MOBILE DATA NETWORKING

Assignment Report

Yansheng Liu z5124787
Zidong Zhou z5103491

Contents

Abstract

In practice what we have learned in this semester and found the features of Uniwide, which a WiFi covers most area of UNSW campus. Our team wrote several android programs which have been used to collect WiFi information. 26 check points in campus were checked in final test. Our team collected some useful link layer data near each check point and caught events happened during device movement from one check point to another.

After reorganizing all data and group discussion, we found that if our device is new enough and the Uniwide signal was strong enough, we can keep TCP connection during device movement in campus. L2 and L3 delay depends on the duration of the movement and the time device uses to find and connect a known WiFi.

Introduction

More and more evidents showed that the internet and smart phone play important roles in our common lives. The aim of this assignment is to practice the skills we have learned in this semester and a better understanding about the mobile network. Uniwide is a wireless network in UNSW campus. We have used android device and a program which is written by ourselves to check the Uniwide. During the program running on android device, it can detect and record the link layer information, which protocol the certain AP is using, signal strength, data rate and the AP density of that specific area. As well, the program still need to record TCP connection issues. Check and record any events procuring WiFi connection which happens during the movement of the android device.

Method and Program

A program is written to check and record the Uniwide and TCP/IP information. The whole program can be divided into 5 parts. UI, permission checking, files creating, thread creating and data writing. The principle of this program will be described in these 5 parts.

UI

The UI is shown as below as Picture 1.

Picture 1. Program UI

The program will automatically check and record information after a user click the RECORD on smart phone's screen. When user arrives at the certain area in campus, click the button shown on the right head side of the

screen and the link layer information will be written in a certain csv file which is stored in external storage *../DCIM/csv/*. In the middle of the screen is a textview. The textview is used to show user necessary information. Like when a file is created, what kinds of data it will record and so on. The information on screen will be deleted automatically when the screen is full. The processing bar in the bottom of the screen can show a user that the program is running normally. Whenever an error or some special flags are turned on in the program, the dot will stop running and a Toast will pop out.

Check Permission

Compared with the previous versions of Android, Android 7.0 has changed a lot, especially in the permission part. Designers can not use the permissions directly after these permissions are written in AndroidManifest.xml. Most important permissions have to be given by a user dynamically. We wrote the permission check part in the *onCreate()* part. If the user gives the permission in location, internet, WiFi and writing data to storage, the program can work normally or it will close automatically. The code is shown below.

```
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            int permissionCheck = ActivityCompat.checkSelfPermission(MainActivity.this, android.Manifest.permission.ACCESS_FINE_LOCATION);
            Log.i("permisiion", String.valueOf(permissionCheck));
            ActivityCompat.requestPermissions(MainActivity.this, new String[]{android.Manifest.permission.ACCESS_FINE_LOCATION, android.Manifest.permission.ACCESS_COARSE_LOCATION}, 1);
        }
        //check external Storage Status:
        if(Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)){
            Log.i("ExternalState","Works well.");
        }else{
            Log.e("ExternalState","Status error!!!!");
        }
        //check permissions:
        if(ContextCompat.checkSelfPermission(MainActivity.this, android.Manifest.permission.WRITE_EXTERNAL_STORAGE)!= PackageManager.PERMISSION_GRANTED){
            int permissionCheck = ContextCompat.checkSelfPermission(MainActivity.this, android.Manifest.permission.WRITE_EXTERNAL_STORAGE);
            Log.e("Permission: ", String.valueOf(permissionCheck));
            ActivityCompat.requestPermissions(MainActivity.this, new String[]{android.Manifest.permission.WRITE_EXTERNAL_STORAGE,android.Manifest.permission.MOUNT_UNMOUNT_FILESYSTEMS},1);
        }else{
            infor=infor+"LinkLayer: permission part works~~~~\n";
            statusTextView.setText(infor);
```

Picture 2. Check permission part

Create Files

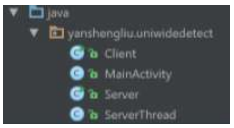
Data needs to be stored on to device whenever the program generated. After the user gives all necessary permissions. The program will check daily if there are files. If the day's files have not been created yet, 4 files will be created in (external storage)../DCIM/csv/. There are *yymmdd*L2.csv, *yymmdd*L3.csv, *yymmdd*L4.csv and *yymmdd*LLA.csv.(*yymmdd* means the year month and day.)

```
    private void CreateOrCheckFile(String FileName) {
        File currentFile=new File(FileName);
        if(!currentFile.exists()){
            try{
                currentFile.createNewFile();
                String temp="";
                if(FileName.equals(mFileNameLLA)){
                    temp="Check Point, TimePointer, Comment, Frequency(GHz), Speed, Signal Strength, AP Density";
                }
                else if(FileName.equals(mFileNameL2)){
                    temp="TimePointer, Comment, L2HOE, L2HOS, L2Period, currentBSSID, tempBSSID";
                }
                else if(FileName.equals(mFileNameL3)){
                    temp="TimePointer, Comment, L3HOE, L3HOS, L3Period, currentIP, tempIP";
                }
                else{
                    temp="Have no idea about this tcp link check!";
                }
                checkPointe0;
                WriteDataToFile(temp, FileName);
                infor=infor+"LinkLayer: First Time File Is Created\n";
                Log.i("File: ","File First time created!");
            } catch (IOException e) {
                e.printStackTrace();
                infor=infor+"LinkLayer: Error Occurred During File Creating\n";
                Log.e("File: ","Error Occurred During File Is Creating");
            }
        }else{
            infor=infor+"LinkLayer: Today's file has already created.\n";
            Log.i("File: ","File has already created sometime!");
        }
        statusTextView.setText(infor);
    }
```

Picture 3. Check and create files if necessary

Threads

There are 4 threads when the program is running. The main thread is used for checking link layer information. 3 sub threads are used for checking L2, L3 handoff and L4 connectivity respectively. The L4 connectivity checking part is written in another java file instead of the MainActivity.java. All java classes we used in this assignment are shown below.



Picture 4. Java classes

All 3 sub threads are based on the *CountDownTimer()* class in Java. Threads will be started after *Oncreate()* and will keep running during the duration of the program. The *CountDownTimer()* class needs two long variables for initializing. One is the total running time and the other is interval time. All 3 sub threads are set to work every millisecond. Which means if there are some events which occurs in 1ms, the program may not detect these kinds of events. When the *CountDownTimer()* times out, we wrote a restart command in its *onFinish()* part to restart itself.

```
317     private class TimeCount extends CountDownTimer {
318
319         private TimeCount(long millisInFuture, long countDownInterval) {
320             super(millisInFuture, countDownInterval);
321         }
322
323         @Override
324         public void onTick(long l) {
325
326             CheckL3Handoff();
327             count++;
328             if (count == 50) {
329                 count = 0;
330                 taskInfo = "";
331             }
332             taskInfo += "*";
333             processTextView.setText(taskInfo);
334         }
335
336         @Override
337         public void onFinish() {
338             infor="";
339             new TimeCount(60000,refreshTime).start();
340             Toast.makeText(MainActivity.this, "Test restart!", Toast.LENGTH_SHORT).show();
341         }
342     }
```

Picture 5. *CountDownTimer()*

MainActivity Link Layer Information

```
//read WiFi info and recorded in the resultData;
private String CheckPointLinkLayerInfor() {
   // String temp="Check Point, Comment, Frequency(GHz), Speed, Signal Strength, AP Density";
    String resultData="";
    mWifiManager.startScan();
    List<ScanResult> list = mWifiManager.getScanResults();
    mWifiInfo= mWifiManager.getConnectionInfo();
    double freq=mWifiInfo.getFrequency()/1000.0;
    int speed=mWifiInfo.getLinkSpeed();
    int apdensity=list.size();
    int signalStrength= mWifiInfo.getRssi();
    resultData=resultData+checkPoint+" , "+TimePointer()+" , "+linkLayerEditText.getText().toString()+" , "+freq+" , "+speed+" , "+signalStrength+" , "+apdensity+" , ";
    checkPoint++;
    String temp="Comment: "+checkPoint;
    linkLayerEditText.setText(temp);
    return resultData;
}
```

Picture 6. *CheckPointLinkLayerInfor()*

This part is in the *onResume()*. It checks and records every link layer information after the user click the RECORD button on the screen. It records these data: current time, WiFi frequency, speed, signal strength and AP density. The protocol of each APs is used will be analyzed by team member manually based on the frequency and speed.

L2 Handoff

```
private void CheckL2Handoff() {
    long L2Period =-2;
    mWifiInfo=mWifiManager.getConnectionInfo();
    String tempBSSID=mWifiInfo.getBSSID();
    String resultInfo="";
    if(L2HandOffSwitch==0 && currentBSSID==null && tempBSSID!=null){// 1st time enter campus zone!
        Toast.makeText(MainActivity.this, "L2 1st enter wifi", Toast.LENGTH_SHORT).show();
        L2HandOffSwitch=1;
        L2Period=-1;
        resultInfo= TimePointer()+", "+"1st Time Connect Uniwide"+", "+L2HOE+", "+L2HOS+", "+L2Period+", "+currentBSSID+", "+tempBSSID;
        currentBSSID=tempBSSID;
//          record L2Period and location and comments(1st enter wifi zone) into file.
    }
```

Picture 6. Part of *L2 Handoff()*

L2 handoff() is used to detect AP changes during movement from one location to another and record the L2 handoff delay. It will run every millisecond. During its running, *WifiInfo.getConnectionInfor().getBSSID()* will be checked. If it changes or lost BSSID, program will record the event into csv file. The event consists of TimePointer, Comment, L2HOE, L2HOS, L2Period, previousBSSID, currentBSSID.L2HOE means L2 handoff ending time point. L2HOS means L2 handoff starting time. Then use L2HOE minus L2HOS to get L2 period. There are 4 kinds of comments. L2switch is an integer variable which is used to aid program maintain different conditions.

1. First Time Connect Uniwide
   This comment will be written into file if the *L2switch* equals 0 and current BSSID changes from null to other number.
2. Disconnect because of low signal
   This comment will be written into file if *L2switch* does not equal 0 and current BSSID changed from a working BSSID(real BSSID) to a non-working BSSID (null or all 0). At the same time, recording the time point for calculate the L2 handoff delay.
3. Reconnect Uniwide
   This comment will be written into file if *L2switch* does not equal 0 and current BSSID changes from non-working BSSID to a working BSSID. At the same time recording the other time point for calculating L2 handoff delay.

4. Connect other AP without disconnect first
   This comment will be written into file if *L2switch* does not equal 0 and current BSSID changes from a working BSSID to a different working BSSID without disconnecting Uniwide. If this scenario happens, the record shows L2 handoff delay equals 0.

L3 Handoff

```
345     private void CheckL3Handoff() {// ip change!
346         String resultInfo="";
347         long L3Period=-2;
348         mWifiInfo=mWifiManager.getConnectionInfo();
349         int tempIPaddress=mWifiInfo.getIpAddress();
350 //       "TimePointer, Comment, L3HOE, L3HOS, L3Period, currentIP, tempIP";
351         if(L3HandOffSwitch==0 && currentIPadrress==0 && tempIPaddress!=0){// 1st time enter campus zone!
352             Toast.makeText(MainActivity.this, "L3 1st Enter!", Toast.LENGTH_SHORT).show();
353             L3HandOffSwitch=1;
354             L3Period=-1;
355             resultInfo= TimePointer()+", "+"1st Time Connect Uniwide"+", "+L3HOE+", "+L3HOS+", "+L3Period+", "+IPFormat(currentIPadrress)+", "+IPFormat(tempIPaddress);
356             currentIPadrress=tempIPaddress;
357         }
```

Picture 7. Part of *L3 Handoff()*

L3 Handoff() is used to detect IP condition during moving. Very similar to *L2 Handoff()*, it runs every millisecond, checking *WifiInfo.getConnectionInfo().getIpAddress()*. If IP address changes, it records the event time and store the information into a csv file. The event consists of TimePointer, Comment, L3HOE, L3HOS, L3Period, previousIP, currentIP. Then use L3HOE minus L3HOS to get L3 handoff delay. There are 4 kinds of comments. *L3switch* is an integer variable which is used to help program maintain different conditions, as with what *L2switch* does in *L2Handoff()*.

1. First time connect Uniwide
   This comment will be written into file if *L3switch* equals 0 and current IP changes from 0.0.0.0 to a working IP.
2. Disconnect because low signal
   This comment will be written into file if *L3switch* does not equal 0 and current IP changes from a working IP to 0.0.0.0.
3. Reconnect Uniwide
   This comment will be written into file if *L3switch* does not equal 0 and current IP changes from 0.0.0.0 to a working IP.
4. IP changed
   This comment will be written into file if *L3switch* does not equal 0 and current working IP is different from the record working IP. But in all tests we have done, this has never happened once.

L4 Connectivity

```
426     private void L4Handoff(int status){//0 disconnect ;1 reconne
427         String resultInfo="";
428         String tcpStatus="";
429         if(status==0){
430             tcpStatus="TCP break";
431         }else if(status==1){
432             tcpStatus="connected with different socket!";
433         }else if(status==2) {
434             tcpStatus="keep alive!";
435         }else{
436             tcpStatus="connected with same socket!";
437         }
438         resultInfo=TimePointer()+", "+tcpStatus;
439         resultInfo=resultInfo+" , "+checkPoint;
440         WriteDataTofile(resultInfo, mFileNameL4);
441     }
```

Picture 8. L4 *Connectivity().L4Handoff()*

```
13  public class Client {
14      public static void main(String[] args) throws IOException{
15          int check=0;
16          Socket client= new Socket("129.94.210.64",20001);
17  //          Socket client = new Socket("127.0.0.1",20001);
18          client.setSoTimeout(500);
19          BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
20          PrintStream out = new PrintStream(client.getOutputStream());
21          BufferedReader buf = new BufferedReader(new InputStreamReader(client.getInputStream()));
22          boolean flag=true;
23          while(flag){
24              String string ="1";
25              out.println(string);
26              if("bye".equals(string)){
27                  flag=false;
28              }else{
29                  try{
30                      String echo= buf.readLine();
31                      System.out.println(echo);
32                  }catch(SocketTimeoutException e){
33                      if(check==3){
34                          System.out.println("Time out,No response");
35                          System.out.println();
36                      }else{
37                          check++;
38                      }
39
40                  }
41              }
42          }
43          input.close();
44          if(client!=null){
45              client.close();
46          }
47      }
```

Picture 9. Client.java

```
public class ServerThread implements Runnable{

    private Socket client = null;
    public ServerThread(Socket client) { this.client= client; }
    @Override
    public void run(){
        try{
            System.out.println("Enter the thread!");
            System.out.println(client.toString());
            PrintStream out = new PrintStream(client.getOutputStream());
            BufferedReader buf = new BufferedReader(new InputStreamReader(client.getInputStream()));
            boolean flag=true;
            while(flag){
                String string=buf.readLine();
                out.println("From Server: "+string);
            }
            out.close();
            client.close();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

Picture 10. ServerThread.java

There are two parts in L4 Connectivity. First is the *L4handoff()* in MainActivity.java. It is used for receiving the information generated by Client.java if there is TCP disconnected event, *L4Handoff()* will record the time and situation. Client.java runs in Android device. There is a simple server which keeps running in a computer in Uniwide environment. After Client.java runs in a cellphone. It sends information every half second. The server responses every information send by client. If client doesn't receive any replies in 2 seconds, program detects TCP connection disconnected and send status to *L4Handoff()*. This disconnection may have been caused by changing AP or changing IP or disconnecting from Uniwide. Further analyzation will be given in the Result and Discussion part of this report.

There are 3 kinds of comments in this part.

1. Keep alive
   It will appear when BSSID changes but the client can still receive the message sent by server.
2. TCP break
   It will appear when client can not detect any information in 2 seconds.
3. Connected with different socket
   It will appear after TCP break and connected again with different socket.
4. Connected with same socket
   It will appear after TCP break and connected again with same socket. But during all the tests, this has never happened.

Write Data

```
private void WriteDataTofile(String usefulData,String currentFileDirection) {
    //read from file first:
    try {
        InputStreamReader read = new InputStreamReader( new FileInputStream(currentFileDirection));
        BufferedReader bufferedReader= new BufferedReader(read);
        String line =bufferedReader.readLine();
        mStringBuilder=new StringBuilder();
        while(line != null ){
            System.out.println("test=read section==>"+line);
            mStringBuilder.append(line);
            mStringBuilder.append("\r\n");
            line =bufferedReader.readLine();

        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    //write all data into file:
    try {
        BufferedWriter out = new BufferedWriter(new FileWriter(currentFileDirection));
        String temp=usefulData+",\n";
        mStringBuilder.append(temp);
        //mStringBuilder.append("\r\n");
        out.write(mStringBuilder.toString());
        out.flush();
        //out.newLine();
        out.close();
        temp= "LinkLayer: write into file>>>"+usefulData;
        infor=infor+temp+"\n";
        statusTextView.setText(infor);
        Log.i("Write:","LinkLayer infor write succeed!");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

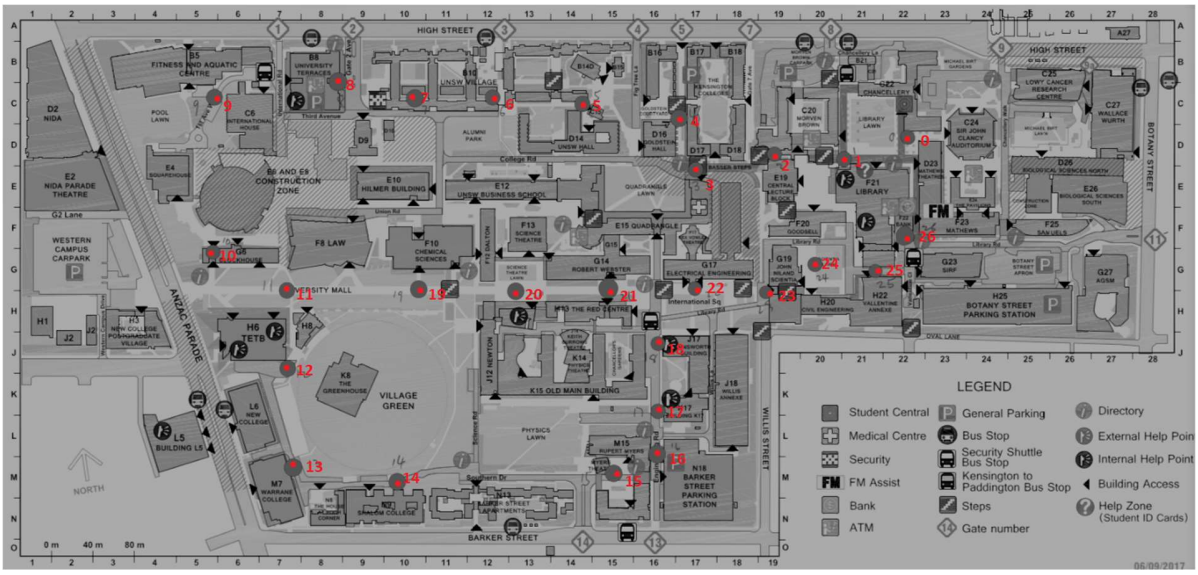
Picture 11. *WriteDataToFile()*

This function is used to write new data in specific files. It receive the information string and path string. Writing information string into file which finds by path string. But before writing the new row into the file. Reading all rows and rewriting all of them including new row into that file. Any threads can use this function.

Result

All data and the corresponding analysis are based on the program we developed and SAMSUNG A5(2017) with Android 7.0.

Check points are shown in the picture below.



Picture 12. Map of UNSW

These red points are the check points. The numbers near the points means the sequence of each points. All data was collected on the way to from one check point to another. Our team collected data from check point 0 to check point 26. The Check Point columns in all tables of this report stand for the check points in this map.

The original data and reorganized data tables will be shown in the Appendix. The mobility data is reorganized by time point. Different colors means different kinds of data. Brown stands for L2 handoff event. Blue stands for L3 handoff event. Green stands for L4 event. L2 and L3 handoff only happens when device is disconnected from Uniwide due to weak signal.

Discussion

Link layer data

Uniwide cover most of UNSW campus. And most of APs in Uniwide run under 802.11n. But there are still some old APs using 802.11 a or g. As the analysis is only based on the WiFi frequency and speed, the result may not 100% accurate.

AP density has a positive correlation with the human density within that area. Generally, the more people in that area, the more APs can be detected. As with check point 11 and check point 2 because these two points are very near to libraries, so large number of APs can be detected. Though check point 23 and 24 are in the central of campus, the number of APs is observably smaller than the previous two check points.

Mobility Data

L4 connectivity

In all tests we have done so far, TCP disconnection only happens when a cellphone is disconnected from Uniwide. What's more, if disconnection happens, neither server nor client can communicate each other with the previous socket after cellphone reconnected the Uniwide(same IP and same communicate port number). They will create a new socket for communication with each other. And the communication port number is an arbitrary number following the port number rule.

L3 Handoff

The subnet IP address have not been changed during all the tests. We have written some test programs before the final version of the android program. And in each tests, we have recorded the same IP address(10.249.16.207) given to the test cellphone. Some tests even did in different days. So we guess that the expirion period of the

subnet IP must be greater than 20 hours. When a cellphone logins the Uniwide, campus subnet server will check the expiry period of this device and change accordingly,which explains why our team could not detect subnet IP changes during the whole tests.

L2 Handoff

If our test phone does not connect any WiFi, there is no BSSID can be shown in *WifiInfo.getIpAddress().* But if the phone is connecting the Uniwide, this command will return a BSSID with all 0. We believe this means device is finding an available AP and communicating with it. In common scenarios, we can get a common BSSID by running the command. Usually, changing AP due to relocation, will not leave the Uniwide WiFi, and device will not be influenced during AP changing. There is only one condition that may lose Internet connection. And that is due to the low WiFi signal. The WiFi signal is too weak to use in some areas in campus. So the android device has to decide to abandon the WiFi. So that L2 handoff delay depends on two fundamental things. One is how long does user spend on moving. The other is how often does the android device scan WiFi and trying to connect Uniwide. In the final version of the android program, we use the android device default WiFi setting, so that we can get the same responses in different tests.

Discussion with other teams

After discussion with other teams we recognized that not all teams got the same results. Some teams who used old versions of android devices get totally different results in L3 and L2 handoff parts. Their results shows that their android devices were disconnecting and then reconnecting to Uniwide during AP changing processes. And because of that, TCP connection was disconnected in this situation. So we speculate that Uniwide AP or new version android devices may have new function which can switch from low signal AP to strong signal AP without TCP reconnection.

Conclusion

The conclusion is based on the data and analyze our team has made by using our own program and SAMSUNG A5(2017) with Android 7.0. The conclusion ignore events which may happens less than 1 millisecond. There are 2 main parts to this conclusion.

Link layer

1. More AP can be found if more people are active near the check point.
2. AP works for Uniwide always use 802.11n with 5Ghz or 2.4Ghz. A few of them use 802.11a or 802.11g with 2.4Ghz.
3. Most area of the campus can receive the Uniwide signal and weak signal can always be find at the edge of the campus.

Mobility

1. A subnet IP of a device will not change with in an expiration period. And the expiry period is larger than 20 hours. So that most people aren't worried about low internet connection due to changing IP address frequency.
2. TCP connection will not disconnects if the device is in the Uniwide environment. It only breaks when the device can not receive the WiFi signal or the signal is not strong enough.
3. Changing AP will not cause internet connection issues. Either device or Uniwide server can modify the android device to a better AP proactively. But this function may be valid only if both of devices support it.
4. Because of previous conclusions, L2 and L3 delay only depends on how fast if user can move from a non Uniwide cover area to somewhere that can receive Uniwide, and how fast the user device finds and connects to a known WiFi.

Reference

Blog.csdn.net. (2017). 【Java TCP/IP Socket】TCP Socket. [online] Available at: http://blog.csdn.net/ns_code/article/details/14105457 [Accessed 12 Oct. 2017].

Developer.android.com. (2017). Android Developers. [online] Available at: https://developer.android.com/index.html [Accessed 12 Oct. 2017].

Moodle.telt.unsw.edu.au. (2017). COMP4336-5177_00927. [online] Available at: https://moodle.telt.unsw.edu.au/course/view.php?id=27663 [Accessed 12 Oct. 2017].

Appendix

Table 1. Link layer data

Check Point	TimePointer	Frequency(GHz)	Speed(Mbps)	Protocol	Signal Strength(db)	AP Density
0	09:30:27	5.745	150	n	-61	77
1	09:31:18	2.412	72	n	-64	84
2	09:31:50	2.412	65	n	-80	231
3	09:33:02	2.412	65	n	-71	231
4	09:33:50	2.462	43	g	-71	272
5	09:34:31	2.437	57	n	-66	127
6	09:35:02	2.437	14	g	-80	123
7	09:35:58	-0.001	-1	null	-127	158
8	09:37:02	5.66	45	a	-79	113
9	09:38:16	5.18	45	a	-89	80
10	09:40:09	5.24	90	n	-80	98
11	09:41:24	5.22	30	a	-78	246
12	09:42:18	-0.001	-1	null	-127	117
13	09:43:44	5.54	15	a	-91	55
14	09:44:51	5.52	30	a	-76	67
15	09:46:52	5.32	81	n	-74	289
16	09:47:15	5.32	60	n	-87	174
17	09:48:15	5.745	120	n	-66	123
18	09:48:45	5.66	90	n	-67	79
19	09:51:17	2.412	57	n	-71	183
20	09:51:55	2.412	7	a	-82	253
21	09:52:51	5.785	81	n	-72	123
22	09:53:31	5.66	45	a	-81	382
23	09:54:27	5.22	60	n	-90	81
24	09:55:01	5.745	200	n	-54	77
25	09:55:57	2.437	72	n	-59	72
26	09:56:35	2.412	72	n	-55	91

Table 2. Reorganized mobility analysis data

Check Point	Classification	TimePointer	Comment	Period	Previous Condition	Current Condition
0	L2 BSSID	09:30:10	1st Time Connect Uniwide	-1	null	00:00:00:00:00:00
0	L2 BSSID	09:30:11	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	7c:0e:ce:de:44:ae
0	L4 TCP	09:30:11	keep alive!			
0	L3 IP	09:30:18	1st Time Connect Uniwide	-1	0.0.0.0	10.249.16.207
1	L2 BSSID	09:31:04	Connected Other AP without Disconnecting	0	7c:0e:ce:de:44:ae	08:cc:68:b5:4b:de
1	L4 TCP	09:31:04	keep alive!			
1	L2 BSSID	09:31:07	Connected Other AP without Disconnecting	0	08:cc:68:b5:4b:de	08:cc:68:b4:fb:3e
1	L4 TCP	09:31:07	keep alive!			
1	L2 BSSID	09:31:17	Connected Other AP without Disconnecting	0	08:cc:68:b4:fb:3e	54:78:1a:bf:62:01
1	L4 TCP	09:31:17	keep alive!			
3	L2 BSSID	09:31:50	Connected Other AP without Disconnecting	0	54:78:1a:bf:62:01	1c:e6:c7:3e:1c:11
3	L4 TCP	09:31:50	keep alive!			
3	L2 BSSID	09:32:17	Connected Other AP without Disconnecting	0	1c:e6:c7:3e:1c:11	54:78:1a:bf:71:5e
3	L4 TCP	09:32:17	keep alive!			
3	L2 BSSID	09:32:30	Connected Other AP without Disconnecting	0	54:78:1a:bf:71:5e	08:cc:68:b4:87:de
3	L4 TCP	09:32:30	keep alive!			
3	L2 BSSID	09:32:36	Connected Other AP without Disconnecting	0	08:cc:68:b4:87:de	08:cc:68:b4:7e:41
3	L4 TCP	09:32:36	keep alive!			
3	L2 BSSID	09:32:41	Connected Other AP without Disconnecting	0	08:cc:68:b4:7e:41	08:cc:68:b4:87:d1
3	L4 TCP	09:32:41	keep alive!			
4	L2 BSSID	09:33:30	Connected Other AP without Disconnecting	0	08:cc:68:b4:87:d1	50:06:04:2f:6b:21
4	L4 TCP	09:33:30	keep alive!			
4	L2 BSSID	09:33:35	Connected Other AP without Disconnecting	0	50:06:04:2f:6b:21	f8:c2:88:c1:26:31
4	L4 TCP	09:33:35	keep alive!			
5	L2 BSSID	09:34:07	Connected Other AP without Disconnecting	0	f8:c2:88:c1:26:31	f8:c2:88:c1:27:31
5	L4 TCP	09:34:07	keep alive!			
7	L3 IP	09:35:19	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
7	L2 BSSID	09:35:19	Disconnected because of low signal	-1	f8:c2:88:c1:27:31	null
7	L4 TCP	09:35:19	TCP break			
8	L2 BSSID	09:36:38	Reconnected Uniwide	78741	null	00:00:00:00:00:00
8	L2 BSSID	09:36:38	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	50:06:04:2f:5a:af
8	L4 TCP	09:36:38	keep alive!			
8	L3 IP	09:36:39	Reconnected	80175	0.0.0.0	10.249.16.207
8	L4 TCP	09:36:39	connected with different socket!			
9	L2 BSSID	09:37:09	Connected Other AP without Disconnecting	0	50:06:04:2f:5a:af	08:cc:68:b4:b6:1f
9	L4 TCP	09:37:09	keep alive!			
9	L2 BSSID	09:37:21	Connected Other AP without Disconnecting	0	08:cc:68:b4:b6:1f	08:cc:68:b4:8e:1f
9	L4 TCP	09:37:21	keep alive!			
9	L2 BSSID	09:37:35	Connected Other AP without Disconnecting	0	08:cc:68:b4:8e:1f	08:cc:68:b4:92:1f
9	L4 TCP	09:37:35	keep alive!			
9	L2 BSSID	09:37:42	Connected Other AP without Disconnecting	0	08:cc:68:b4:92:1f	08:cc:68:b4:a5:8f
9	L4 TCP	09:37:42	keep alive!			
9	L2 BSSID	09:38:00	Connected Other AP without Disconnecting	0	08:cc:68:b4:a5:8f	08:cc:68:b4:9a:cf
9	L4 TCP	09:38:00	keep alive!			
10	L2 BSSID	09:38:42	Connected Other AP without Disconnecting	0	08:cc:68:b4:9a:cf	08:cc:68:b5:d2:6f
10	L4 TCP	09:38:42	keep alive!			
10	L2 BSSID	09:38:49	Connected Other AP without Disconnecting	0	08:cc:68:b5:d2:6f	08:cc:68:b4:9b:2e
10	L4 TCP	09:38:49	keep alive!			
10	L2 BSSID	09:39:20	Connected Other AP without Disconnecting	0	08:cc:68:b4:9b:2e	58:97:1e:56:48:9f
10	L4 TCP	09:39:20	keep alive!			
10	L2 BSSID	09:39:47	Connected Other AP without Disconnecting	0	58:97:1e:56:48:9f	54:78:1a:bf:ef:8e
10	L4 TCP	09:39:47	keep alive!			

11	L2 BSSID	09:40:29	Connected Other AP without Disconnecting	0	54:78:1a:bf:ef:8e	08:cc:68:cc:3b:8e
11	L4 TCP	09:40:29	keep alive!			
11	L3 IP	09:40:45	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
11	L2 BSSID	09:40:45	Disconnected because of low signal	-1	08:cc:68:cc:3b:8e	null
11	L4 TCP	09:40:45	TCP break			
11	L2 BSSID	09:40:47	Reconnected Uniwide	1359	null	00:00:00:00:00:00
11	L2 BSSID	09:41:12	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	58:97:1e:56:56:26
11	L4 TCP	09:41:12	keep alive!			
11	L3 IP	09:41:13	Reconnected	27675	0.0.0.0	10.249.16.207
11	L4 TCP	09:41:13	connected with different socket!			
12	L2 BSSID	09:41:28	Connected Other AP without Disconnecting	0	58:97:1e:56:56:26	d0:72:dc:5c:0a:df
12	L4 TCP	09:41:28	keep alive!			
12	L2 BSSID	09:41:34	Connected Other AP without Disconnecting	0	d0:72:dc:5c:0a:df	d0:72:dc:53:5e:af
12	L4 TCP	09:41:34	keep alive!			
12	L3 IP	09:42:18	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
12	L2 BSSID	09:42:18	Disconnected because of low signal	-1	d0:72:dc:53:5e:af	null
12	L4 TCP	09:42:18	TCP break			
13	L2 BSSID	09:42:29	Reconnected Uniwide	11556	null	00:00:00:00:00:00
13	L2 BSSID	09:42:39	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b4:af:4f
13	L4 TCP	09:42:39	keep alive!			
13	L3 IP	09:42:40	Reconnected	22891	0.0.0.0	10.249.16.207
13	L4 TCP	09:42:40	connected with different socket!			
14	L2 BSSID	09:44:32	Connected Other AP without Disconnecting	0	08:cc:68:b4:af:4f	7c:0e:ce:dc:5d:0e
14	L4 TCP	09:44:32	keep alive!			
15	L2 BSSID	09:45:46	Connected Other AP without Disconnecting	0	7c:0e:ce:dc:5d:0e	f0:9e:63:14:a9:21
15	L4 TCP	09:45:46	keep alive!			
15	L2 BSSID	09:46:12	Connected Other AP without Disconnecting	0	f0:9e:63:14:a9:21	f0:9e:63:1c:b4:61
15	L4 TCP	09:46:12	keep alive!			
15	L2 BSSID	09:46:49	Connected Other AP without Disconnecting	0	f0:9e:63:1c:b4:61	1c:e6:c7:0d:56:ee
15	L4 TCP	09:46:49	keep alive!			
17	L2 BSSID	09:47:21	Disconnected because of low signal	-1	1c:e6:c7:0d:56:ee	null
17	L3 IP	09:47:22	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
17	L4 TCP	09:47:22	TCP break			
17	L2 BSSID	09:47:23	Reconnected Uniwide	1529	null	00:00:00:00:00:00
17	L2 BSSID	09:48:09	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b4:8c:fa
17	L4 TCP	09:48:09	keep alive!			
17	L3 IP	09:48:10	Reconnected	48299	0.0.0.0	10.249.16.207
17	L4 TCP	09:48:10	connected with different socket!			
17	L2 BSSID	09:48:15	Connected Other AP without Disconnecting	0	08:cc:68:b4:8c:fa	08:cc:68:b5:86:8e
17	L4 TCP	09:48:15	keep alive!			
18	L2 BSSID	09:48:39	Connected Other AP without Disconnecting	0	08:cc:68:b5:86:8e	08:cc:68:b4:8c:1e
18	L4 TCP	09:48:39	keep alive!			
19	L2 BSSID	09:48:56	Connected Other AP without Disconnecting	0	08:cc:68:b4:8c:1e	08:cc:68:b5:8f:be
19	L4 TCP	09:48:56	keep alive!			
19	L2 BSSID	09:49:02	Connected Other AP without Disconnecting	0	08:cc:68:b5:8f:be	d0:72:dc:04:b9:0e
19	L4 TCP	09:49:02	keep alive!			
19	L3 IP	09:49:24	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
19	L2 BSSID	09:49:24	Disconnected because of low signal	-1	d0:72:dc:04:b9:0e	null
19	L4 TCP	09:49:24	TCP break			
19	L3 IP	09:49:26	Reconnected	1753	0.0.0.0	10.249.16.207
19	L2 BSSID	09:49:26	Reconnected Uniwide	1565	null	00:00:00:00:00:00
19	L2 BSSID	09:49:26	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b5:f4:6e
19	L4 TCP	09:49:26	keep alive!			
19	L4 TCP	09:49:26	connected with different socket!			
19	L2 BSSID	09:49:48	Connected Other AP without Disconnecting	0	08:cc:68:b5:f4:6e	08:cc:68:b4:bf:fe

19	L4 TCP	09:49:48	keep alive!			
19	L2 BSSID	09:50:11	Connected Other AP without Disconnecting	0	08:cc:68:b4:bf:fe	08:cc:68:b5:8c:7e
19	L4 TCP	09:50:11	keep alive!			
19	L2 BSSID	09:50:34	Connected Other AP without Disconnecting	0	08:cc:68:b5:8c:7e	08:cc:68:b5:db:be
19	L4 TCP	09:50:34	keep alive!			
19	L2 BSSID	09:51:04	Connected Other AP without Disconnecting	0	08:cc:68:b5:db:be	08:cc:68:b4:9e:3e
19	L4 TCP	09:51:04	keep alive!			
19	L2 BSSID	09:51:09	Connected Other AP without Disconnecting	0	08:cc:68:b4:9e:3e	d0:72:dc:3b:01:31
19	L4 TCP	09:51:09	keep alive!			
21	L2 BSSID	09:51:58	Connected Other AP without Disconnecting	0	d0:72:dc:3b:01:31	f8:c2:88:bd:f1:de
21	L4 TCP	09:51:58	keep alive!			
21	L2 BSSID	09:52:28	Connected Other AP without Disconnecting	0	f8:c2:88:bd:f1:de	08:cc:68:b5:8c:7e
21	L4 TCP	09:52:28	keep alive!			
21	L2 BSSID	09:52:42	Connected Other AP without Disconnecting	0	08:cc:68:b5:8c:7e	1c:e6:c7:f0:21:3e
21	L4 TCP	09:52:42	keep alive!			
21	L2 BSSID	09:52:49	Connected Other AP without Disconnecting	0	1c:e6:c7:f0:21:3e	54:78:1a:bf:70:ee
21	L4 TCP	09:52:49	keep alive!			
22	L2 BSSID	09:53:02	Connected Other AP without Disconnecting	0	54:78:1a:bf:70:ee	08:cc:68:b5:f4:61
22	L4 TCP	09:53:02	keep alive!			
22	L2 BSSID	09:53:15	Connected Other AP without Disconnecting	0	08:cc:68:b5:f4:61	08:cc:68:b5:30:9e
22	L4 TCP	09:53:15	keep alive!			
23	L2 BSSID	09:53:33	Connected Other AP without Disconnecting	0	08:cc:68:b5:30:9e	08:cc:68:b5:ea:fe
23	L4 TCP	09:53:33	keep alive!			
23	L2 BSSID	09:53:44	Connected Other AP without Disconnecting	0	08:cc:68:b5:ea:fe	08:cc:68:b5:f7:3e
23	L4 TCP	09:53:44	keep alive!			
23	L2 BSSID	09:54:02	Connected Other AP without Disconnecting	0	08:cc:68:b5:f7:3e	08:cc:68:b5:8f:be
23	L4 TCP	09:54:02	keep alive!			
23	L2 BSSID	09:54:25	Connected Other AP without Disconnecting	0	08:cc:68:b5:8f:be	1c:e6:c7:0d:9d:9e
23	L4 TCP	09:54:25	keep alive!			
24	L2 BSSID	09:54:32	Connected Other AP without Disconnecting	0	1c:e6:c7:0d:9d:9e	54:78:1a:bf:7e:26
24	L4 TCP	09:54:32	keep alive!			
24	L2 BSSID	09:54:45	Connected Other AP without Disconnecting	0	54:78:1a:bf:7e:26	d0:72:dc:5c:08:fe
24	L4 TCP	09:54:45	keep alive!			
25	L2 BSSID	09:55:25	Connected Other AP without Disconnecting	0	d0:72:dc:5c:08:fe	08:cc:68:b5:d7:ae
25	L4 TCP	09:55:25	keep alive!			
25	L2 BSSID	09:55:37	Connected Other AP without Disconnecting	0	08:cc:68:b5:d7:ae	54:78:1a:bf:a7:d1
25	L4 TCP	09:55:37	keep alive!			
26	L2 BSSID	09:56:20	Connected Other AP without Disconnecting	0	54:78:1a:bf:a7:d1	1c:e6:c7:3e:39:11
26	L4 TCP	09:56:20	keep alive!			

Table 3. L2 handoff data

Check Point	Classification	TimePointer	Comment	L2Period(ms	currentBSSID	tempBSSID
0	L2 BSSID	09:30:10	1st Time Connect Uniwide	-1	null	00:00:00:00:00:00
0	L2 BSSID	09:30:11	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	7c:0e:ce:de:44:ae
1	L2 BSSID	09:31:04	Connected Other AP without Disconnecting	0	7c:0e:ce:de:44:ae	08:cc:68:b5:4b:de
1	L2 BSSID	09:31:07	Connected Other AP without Disconnecting	0	08:cc:68:b5:4b:de	08:cc:68:b4:fb:3e
1	L2 BSSID	09:31:17	Connected Other AP without Disconnecting	0	08:cc:68:b4:fb:3e	54:78:1a:bf:62:01
3	L2 BSSID	09:31:50	Connected Other AP without Disconnecting	0	54:78:1a:bf:62:01	1c:e6:c7:3e:1c:11
3	L2 BSSID	09:32:17	Connected Other AP without Disconnecting	0	1c:e6:c7:3e:1c:11	54:78:1a:bf:71:5e
3	L2 BSSID	09:32:30	Connected Other AP without Disconnecting	0	54:78:1a:bf:71:5e	08:cc:68:b4:87:de
3	L2 BSSID	09:32:36	Connected Other AP without Disconnecting	0	08:cc:68:b4:87:de	08:cc:68:b4:7e:41
3	L2 BSSID	09:32:41	Connected Other AP without Disconnecting	0	08:cc:68:b4:7e:41	08:cc:68:b4:87:d1
4	L2 BSSID	09:33:30	Connected Other AP without Disconnecting	0	08:cc:68:b4:87:d1	50:06:04:2f:6b:21
4	L2 BSSID	09:33:35	Connected Other AP without Disconnecting	0	50:06:04:2f:6b:21	f8:c2:88:c1:26:31
5	L2 BSSID	09:34:07	Connected Other AP without Disconnecting	0	f8:c2:88:c1:26:31	f8:c2:88:c1:27:31
7	L2 BSSID	09:35:19	Disconnected because of low signal	-1	f8:c2:88:c1:27:31	null
8	L2 BSSID	09:36:38	Reconnected Uniwide	78741	null	00:00:00:00:00:00
8	L2 BSSID	09:36:38	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	50:06:04:2f:5a:af
9	L2 BSSID	09:37:09	Connected Other AP without Disconnecting	0	50:06:04:2f:5a:af	08:cc:68:b4:b6:1f
9	L2 BSSID	09:37:21	Connected Other AP without Disconnecting	0	08:cc:68:b4:b6:1f	08:cc:68:b4:8e:1f
9	L2 BSSID	09:37:35	Connected Other AP without Disconnecting	0	08:cc:68:b4:8e:1f	08:cc:68:b4:92:1f
9	L2 BSSID	09:37:42	Connected Other AP without Disconnecting	0	08:cc:68:b4:92:1f	08:cc:68:b4:a5:8f
9	L2 BSSID	09:38:00	Connected Other AP without Disconnecting	0	08:cc:68:b4:a5:8f	08:cc:68:b4:9a:cf
10	L2 BSSID	09:38:42	Connected Other AP without Disconnecting	0	08:cc:68:b4:9a:cf	08:cc:68:b5:d2:6f
10	L2 BSSID	09:38:49	Connected Other AP without Disconnecting	0	08:cc:68:b5:d2:6f	08:cc:68:b4:9b:2e
10	L2 BSSID	09:39:20	Connected Other AP without Disconnecting	0	08:cc:68:b4:9b:2e	58:97:1e:56:48:9f
10	L2 BSSID	09:39:47	Connected Other AP without Disconnecting	0	58:97:1e:56:48:9f	54:78:1a:bf:ef:8e
11	L2 BSSID	09:40:29	Connected Other AP without Disconnecting	0	54:78:1a:bf:ef:8e	08:cc:68:cc:3b:8e
11	L2 BSSID	09:40:45	Disconnected because of low signal	-1	08:cc:68:cc:3b:8e	null
11	L2 BSSID	09:40:47	Reconnected Uniwide	1359	null	00:00:00:00:00:00
11	L2 BSSID	09:41:12	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	58:97:1e:56:56:26
12	L2 BSSID	09:41:28	Connected Other AP without Disconnecting	0	58:97:1e:56:56:26	d0:72:dc:5c:0a:df
12	L2 BSSID	09:41:34	Connected Other AP without Disconnecting	0	d0:72:dc:5c:0a:df	d0:72:dc:53:5e:af
12	L2 BSSID	09:42:18	Disconnected because of low signal	-1	d0:72:dc:53:5e:af	null
13	L2 BSSID	09:42:29	Reconnected Uniwide	11556	null	00:00:00:00:00:00
13	L2 BSSID	09:42:39	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b4:af:4f
14	L2 BSSID	09:44:32	Connected Other AP without Disconnecting	0	08:cc:68:b4:af:4f	7c:0e:ce:dc:5d:0e
15	L2 BSSID	09:45:46	Connected Other AP without Disconnecting	0	7c:0e:ce:dc:5d:0e	f0:9e:63:14:a9:21
15	L2 BSSID	09:46:12	Connected Other AP without Disconnecting	0	f0:9e:63:14:a9:21	f0:9e:63:1c:b4:61
15	L2 BSSID	09:46:49	Connected Other AP without Disconnecting	0	f0:9e:63:1c:b4:61	1c:e6:c7:0d:56:ee
17	L2 BSSID	09:47:21	Disconnected because of low signal	-1	1c:e6:c7:0d:56:ee	null
17	L2 BSSID	09:47:23	Reconnected Uniwide	1529	null	00:00:00:00:00:00
17	L2 BSSID	09:48:09	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b4:8c:fa
17	L2 BSSID	09:48:15	Connected Other AP without Disconnecting	0	08:cc:68:b4:8c:fa	08:cc:68:b5:86:8e
18	L2 BSSID	09:48:39	Connected Other AP without Disconnecting	0	08:cc:68:b5:86:8e	08:cc:68:b4:8c:1e
19	L2 BSSID	09:48:56	Connected Other AP without Disconnecting	0	08:cc:68:b4:8c:1e	08:cc:68:b5:8f:be
19	L2 BSSID	09:49:02	Connected Other AP without Disconnecting	0	08:cc:68:b5:8f:be	d0:72:dc:04:b9:0e
19	L2 BSSID	09:49:24	Disconnected because of low signal	-1	d0:72:dc:04:b9:0e	null
19	L2 BSSID	09:49:26	Reconnected Uniwide	1565	null	00:00:00:00:00:00
19	L2 BSSID	09:49:26	Connected Other AP without Disconnecting	0	00:00:00:00:00:00	08:cc:68:b5:f4:6e
19	L2 BSSID	09:49:48	Connected Other AP without Disconnecting	0	08:cc:68:b5:f4:6e	08:cc:68:b4:bf:fe
19	L2 BSSID	09:50:11	Connected Other AP without Disconnecting	0	08:cc:68:b4:bf:fe	08:cc:68:b5:8c:7e
19	L2 BSSID	09:50:34	Connected Other AP without Disconnecting	0	08:cc:68:b5:8c:7e	08:cc:68:b5:db:be
19	L2 BSSID	09:51:04	Connected Other AP without Disconnecting	0	08:cc:68:b5:db:be	08:cc:68:b4:9e:3e
19	L2 BSSID	09:51:09	Connected Other AP without Disconnecting	0	08:cc:68:b4:9e:3e	d0:72:dc:3b:01:31
21	L2 BSSID	09:51:58	Connected Other AP without Disconnecting	0	d0:72:dc:3b:01:31	f8:c2:88:bd:f1:de
21	L2 BSSID	09:52:28	Connected Other AP without Disconnecting	0	f8:c2:88:bd:f1:de	08:cc:68:b5:8c:7e
21	L2 BSSID	09:52:42	Connected Other AP without Disconnecting	0	08:cc:68:b5:8c:7e	1c:e6:c7:f0:21:3e
21	L2 BSSID	09:52:49	Connected Other AP without Disconnecting	0	1c:e6:c7:f0:21:3e	54:78:1a:bf:70:ee
22	L2 BSSID	09:53:02	Connected Other AP without Disconnecting	0	54:78:1a:bf:70:ee	08:cc:68:b5:f4:61
22	L2 BSSID	09:53:15	Connected Other AP without Disconnecting	0	08:cc:68:b5:f4:61	08:cc:68:b5:30:9e
23	L2 BSSID	09:53:33	Connected Other AP without Disconnecting	0	08:cc:68:b5:30:9e	08:cc:68:b5:ea:fe
23	L2 BSSID	09:53:44	Connected Other AP without Disconnecting	0	08:cc:68:b5:ea:fe	08:cc:68:b5:f7:3e
23	L2 BSSID	09:54:02	Connected Other AP without Disconnecting	0	08:cc:68:b5:f7:3e	08:cc:68:b5:8f:be
23	L2 BSSID	09:54:25	Connected Other AP without Disconnecting	0	08:cc:68:b5:8f:be	1c:e6:c7:0d:9d:9e
24	L2 BSSID	09:54:32	Connected Other AP without Disconnecting	0	1c:e6:c7:0d:9d:9e	54:78:1a:bf:7e:26
24	L2 BSSID	09:54:45	Connected Other AP without Disconnecting	0	54:78:1a:bf:7e:26	d0:72:dc:5c:08:fe
25	L2 BSSID	09:55:25	Connected Other AP without Disconnecting	0	d0:72:dc:5c:08:fe	08:cc:68:b5:d7:ae
25	L2 BSSID	09:55:37	Connected Other AP without Disconnecting	0	08:cc:68:b5:d7:ae	54:78:1a:bf:a7:d1
26	L2 BSSID	09:56:20	Connected Other AP without Disconnecting	0	54:78:1a:bf:a7:d1	1c:e6:c7:3e:39:11

Table 4. L3handoff data

Check Point	Classification	TimePointer	Comment	L3Period	currentIP	tempIP
0	L3 IP	09:30:18	1st Time Connect Uniwide	-1	0.0.0.0	10.249.16.207
7	L3 IP	09:35:19	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
8	L3 IP	09:36:39	Reconnected	80175	0.0.0.0	10.249.16.207
11	L3 IP	09:40:45	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
11	L3 IP	09:41:13	Reconnected	27675	0.0.0.0	10.249.16.207
12	L3 IP	09:42:18	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
13	L3 IP	09:42:40	Reconnected	22891	0.0.0.0	10.249.16.207
17	L3 IP	09:47:22	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
17	L3 IP	09:48:10	Reconnected	48299	0.0.0.0	10.249.16.207
19	L3 IP	09:49:24	Disconnected Because of Low Signal	-1	10.249.16.207	0.0.0.0
19	L3 IP	09:49:26	Reconnected	1753	0.0.0.0	10.249.16.207

Table 5. L4 connectivity

Check Point	Classification	TimePointer	Comment
0	L4 TCP	09:30:11	keep alive!
1	L4 TCP	09:31:04	keep alive!
1	L4 TCP	09:31:07	keep alive!
1	L4 TCP	09:31:17	keep alive!
3	L4 TCP	09:31:50	keep alive!
3	L4 TCP	09:32:17	keep alive!
3	L4 TCP	09:32:30	keep alive!
3	L4 TCP	09:32:36	keep alive!
3	L4 TCP	09:32:41	keep alive!
4	L4 TCP	09:33:30	keep alive!
4	L4 TCP	09:33:35	keep alive!
5	L4 TCP	09:34:07	keep alive!
7	L4 TCP	09:35:19	TCP break
8	L4 TCP	09:36:38	keep alive!
8	L4 TCP	09:36:39	connected with different socket!
9	L4 TCP	09:37:09	keep alive!
9	L4 TCP	09:37:21	keep alive!
9	L4 TCP	09:37:35	keep alive!
9	L4 TCP	09:37:42	keep alive!
9	L4 TCP	09:38:00	keep alive!
10	L4 TCP	09:38:42	keep alive!
10	L4 TCP	09:38:49	keep alive!
10	L4 TCP	09:39:20	keep alive!
10	L4 TCP	09:39:47	keep alive!
11	L4 TCP	09:40:29	keep alive!
11	L4 TCP	09:40:45	TCP break
11	L4 TCP	09:41:12	keep alive!
11	L4 TCP	09:41:13	connected with different socket!
12	L4 TCP	09:41:28	keep alive!
12	L4 TCP	09:41:34	keep alive!
12	L4 TCP	09:42:18	TCP break
13	L4 TCP	09:42:39	keep alive!
13	L4 TCP	09:42:40	connected with different socket!
14	L4 TCP	09:44:32	keep alive!
15	L4 TCP	09:45:46	keep alive!
15	L4 TCP	09:46:12	keep alive!
15	L4 TCP	09:46:49	keep alive!
17	L4 TCP	09:47:22	TCP break
17	L4 TCP	09:48:09	keep alive!
17	L4 TCP	09:48:10	connected with different socket!
17	L4 TCP	09:48:15	keep alive!
18	L4 TCP	09:48:39	keep alive!
19	L4 TCP	09:48:56	keep alive!
19	L4 TCP	09:49:02	keep alive!
19	L4 TCP	09:49:24	TCP break
19	L4 TCP	09:49:26	keep alive!
19	L4 TCP	09:49:26	connected with different socket!
19	L4 TCP	09:49:48	keep alive!
19	L4 TCP	09:50:11	keep alive!
19	L4 TCP	09:50:34	keep alive!
19	L4 TCP	09:51:04	keep alive!
19	L4 TCP	09:51:09	keep alive!
21	L4 TCP	09:51:58	keep alive!
21	L4 TCP	09:52:28	keep alive!
21	L4 TCP	09:52:42	keep alive!
21	L4 TCP	09:52:49	keep alive!
22	L4 TCP	09:53:02	keep alive!
22	L4 TCP	09:53:15	keep alive!
23	L4 TCP	09:53:33	keep alive!
23	L4 TCP	09:53:44	keep alive!
23	L4 TCP	09:54:02	keep alive!
23	L4 TCP	09:54:25	keep alive!
24	L4 TCP	09:54:32	keep alive!
24	L4 TCP	09:54:45	keep alive!
25	L4 TCP	09:55:25	keep alive!
25	L4 TCP	09:55:37	keep alive!
26	L4 TCP	09:56:20	keep alive!