients on Guest zone and Home zone does not work

1.4.9.3 HomePass - Guest zone - Ethernet devices

Case ID

C2074691

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a password for guest zone (App settings –> Adapt –> WPA2 –> Guests)
2) Connect one Wi–Fi client to Guest Zone
3) Connect one Ethernet client to one of the pods
4) Ping Ethernet client from client connected to Guest zone

Expected Results

1) Password created
2) Client connects to the Guest zone and is seen in the app under Guest zone
3) Client connects to the Home zone and is shown under Home zone
4) Connection between client on Guest Zone (same password) and client connected to Ethernet is not possible

1.4.9.4 HomePass - Intranet zone connectivity

Case ID

C1859178

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Device / Client:
– 1x client with Linux/WinOS with wired connection
– 1x Android client with WiFi connection
– 1x iOS client with WiFi connection
– 2x other Wi–Fi clients
– 1x client with Homepass app installed

Test Steps

1) Make a two separate passwords for two guests (App settings –> Adapt –> WPA2 –> Guests)
2) Connect two clients to with each password (4 clients altogether)
3) Share some of the devices from the Home zone to created Guest zone
4) Check if the Home zone devices that are shared with the Guest zone can be accessed by the devices in the right Guest zone

Expected Results

1) Password created
2) Client connects to the Guest zone and is seen in the app under Guest zone
3) Devices are successfully shared in HomePass app
4) Devices shared with Guest Zone can be pinged by devices in Guest Zone device which it was shared to, and not from the devices in Guest Zone which device was not shared to

1.4.9.5 HomePass - Internet Only zone

Case ID

C1859177

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a new password in the Internet only zone
2) Connect the client with the new password
3) Check if client can access different services on the Internet
4) Repeat step 2–3 with a different device
5) Use command from client connected to Internet only zone to check connectivity to other clients
fping –a –r 0 –g 192.168.40.0/24
where 192.168.40.0/24 is subnet you are using

Expected Results

1) Internet only zone is created
2) Client connects to Internet only zone and can be see under that zone in the app
3) Client can access the internet but cannot ping other devices on the same network
4) Clients cannot ping each other
5) Client on Internet only zone can only ping pods and itself

1.4.10 Video/Audio Services

1.4.10.1 FaceTime

Case ID

C1859155

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client for roaming
– 1x iOS/macOS independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the iOS/macOS client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish FaceTime call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) FaceTime call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.2 MS Teams

Case ID

C1859156

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS roaming client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish MS Teams call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on static client
7) Start a Speedtest on the Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.3 Google Meet

Case ID

C1859157

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Google Meet call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.4 Zoom

Case ID

C1859160

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Zoom call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.5 FB Messenger

Case ID

C1859161

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Messenger call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.6 Viber

Case ID

C1859147

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish Viber call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.7 WhatsApp

Case ID

C1859149

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish WhatsApp call from roaming client to static client
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.10.8 Wi-Fi Calling

Case ID

C1859148

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client with Wi–Fi calling enabled
– 1x independent static client
– 1x Wi–Fi client

Test Steps

1) Enable Airplane mode on the iOS/iPadOS client and connect it to Wi–Fi network
2) Connect the static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other client to Wi–Fi network
4) Establish a call from roaming client to static client and vice versa
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe audio quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Client connects to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.11 Group Calls

1.4.11.1 MS Teams

Case ID

C1859194

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 2x static clients
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect a static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other static client and the last remaining client to Wi–Fi network
4) Establish MS Teams call from roaming client to static clients (so that a minimum of 3 users are on the call)
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Clients connect to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.11.2 Google Meet

Case ID

C1859195

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 2x static clients
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect a static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other static client and the last remaining client to Wi–Fi network
4) Establish Google Meets call from roaming client to static clients (so that a minimum of 3 users are on the call)
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Clients connect to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.11.3 Zoom

Case ID

C1859196

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Access to Frontline or script/app to track roaming
3) Clients:
– 1x iOS/iPadOS client
– 2x static clients
– 1x Wi–Fi client

Test Steps

1) Connect the iOS/iPadOS client to Wi–Fi network
2) Connect a static client to LTE/other independent network outside of your Wi–Fi network
3) Connect the other static client and the last remaining client to Wi–Fi network
4) Establish Zoom call from roaming client to static clients (so that a minimum of 3 users are on the call)
5) Move around the house & track roaming through Frontline, console or mobile app
6) Observe video quality on independent client
7) Start a Speedtest on other Wi–Fi Client

Expected Results

1) Client connects to Wi–Fi
3) Clients connect to Wi–Fi
4) Call is established
5) The device roams
6) Call works well during roams (no bigger video/audio quality disruptions or major delays)
7) Speedtest does not greatly affect the call

1.4.12 WPA2-WPA3 Intranet Connectivity

1.4.12.1 HomePass - Home zone split SSID connectivity

Case ID

C2057959

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Split SSID network (WPA2+WPA3)
3) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Connect Client A to WPA3 network
2) Connect Client B to WPA2 network's Home Zone
3) Check that both clients have internet access
4) Check connectivity between client A and client B (ping)

Expected Results

1) Client A connects to WPA3 network
2) Client B connects to WPA2 Home Zone
3) Both clients can access the internet
4) Client A can ping Client B

1.4.12.2 HomePass - Guest zone split SSID connectivity

Case ID

C2057961

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Split SSID network (WPA2+WPA3)
3) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a password for a guest, or use existing guest password (App settings –> Adapt –> WPA2 –> Guests)
2) Connect two clients to Guest zone with guest password
3) Connect all other clients to the WPA3 SSID (Home zone)
4) Check connectivity between devices connected to the same guest zone (ping)
5) Check connectivity between devices connected to Guest zone and Home zone

Expected Results

1) Passwords created
2) Client connects to the Guest zone and is seen in the app under Guest zone
3) Clients connect to the Home zone and are seen in the app under Home zone
4) Connectivity between clients on the same Guest Zone (same password) is there
5) Connection between clients on Guest zone and Home zone does not work

1.4.12.3 HomePass - Intranet split SSID connectivity

Case ID

C2057960

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Split SSID network (WPA2+WPA3)
3) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a two separate passwords for two guests (App settings –> Adapt –> WPA2 –> Guests)
2) Connect two clients to with each password (4 clients altogether)
3) Share some of the devices from the WPA3 SSID (Home zone) to created Guest zone
4) Check if the Home zone devices that are shared with the Guest zone can be accessed by the devices in the right Guest zone

Expected Results

1) Password created
2) Client connects to the Guest zone and is seen in the app under Guest zone
3) Devices are successfully shared in HomePass app
4) Devices shared with Guest Zone can be pinged by devices in Guest Zone device which it was shared to, and not from the devices in Guest Zone which device was not shared to

1.4.12.4 HomePass - Internet Only zone split SSID connectivity

Case ID

C2057962

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Split SSID network (WPA2+WPA3)
3) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Make a new password in the Internet only zone
2) Connect the client with the new password
3) Check if client can access different services on the Internet
4) Repeat step 2–3 with a different device
5) Use command from client connected to Internet only zone to check connectivity to other clients
fping –a –r 0 –g 192.168.40.0/24
where 192.168.40.0/24 is subnet you are using

Expected Results

1) Internet only zone is created
2) Client connects to Internet only zone and can be see under that zone in the app
3) Client can access the internet but cannot ping other devices on the same network
4) Clients cannot ping each other
5) Client on Internet only zone can only ping pods and itself

1.4.13 WPA3

1.4.13.1 HomePass - WPA3

Case ID

C2057964

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) WPA3 Network
3) Client:
– 1x client with Linux/WinOS
– 1x smartphone client with Android
– 1x smartphone client with iOS
– 1x client with HomePass app installed

Test Steps

1) Connect clients to WPA3 network
2) Change WPA3 password
3) Reconnect clients to WPA3 network with new password
4) Ping clients between each other

Expected Results

1) Clients connect to the WPA3 network
2) Password is successfully changed and clients disconnect
3) Clients reconnect to the WPA3 with the new password
4) Clients can ping each other

1.5 Control

1.5.1 Device Freeze

1.5.1.1 Freeze a client for 2 minutes - Wireless clients

Case ID

C1868796

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x iOS/Android phone running HomePass app connected to the account of the test network
– 1x PC client
– 1x iOS/Android client

Test Steps

1) Open HomePass app and navigate to the Devices tab and select the client you want to freeze
2) Freeze the client using pause button on the top right corner of the device square
3) Set the timer for 2 minutes and press OK
4) Once the timer is active, try accessing the internet via a web browser (optionally run a "ping 8.8.8.8" command)
5) Right after the timer clears, try to access the internet again on the device that was previously frozen
6) Retest with the other client

Expected Results

1) You can find a suitable client to freeze
2) Menu with timer opens
3) Client gets frozen in the app
4) Client does not have internet connectivity and cannot ping outside to the internet
5) Client regains internet connectivity and you can use it normally

1.5.1.2 Freeze a client for 2 minutes - Wired clients

Case ID

C1880312

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x iOS/Android phone running HomePass app connected to the account of the test network
– 1x PC client with Eth port/dongle
– 1x iOS/Android client

Test Steps

1) Connect client with ethernet cable to the network node
2) Open HomePass app and navigate to the Devices tab and select the client that you connected with ethernet
3) Freeze the client using pause button on the top right corner of the device square
4) Set the timer for 2 minutes and press OK
5) Once the timer is active, try accessing the internet via a web browser (optionally run a "ping 8.8.8.8" command)
6) Right after the timer clears, try to access the internet again on the device that was previously frozen

Expected Results

1) Client gets internet access through ethernet cable
2) You can find a suitable client to freeze
3) Menu with timer opens
4) Client gets frozen in the app
5) Client does not have internet connectivity and cannot ping outside to the internet
6) Client regains internet connectivity and you can use it normally

1.5.1.3 Schedule client freeze

Case ID

C1868797

Test type

None

Test case coverage

None

Preconditions

1) Initial test environment setup
2) Clients:
– 1x iOS/Android phone running HomePass/Workpass app connected to the account of the test network
– 1x PC client
– 1x iOS/Android client

Test Steps

1) Navigate to the Devices tab
2) Select the client
3) Press schedule internet freeze
4) Press New freeze schedule
5) Set freeze time for 5 minutes from your current time
6) Set unfreeze time for 10 minutes from your current time
7) Title the test as you wish
8) Set repetition for your current day, then press save
9) Once the timer is active, try accessing the internet via a web browse (optionally run a "ping 8.8.8.8" command)
10) Once the scheduled freeze is over, try accessing the internet again

Expected Results

3, 4, 5) Freeze time is scheduled for 5 minutes from now
6) Unfreeze time is scheduled for 10 minutes from now
8) Scheduled Freeze is successfully saved
9) During the scheduled freeze, client does not have access to the internet
10) Client can successfully access internet after the scheduled freeze is over

Plume