WiCroft Architecture and Usage

March 25, 2019

1 Architecture Details

The architecture and working of the crowdsourced framework is explained here. Figure 1 shows the communication between client and server.

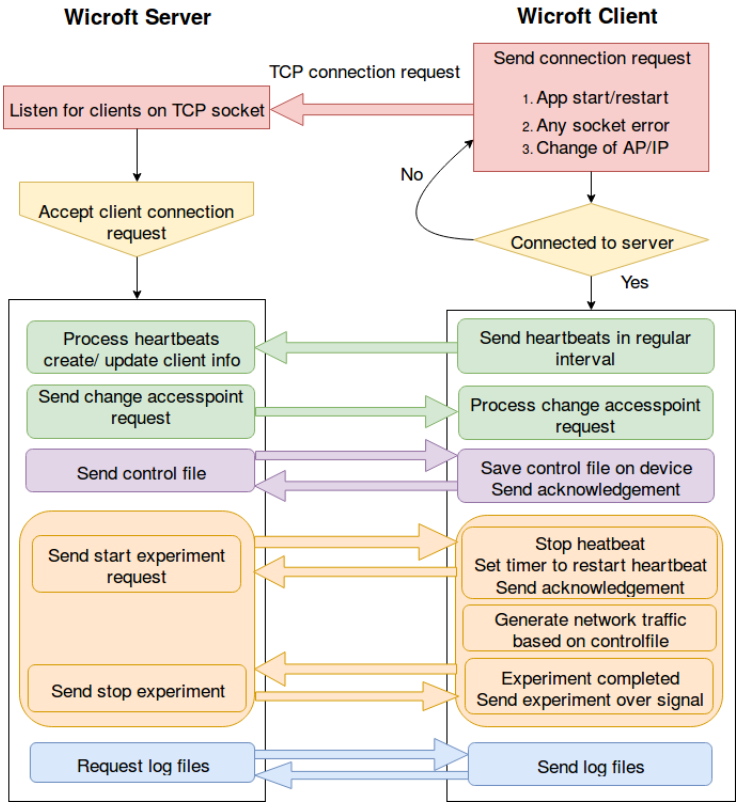


Figure 1: Crowdsouce application Architecture, message exchange between client and server

1.1 Working of the Crowdsoucing System

- In this design, the server is a Java Servelet application and client is an Android application.
- The server needs to start on a Servelet container like Apache Tomcat web server.
- The server will start with creating a TCP socket and listen for clients to connect. Each connected clients will be handled by Java thread.
- The Wicroft app needs to be installed on the smartphone. It is a background application which will run with minimum user intervention and tries to initiate and maintain a TCP connection to the server with provided server IP and port. This

approach solves the NAT issue and now we can make use of clients running behind a NAT.

- The client application tries to keep the TCP connection active. There in only one TCP connection per client to server exists. All the communication between client and server is served by this active TCP connection. The client will send heartbeat messages at regular intervals (every minute) to the server for indicating its presence.
- The heartbeat message contains IP, Port, Mac address, RSSI, SSID, BSSID, information about nearest access points etc. The server will update the information about client based on the data available in heartbeat. Table 1 describes the fields of the heartbeat message.

Table 1: Mac address to client information mapping

Key	**Value**
Mac Address	IP Address Port number SSID BSSID RSSI(dBm) LinkSpeed(Mbps) NearBSSList[] Processor Speed MemoryInfo Storge Space Number of cores Android OS version Device Model information LastHeatBeatTime Wicroft app version *Thread *Socket

- From server dashboard, it is easy to understand the number of clients connected and how it is distributed across access points, how many clients are active in sending heartbeats and how many are passive, this information is updated with every heartbeat from client.
- Sometimes it may be required to conduct scalability testing on a particular access point, WiCroft system support the feature to programmatically connect to a specific AP by providing it's SSID, security and authentication details.

- Once there are enough number of clients connected to server, an experiment can be conducted by crowdsourcing the network traffic generation to these clients. A file called as the control file, that specifies the pattern of URL request need to generate along with the request time and size. The URL pattern in control file can be a model of any application traffic like SAFE. The control files can send to selected clients in batches to reduce the WiFi contention. Manual retry is also added to ensure we ensure all selected clients receive the file. The clients will acknowledge after successfully receiving the control file.

- An experiment can be started by selecting those clients with a given control file. The start experiment request will be sent in batches, and with a single automatic retransmission to that client who didn't receive or acknowledge for first round request. Since the heartbeat traffic is not part of the traffic being simulated by Wicroft system, the client will temporarily stop the heartbeat during the experiment.

- The Wicroft app will log the status of URL requests like the success or failure, the reason for failure, response time of requests etc. Once all URL requests are done (success or failure) the client informs the server by an "experiment over" message.

- Once the experiment is over, the user from server side can stop the experiment so that the client can stop any running experiment and start sending the heartbeat.

- The experiment log files can be fetched from server interface by requesting to the clients in batches. Since the log files are of size 900 bytes to 1 Kb. There is chance to increase contention at the AP during this transfer. So request to log file in batches will result in sending the log files also in batches and avoid losing the log files. The client will send all pending log file during this transfer. The process of fetching log files is made separate from being part of an experiment, this also avoids overlapping with experiment traffic.

- Once experiment log files are received, the user can analyse it for identifying the client-side performance during the experiment.

1.2 Control file

The control file is one of the major components of a crowdsoucing system. It specifies the URL traffic need to be generated. It can be a model of any application traffic, like SAFE. Each entry of the file a specific format. Figure 2 shows one sample SAFE modelled traffic control file.

```
POST/GET url_request_time WEBVIEW URL size_of_request url_dependency
or
DASH/MP4 url_request_time EXO URL size_of_request url_dependency
```

The url_dependency equals to zero indicate this URL request is independent of previous URLs and this request can be generated after url_request_time from receiving the start experiment request at the client. A non-zero value indicates the URL request number on which this URL request depended on. i.e this URL request will be issued after url_request_time from completion of it's depended URL request. Control files for multiple clients can be specified in a single file with proper separation of 5 stars (∗ ∗ ∗ ∗ ∗). More details on modelling SAFE URL refer [3]

```
POST 60 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=quiz 340 0
POST 23 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=quiz-get 327 1
POST 178 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=partialsubmission 593 1
POST 290 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=add-log 480 1
POST 393 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=submit 586 1
POST 507 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=ldap-auth 341 0
POST 3 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=get-submission 314 6
POST 15 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=get-summary 357 6
*****
POST 69 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=quiz 340 0
POST 24 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=quiz-get 327 1
POST 170 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=partialsubmission 593 1
POST 179 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=submit 586 1
POST 348 WEBVIEW http://10.129.28.176:8080/TestServelet/testservelet?request=add-log 480 1
*****
```

Figure 2: Modelled URL traffic pattern for 5 minutes SAFE quiz for two clients

1.3 Log files from clients

The Wicroft Android application maintain following types of log files,

1. Experiment log file

   This file is associated with a particular experiment. Following information are logged into this file

   - URL request starting time and ending time, that gives the response time of URL request
   - URL request status, success of failure, reason for failure
   - Mac address, IP, Port number, SSID and BSSID of associated AP
   - Total playback time, number of stalls, video buffer time and average bit rate for video download experiments

2. Connection log file

   This file logs the WiFi connection and disconnection information of client device. The time of the event, SSID and BSSID of associated AP is also logged. This information helpful for finding the connection delay in a WiFi environment.

3. Debug log file

   The file log information like runtime exception, information about message sending to and received from a server. That information is helpful for debugging any issues with client devices while using the Wicroft app. This information is used in later part of the project to debug the hit ratio of clients who participated in experiments.

The complete architecture and added features of the client application are described in [3]. The server architecture and the implemented features are described in the following section.

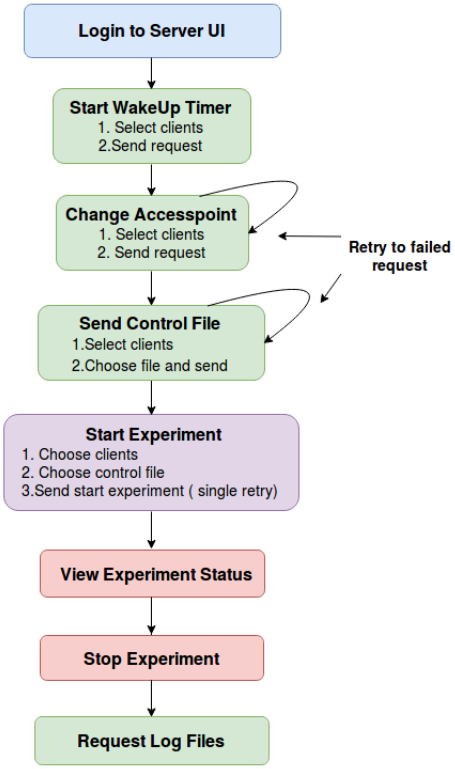


Figure 3: Flow chart for conducting an experiment

1.4 Server Architecture

1. Start the server and listen for clients to connect

   The server will start by creating a TCP socket for listening to clients to connect. Each connected clients are handled by single Java thread. The server maintains a mapping between the mac address of the client to all details including reference to

the thread. So whenever a client reconnects, the existing thread will be killed and a new thread will serve the client.

2. Receive Heartbeats from clients

   Once the client successfully connected to the socket, all communication between server and client will happen by using this TCP connection. The client will continue to send heartbeats at regular intervals (one minute) to indicate it's presence. The heartbeat contains the IP, Port, Mac address, received signal strength (RSSI), SSID and BSSID of associated AP, device hardware information, SSID and BSSID information of nearest APs, shown in Table 1. The server will maintain a map between client's macAddress to these details and updated with each heartbeat.

3. Start a wakeup timer

   The Wicroft is a background running android application. In some models of Android smartphone, it was observed that the WiFi manager instance is not passed to background services, power saving setting disturbs the functioning of background service, in the Android version of 6 and above the device will move to doze mode (with WiFi turned off) after some point of idle time. By moving the app to foreground the first two problems can be solved. A wake timer can be set from the server to instruct the Wicroft app to run in the foreground. Starting a wakeup timer is **recommened**, it will be helpful to maintain all connected clients for the duration of an experiment without loosing.

4. Request to change accesspoint

   Sometimes it may be required to conduct scale or performance testing based on a particular AP, it is also difficult to instruct the users to connect to a specific AP in a classroom. This module helps to avoid those difficulties, and the user can provide the target APs SSID, security and associated credentials to connect to it. There is no control over BSSID of AP. i.e If there are multiple BSSID with the same SSID, the user cannot force the client to connect to specific BSSID, it is not supported by Android. SSID level connection requests the user can make. The Wicroft app will programmatically try to connect to the specified AP.

5. Send controlfile

   The sending of a control file is done in batches to clients to reduce the WiFi contention. The Wicroft app will acknowledge successful receipt of the file, if not from the server this information are available, and file can be retried. There is an option provided to send a new control file or reuse already existing file.

6. Create and Start an experiment

   The user can create and save an experiment configuration and can be conducted at required time. While starting an experiment the user need to select the correct control file and also clients need to take part in the experiment. Sending start experiment

request happens in batches to reduce the contention. There is a single automatic retransmission to those clients who didn't receive the start experiment request in the first transfer. The start experiment request including retry is sent in such a way that all clients start the experiment synchronously. Once the client started with experiment it stops sending the heartbeat.

7. Stop a running experiment

   From the server side, the user can instruct to stop a running experiment. It may be for starting a new experiment or stopping a wrong experiment. After receiving the stop experiment command, the client will start sending the heartbeat.

8. Request log files

   The clients will maintain a log file, which includes the success of failure status of each URL request. The response time of each request. After completion of the experiment, the user sitting in front of the server can request for these log files. The request can be sent in batches so that the clients will send these logs in a way to reduce the contention. This module helps to fetch all pending log files from the client.

A flow chart describing the one sample flow of conducting experiment is shown in Figure 3. The format of messages exchanged between client and server are explained in Appendix.

1.5 Server implementation

The server with GUI for crowdsourcing application is implemented with a set of features discussed so far are described below. A flow chart for conducting an experiment is shown in Figure 6. A dummy server is also implemented to simulate actual SAFE server to handle the SAFE modelled request from clients.

1.6 Wicroft Server implementation [2]

The server for the Android client application is developed using Java, JSP, and Bootstrap. This application uses MySQL database to store the information related to the experiments, like start and end time, number of clients participated, client information (IP, PORT, Mac address, associated AP info), the status of start experiment request, experiment is finished or not, log file status etc. All architecture features are implemented and the following sections will describe the features implemented at the server along with the screenshots of the user interface.

1.6.1 Dashboard

The dashboard (Figure 4) is the front page where the user will login into, and shows the following information,

- The count of total number, active and passive clients connected to the server

  - Active client : The connected clients from which the server has received the heartbeat within expected heartbeat time window
  - Passive client : The connected clients from which the server has not received the heartbeat within expected heartbeat time window

- Individual client information like mac address, IP address, last heartbeat time etc.
- Accesspoint information, through which the clients are connected to the server
- Server timestamp

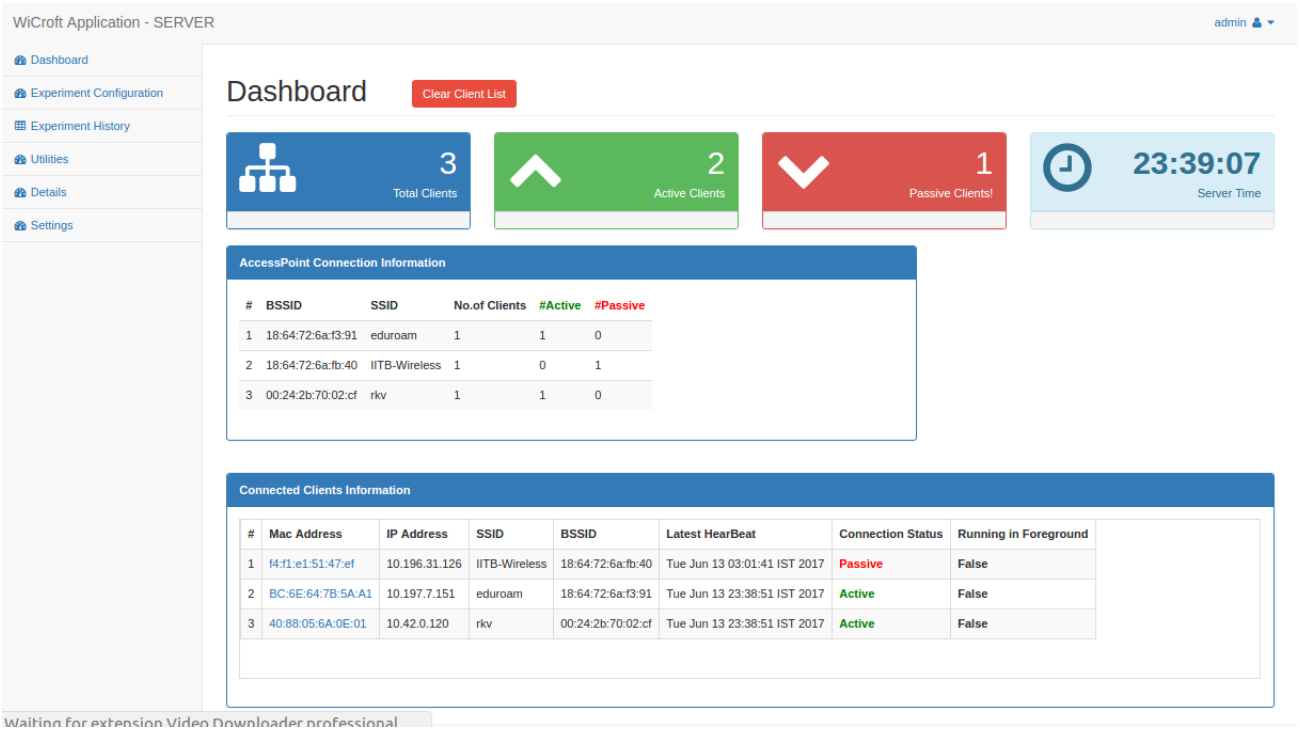


Figure 4: Dashboard of Wicroft server GUI

1.6.2 Request to change Accesspoint of clients

The server have option to send requests to each client for connecting to a specified access-point based on the experimenter's interest. This option is helpful to conduct scale test on a specific accesspoint. The Wicroft Android application is programmed to connect to a specific AP based on this request, which saves a lot of time from manually connecting to an accesspoint from each client endpoint. The Figure 5 shows the configuration to send the request, which has option to configure the following,

- SSID of the target accessoint
- Security type, Wicroft Android app currently support the eap, wep, open and wpa-psk security types
- Credentials based on the security type
- Timer : The time at when the client need to try to connect to the target AP, after receiving the request from server. This field can be used to connect a set of clients to a spectific AP at the same time.

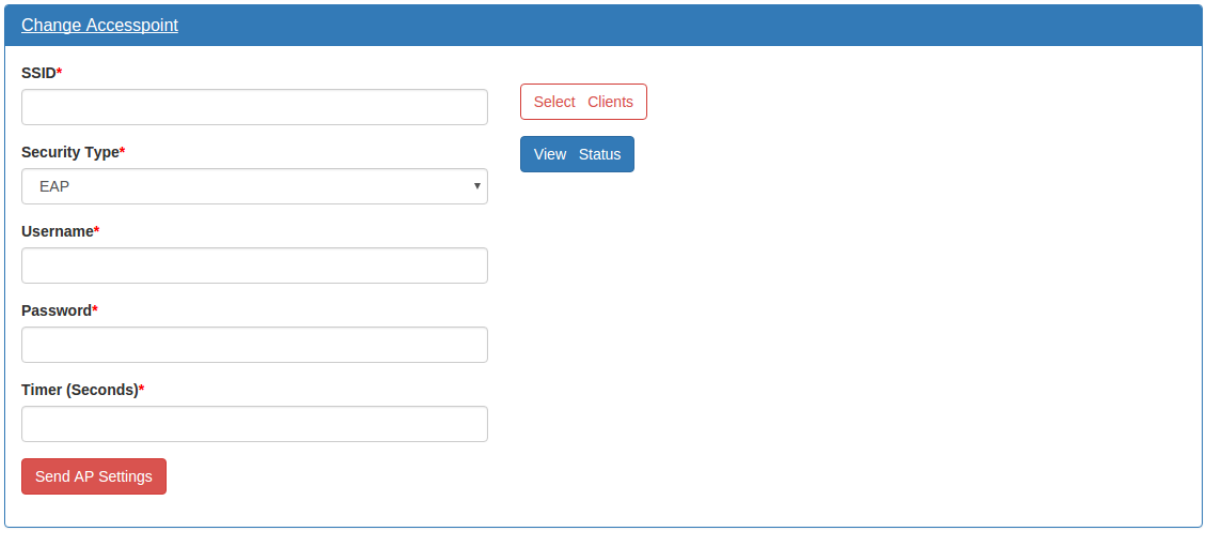


Figure 5: Create request to connect to a specific accesspoint

1.6.3 Select BSSIDs to choose clients

In order to view information about connected clients, the user need to select the BSSIDs from the available BSSID list obtained from clients heartbeat. The clients which are associated only with these selected BSSID lists will be displayed on the dashboard (shown in 4), also for conducting an experiment, sending control file and for fetching log files. The information is updated with every heartbeat. Last heartbeat time is also listed to view which all clients are active during a particular time. The Figure 6, shows option to choose the BSSIDs.

1.6.4 Control file information

The control files are the important components of the crowdsourced system (see section 3.2). The server has the following option to send and view the control files sent to the clients.

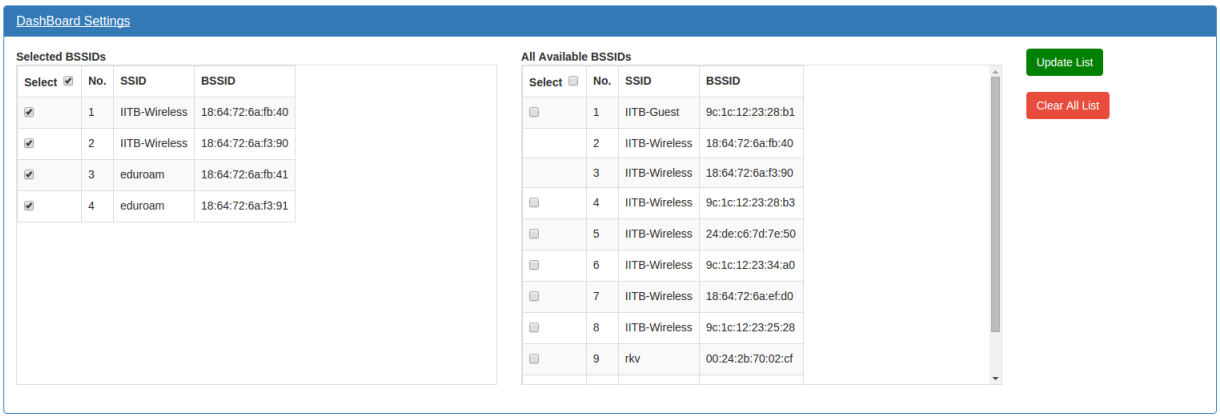


Figure 6: Select accesspoints to filter clients

1. Send new control files to the clients

   The Figure 7, shows the option to choose new control file and list of clients to send the file. An option is provided to send the file to set of clients in rounds, in order to reduce the traffic/congestion on the WiFi channel, due to bulk transfer. This is to maximize the successful reception of the file by the clients. We can specify the number of clients to send the files in a round and the time gap between the rounds.

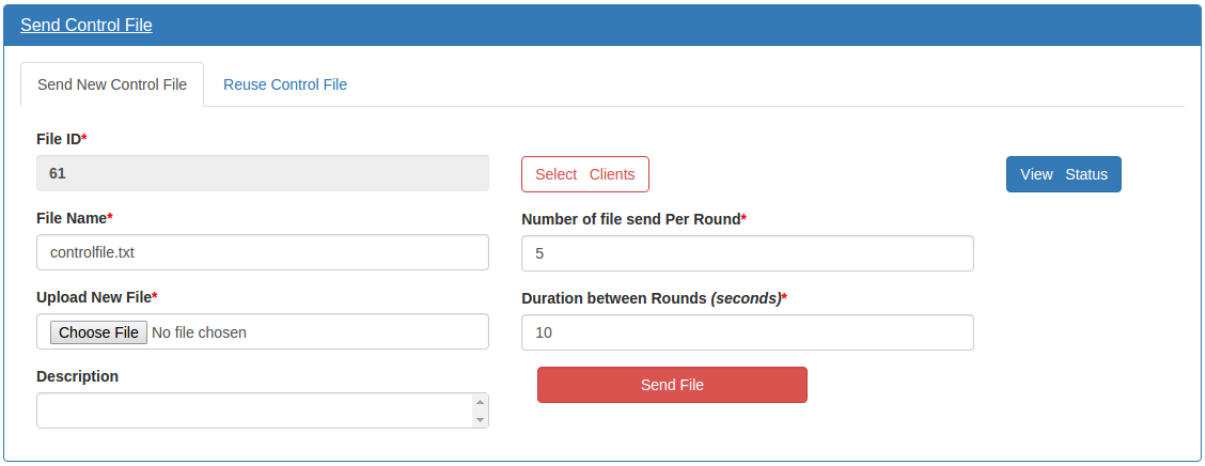


Figure 7: Configuration to send new control file to clients

2. Send previously used control files to the clients

   The Figure 8, shows the option to choose previously used control files and list of clients, to which the file was received earlier. Also an option is provided to send the file to set of clients in rounds, in order to reduce the traffic on the WiFi channel,

due to bulk transfer. This is to maximize the successful reception of the files by the clients. We can specify the number of clients to send the files in a round and the time gap between the rounds.

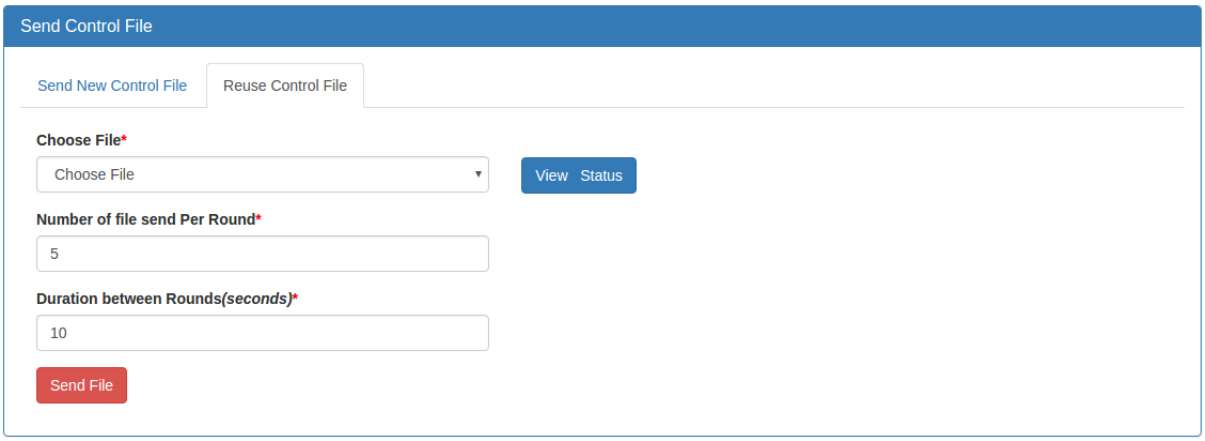


Figure 8: Configuration to reuse older control file

3. To view information about the transferred control files

   The Figure 9, shows information about already sent control files and number and details (link provided) of clients who has received the file successfully.

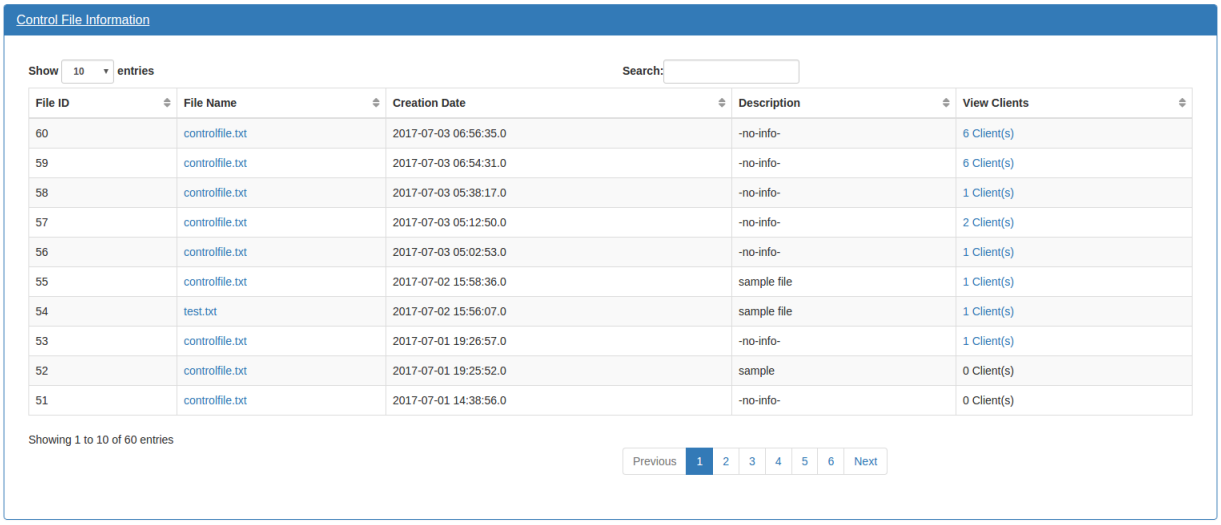


Figure 9: Information about transferred control files

1.6.5 Start a wakeup timer

The Figure 10 shows option to specify the duration of the timer and list of clients to choose for sending the request, refer section 3.4 for more information.

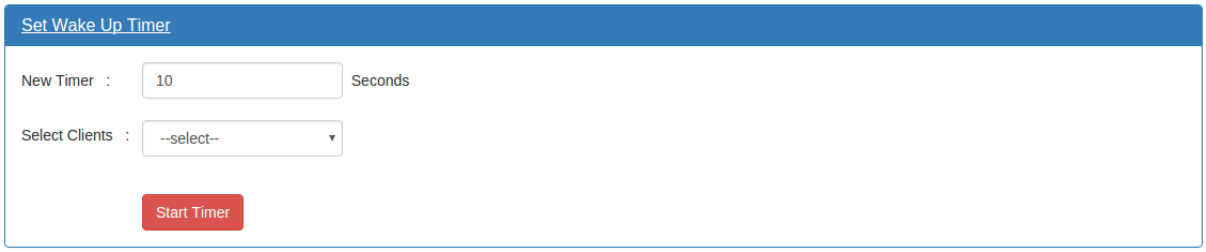


Figure 10: Option to start a wakeup timer for conducting an experiment

1.6.6 Create and save experiment

The Wicroft server provide the options to start with an experiment directly with available clients or to save and start the experiment later on.

1. Start an experiment

   The user can start an experiment by specifying the following information on the user interface (Figure 11)

   - Experiment details
   - Experiment timeout, at which the client stop the experiment, this is the expected experiment duration.
   - Start experiment request sends in rounds : The request per round and time gap between the rounds. This is to maximize the successful reception of the requests by the clients
   - Experiment acknowledgement waiting time : The maximum time at which the server is expected to receive the ACK for the experiment request received from the clients.
   - Experiment schedule time for retry request : If no ACK received from a client, server will have this duration of time to retry the request, the retry to start experiment will be send to automatically to these clients.
   - Control file
   - Select clients in the next step

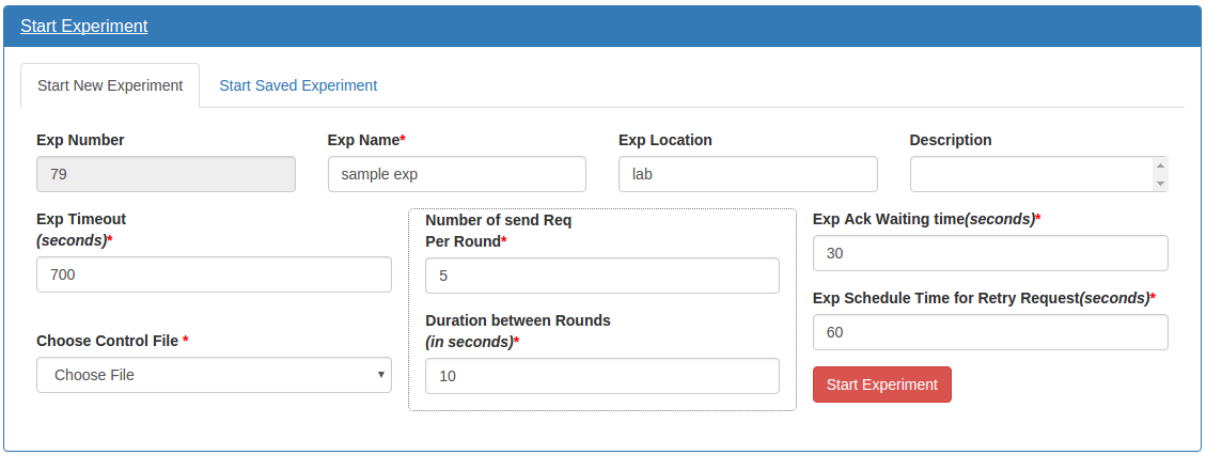


Figure 11: Configurations to start a new experiment

2. Save and load an experiment

   The user can create and save an experiment with or without a control file by providing experiment details. This saved experiment can be load and start whenever required. The Figure 12, shows the options to create and save an experiment. The user can specify the experiment details and optionally choose the control file.
   The Figure 13, shows option to load an already saved experiment, and option to choose the control file will be provided if the experiment was not configured with any. An option will be provided to select the clients once the experiment is loaded/selected.

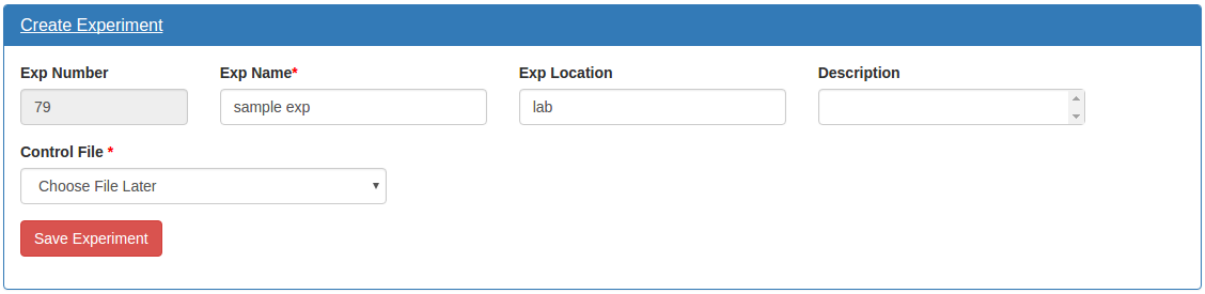


Figure 12: Configuration to create and save an experiment

1.6.7 View Experiment details or history

The Wicroft server maintain the information about the saved and executed experiments. The UI provide the information to view the experiment details (Figure 14)

The following information related to past experiment can be viewed in this module,

- Experiment status : Saved, running and completed experiments

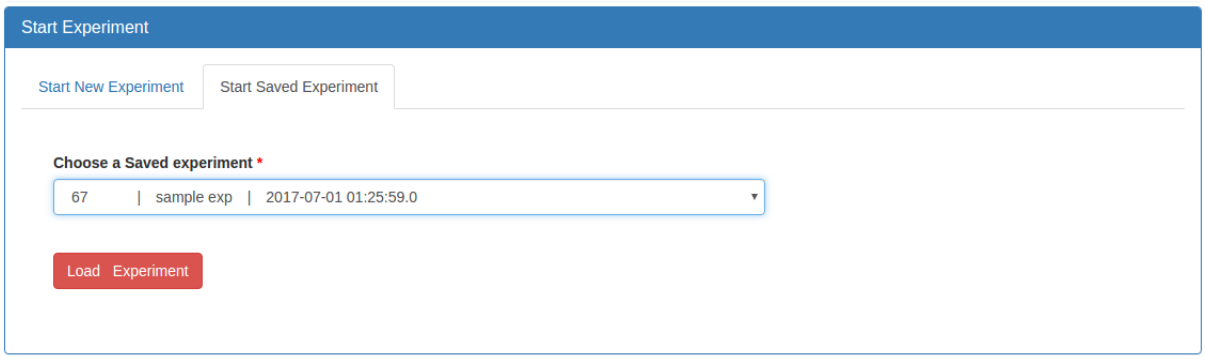
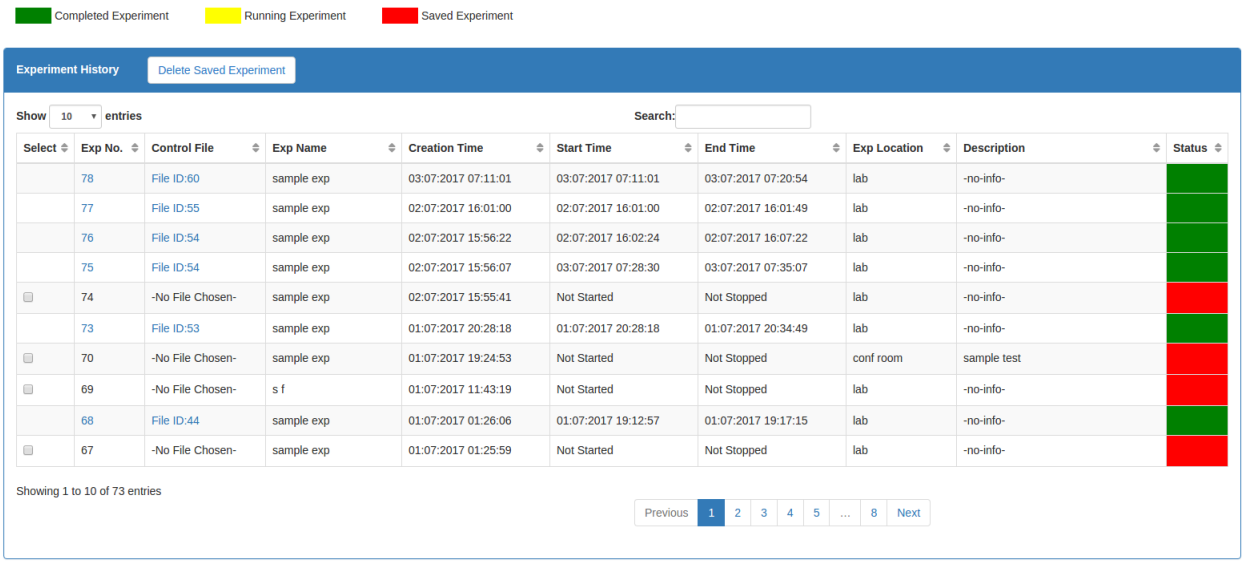


Figure 13: Option to load a saved experiment

- Experiment details like, creation time, location and control file used
- Duration of each experiment by start and end time
- IP, MAC, SSID, BSSID etc. details of clients participated in each experiment



Select	Exp No.	Control File	Exp Name	Creation Time	Start Time	End Time	Exp Location	Description	Status
	78	File ID:60	sample exp	03:07:2017 07:11:01	03:07:2017 07:11:01	03:07:2017 07:20:54	lab	-no-info-	
	77	File ID:55	sample exp	02:07:2017 16:01:00	02:07:2017 16:01:00	02:07:2017 16:01:49	lab	-no-info-	
	76	File ID:54	sample exp	02:07:2017 15:56:22	02:07:2017 16:02:24	02:07:2017 16:07:22	lab	-no-info-	
	75	File ID:54	sample exp	02:07:2017 15:56:07	03:07:2017 07:28:30	03:07:2017 07:35:07	lab	-no-info-	
	74	-No File Chosen-	sample exp	02:07:2017 15:55:41	Not Started	Not Stopped	lab	-no-info-	
	73	File ID:53	sample exp	01:07:2017 20:28:18	01:07:2017 20:28:18	01:07:2017 20:34:49	lab	-no-info-	
	70	-No File Chosen-	sample exp	01:07:2017 19:24:53	Not Started	Not Stopped	conf room	sample test	
	69	-No File Chosen-	s f	01:07:2017 11:43:19	Not Started	Not Stopped	lab	-no-info-	
	68	File ID:44	sample exp	01:07:2017 01:26:06	01:07:2017 19:12:57	01:07:2017 19:17:15	lab	-no-info-	
	67	-No File Chosen-	sample exp	01:07:2017 01:25:59	Not Started	Not Stopped	lab	-no-info-	

Figure 14: Details of experiments

1.6.8 Request for experiment log files

The Figure 15, show the option to send requests to clients for the experiment logs files, which is not transferred to the server yet. The user can select the clients and the requests

will be send in rounds, such that the clients will send back the log files in rounds, to reduce the WiFi channel traffic and to reduce the number of retries.

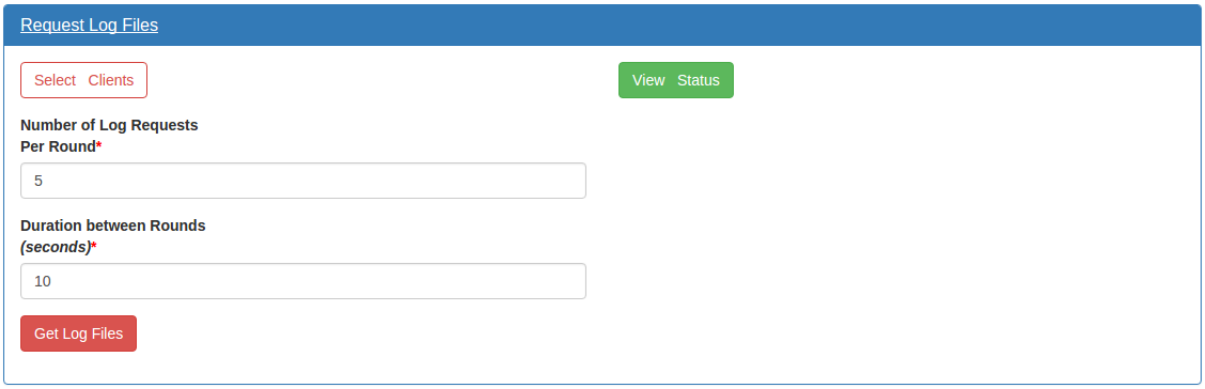


Figure 15: Configuration to send request to collect log files from clients

1.6.9 Wicroft App User informations

The server GUI is also provided with client information (MAC address and Gmail address, device name) who installed Wicroft Android app, and the version of the App they have. The last received heartbeat time also tracked to identify the non-cooperating client devices during an experiment. Figure 16 show the client information.

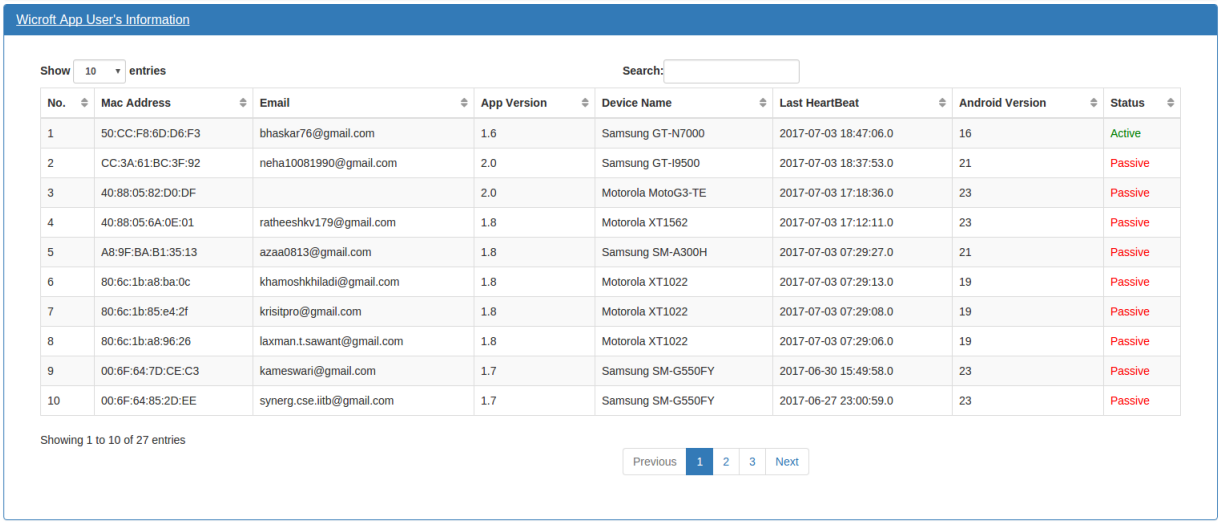


Figure 16: Information about users having installed the Wicroft Android application (Android version is the Android API level)

1.6.10 Create User account

The Wicroft server provides an admin account to user. In addition to that the user can create additional account from this section. Each user can perform all functionalities mentioned earlier. The UI provide the following option to handle the user accounts.

1. Create a new user account

   The user need to provide the new username and password (Figure 17)

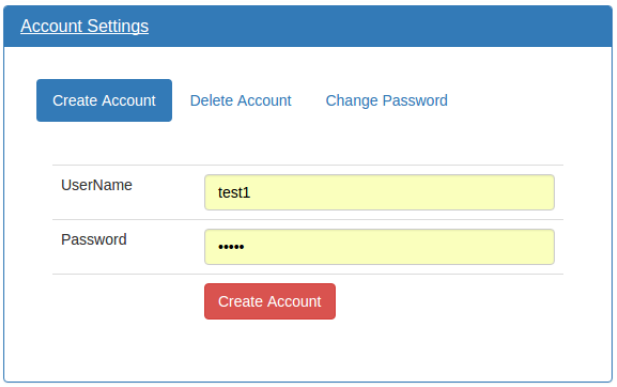


Figure 17: Option to create new user account

2. Delete an user account

   The user need to provide the username and password of the account (Figure 18)

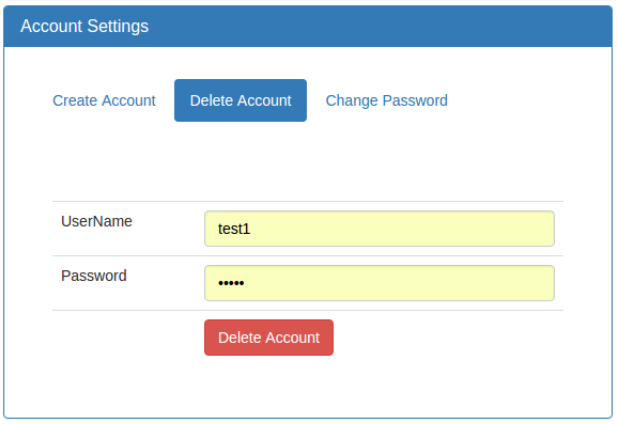


Figure 18: Option to delete a user account

3. Change password for an user account

   The user need to provide the username, current password and new password for password change (Figure 19)

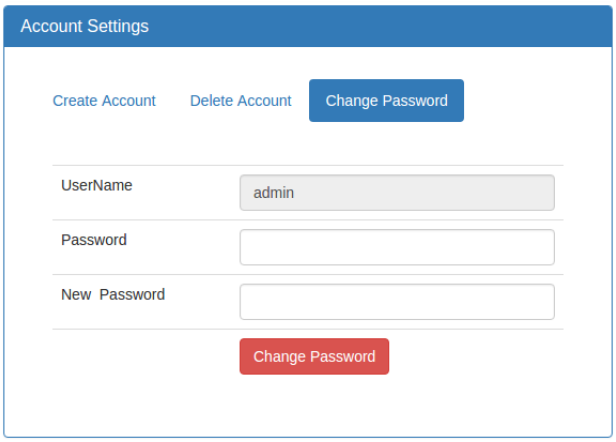


Figure 19: Option to changes password for an user

1.6.11 Send Wicroft Android app update notification

The Figure 20, shows the option to select the clients and send a message to notify the user about the availability of the new version of the Wicroft Android application on the Google PlayStore. The message once received will show a notification on the user's smartphone to inform the user to upgrade the existing app [3]. This option is helpful to notify about the availability of a new version of Wicroft application to the user.

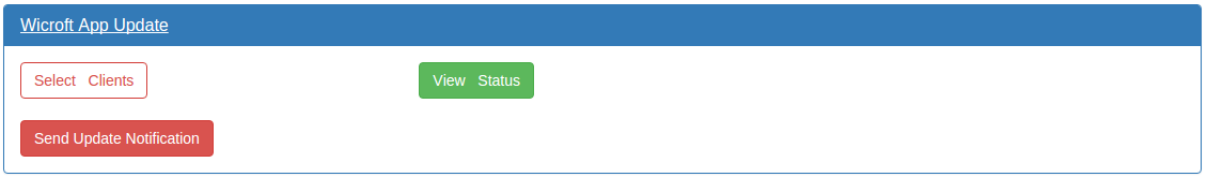


Figure 20: Option to send notification for Android app update to clients

1.7 Dummy SAFE Server implementation [2]

A dummy server is implemented to mimic the actual SAFE server. This server will respond with HTTP 200 OK message to all received HTTP GET/POST requests. Dummy server is implemented using JSP. The HTTP response messages have the following format to avoid additional *favicon.ico* GET request from a client.

```
<!DOCTYPE html>
<html><head>
<meta charset="UTF-8">
<link rel="icon" type="image/x-icon" href="data:image/x-icon">
```

```
</head>
<body></body>
</html>
```

The response time to each request and size of the response is modelled probabilistically based on analysis of actual SAFE server responses. The comparison between actual SAFE experiment delay versus Wicroft experiment delay is described later in this report in Section 7.

2 Appendix

2.1 Wicroft server GUI is explained

Login with credentials, an alert box will pop up to notifying that new clients are connected. In order to see those clients in a dashboard, go to setting, dashboard setting. Choose or check required BSSIDs from all available and update the list. The selected bssid will be shown in the corresponding list. The dashboard will reflect only those clients which are connected to these selected BSSIDs. This will be helpful once we need to conduct an experiment and need only those clients within than environment.

2.1.1 Dashboard

1. From dashboard, the user can able to understand how many clients are connected, among them how many are passive and active

   (a) Active clients

   The clients with Wicroft app installed will send heartbeats in every minute, those heartbeats may not be reached at the server in the same interval due to WIFi disconnection issues due to congestion or clients may switch off the WiFi after sending few heartbeats. So in the design of the server, a threshold is kept to classify clients whether a heartbeat is received within last threshold value of intervals, if yes, it is an active client else, passive clients. Passive does not always mean the clients left the environment, there can be cases like socket connection error, WiFi disconnection so the heartbeats may not be reached in regular intervals.

   (b) Passive clients

   Those clients from which server didn't receive heartbeats during the threshold interval. Current threshold value in 90 seconds for heartbeat duration of 60 seconds.

2. Distribution of clients among BSSIDs

3. Heartbeat information of connected clients

2.1.2 Experiment configuration

1. Create experiment

   (a) This module helps to create and save an experiment. The user need to provide some details of the experiment like name, location, description and also the user can choose or not choose a file for this experiment.

   (b) The saved experiments are able to see in Experiment History module.

2. Change accesspoint

   When the experiments are needed to conducted using a specific access point. It is very difficult to instruct every client to instruct to do so. This module will help in this scenario, from server side we can request to selected clients to connect to a specific access point by providing it's SSID, security and required credentials. We have no control over BSSID. i.e if there are multiple BSSIDs with the same SSID we cannot force the clients to connect based on BSSID. The clients may choose an AP based on signal strength it receives.

   (a) Using Wicroft the clients able to connect to desired AP. The security types can be WEP, OPEN, WPA-PSK and EAP.

   (b) A timer also can be specified to indicate, start trying to connect after these many time.

2.2 Events and message format

The server uses the existing TCP connection from clients for sending and receiving messages. The event types and corresponding message format are described below. The **"action"** key in JSON message identifies the event type of the message like heartbeats, control file etc.

1. HeartBeat from client application

   The heartbeat contains connection information like IP, BSSID, SSID of the associated access point, a list of BSSIDs of nearest AP, device information like storageSpace, processor speed etc. The heartbeat message format is,

```
{
"action"       : "heartBeat",
"macAddress"   : "mac address",
"ip"           : "ip addr",
"port"         : "port no",
"bssid"        : "bssid ",
"ssid"         : "ssid ",
"bssidList"    : "ssid1, bssid1, rssi1; ssid2, bssid2, rssi2;...",
"numberOfCores": "4",
"memory"       : "882",
"storageSpace" : "1371"
}
```

   Once the heartbeat is received from a client, the server will maintain a map between mac address of client with information available from heartbeat and it will be updated

on each heartbeat, see Table 1. The reference to thread and input and output stream from TCP connection is also maintained in the map. The heartbeat will also help the server to maintain the list of active clients.

2. Sending control file from the server as part of starting an experiment, this message contains control file data, server time for executing the URL request at a specified time.

```
 {
"action"        : "controlFile",
"serverTime"    : "SERVER TIME",
"message"       : "CONTROL FILE DATA "
 }
```

3. Change AP settings request from server

   This message is a request for connecting to a specified target access point using the AP change feature in the client application. This helps in remotely connecting the clients to specific AP. The target APs SSID, BSSID, security and authentication details need to be specified in the message.

```
  {
"action"  : "apSettings",
"username": "user name ",
"password": "password ",
"bssid"   : "bssid value",
"ssid"    : "ssid value"
"security": "wep/eap"
}
```

4. Request log files from server

   This request is sent from the server for getting pending log files from a client. Request for pending log file can be done at any time. The client will send the pending log file to the server by HTTP POST method

```
{"action" : "getLogFiles"}
```

5. Experiment over message from client This message is to indicate the experiment is over for that client. So that the server can request for log files.

```
{
"action"    : "expOver",
"exp"       : "experiment_number",
"macAddress": "mac address"
}
```

6. Acknowledgement for controlfile received from client

   The client will send an acknowledgement to indicate the control file for the respective experiment is received and it is ready to start the experiment. This is a level of application reliability added so that from server side it is able to show the status of control file sending.

```
{
"action" : "ack",
"exp"    : "exp_number"
}
```

References

[1] WiFi Data frames. http://www.my80211.com/home/2009/12/5/80211-null-data-frames.html.

[2] Ratheesh kv. Server source code. https://github.com/ratheeshkv179/MTP.

[3] Swinky Mann. Dense WiFi To test scalability of SAFE app. http://www.cse.iitb.ac.in/synerg/lib/exe/fetch.php?media=public:mtp1-oct16:dense_wifi_mtp1_153050021.pdf.

[4] Cisco Networks. WiFi Frame Types. https://supportforums.cisco.com/document/52391/80211-frames-starter-guide-learn-wireless-sniffer-traces.

[5] Cisco Networks. WiFi Frame Types. https://supportforums.cisco.com/document/101431/80211-sniffer-capture-analysis-management-frames-and-open-auth.

[6] Ashish Sonone and Sanchit Garg. Network Load generator. https://github.com/ratheeshkv179/Loadgenerator/blob/master/Doc/sanchit-RnD-II-Report.pdf.