Implementation of PingER on Android Mobile Devices Using Firebase

Ananthnarayan Rajappa
Department of CSE, ASET
Amity University
Noida, UP, India
ananthnarayan.rajappa@yahoo.com

Prof (Dr.) A. Sai Sabitha
Department of CSE, ASET
Amity University
Noida, UP, India
assabhith@amity.edu

Prof(Dr) Bebo White
SLAC National Accelerator Laboratory
Stanford, CA, USA
bebo@slac.stanford.edu

Aayush Upadhyay
Department of CSE, ASET
Amity University
Noida, UP, India
aayushupadhyay.au5@gmail.com

Prof (Dr) Abhay Bansal
Department of CSE, ASET
Amity University
Noida, UP, India
abansal1@amity.edu

Prof(Dr) Les Cottrell
SLAC National Accelerator Laboratory
Stanford, CA, USA
cottrell@slac.stanford.edu

***Abstract* - PingER (Ping End-to-End Reporting) is a tool developed by SLAC National Accelerator Laboratory for the purpose of Internet End-to-end Performance Monitoring (IEPM). The aim of this research work is to develop a mobile application for Android mobile devices using Firebase for storing the data, obtained from pinging the beacons, and authenticating the users. The Measuring Agent (MA) pings the beacon list, the data obtained is formatted with the help of a Regular Expression library before being pushed to Firebase. In addition, the location of the MA, latitude and longitude, is also tracked with the help of Google's Geolocation API. This data is also stored in the database.**

Keywords - PingER, Firebase, Android, Regex, Geolocation Application Programming Interface (API)

I. INTRODUCTION

The PingER project has over 700 sites worldwide which it monitors for the purpose of measuring the Digital Divide across different regions of the world [5][6][7]. 'Digital Divide' refers to the discrepancy between regions which have access to communication technologies and information, and those that don't.

The objective of this research work was to develop an android application and organize the data output in a pragmatic manner so that it can be studied and analyzed easily. The model proposed by Jain, David et al involved SLAC extracting the ping data from text files generated by Measuring Agents (MAs) around the world [11]. Instead if MAs sent their data to a real time and cloud hosted platform, it would be easier for SLAC to access the data obtained from various locations at their convenience. The proposed model involves the android mobile application using the services provided by Firebase. Firebase is a toolkit, powered by Google, that provides services ranging from analytics, databases, authentication etc. It is a reliable and trustworthy platform used by well known companies such as Alibaba, New York Times, Duolingo, Shazam etc. Firebase Realtime Database has been used in the proposed model to store the data obtained from pinging the beacons and the location data of the MAs. In addition the application also uses Firebase Authentication so that users around the world can perform secure logins to the application.

II. LITERATURE SURVEY

A. *PingER*

PingER stands for Ping End-to-End Reporting. It is a project that was initiated in January 1995 and is led by SLAC National Accelerator Laboratory [1][2][3]. The purpose of this project is to measure the round trip time between nodes sending data packets over the internet. Internet Control Message Protocol (ICMP) Echo is the main mechanism that is adopted- a packet of user selected length is sent to a remote host and is echoed back. PingER collects information such as latency, packet loss and jitters. PingER is able to determine the quality of internet in various regions around the world with this data [4]. A file called pinger.xml contains a set of beacons or remote hosts to ping [12]. There are beacons pinged from over 700 monitoring sites across 160 countries.

B. *Android*

Android is an operating system based on the linux kernel. It was developed predominantly for touchscreen mobile devices such as tablets and smartphones. It was built initially by Android Inc however it was acquired by Google in 2005 and launched by Google itself in 2007.

Java is the main language through which android code is written. Due to low investment and high Return On Investment developers prefer Android over other popular smartphone operating systems. Furthermore, since 2011, Android has been the most popular and top-selling OS on smartphones with a user base of over 1 billion.

C. *Regex*

Regex is short for Regular Expression and it is a tool that is used for pattern matching and also in 'find and replace' operations. A regular expression can be defined as a sequence of characters that define a search pattern. We use the regex library in Java to extract useful information, such as the round trip time, minimum response time, average response time from the data obtained from pinging the beacons.

Many programming languages provide regex either built-in or via an external library. A regex processor is able to translate a regular expression into an internal representation which is executed and matched against a string.

D. *Firebase*

Firebase is a platform for mobile and web application development that provides services and tools needed to develop a successful application [8][9]. With the help of Firebase you can improve the quality of your app by using services such as crashlytics, performance monitoring, test lab, realtime database, authentication and many more [10]. The platform is run by Google and trusted by top apps on the Playstore such as The New York Times, Shazam and Duolingo.

Firebase Realtime database and Firebase Authentication is used in the application to store the ping data and secure our application. Firebase Realtime database displays data in JSON format. It is a cloud-hosted database and one of the advantages of using this tool of Firebase is that the data is synchronized across all clients in realtime and is even available when the app goes offline. On the other hand Firebase Authentication, provides ways to authenticate users and thereby provide customized services. In the proposed application the users enter their mobile number to authenticate themselves.

III. WORKFLOW OF EXISTING MODEL

The existing model works as illustrated in Fig 1. The application that was developed ran on both rooted and non-rooted devices. The application launches and executes itself on the device to run services such as pinging and updating the beacon list.

The application was found to be largely unsuccessful because it used to track location continuously instead of every 30 minutes, thus draining the device's battery and eating the system resources quickly. The application also failed to auto start in case the user closed it. Also, transferring files every day is a very hectic process. In case the files failed to transfer and the application's memory is wiped out the files will be lost completely. This model was not feasible because of large amount of data that it failed to capture in case the application is not running as it failed to auto start. The application's working is show in the following diagrams.

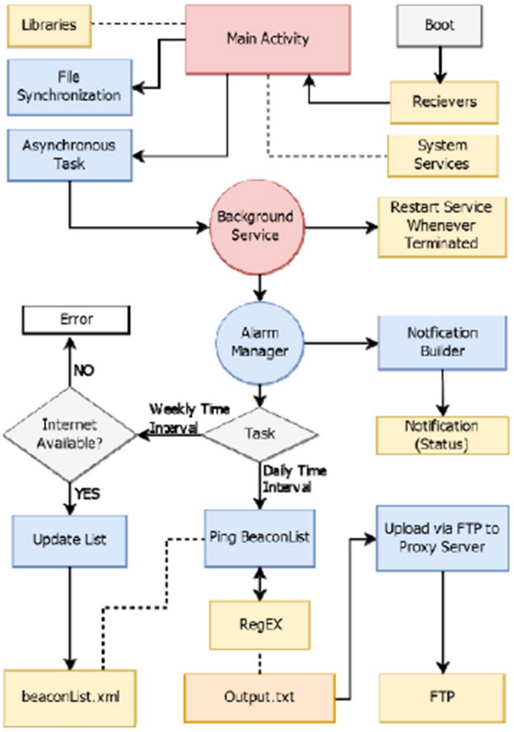


Fig 1. Workflow of Model developed by A. Jain and J. David [11]

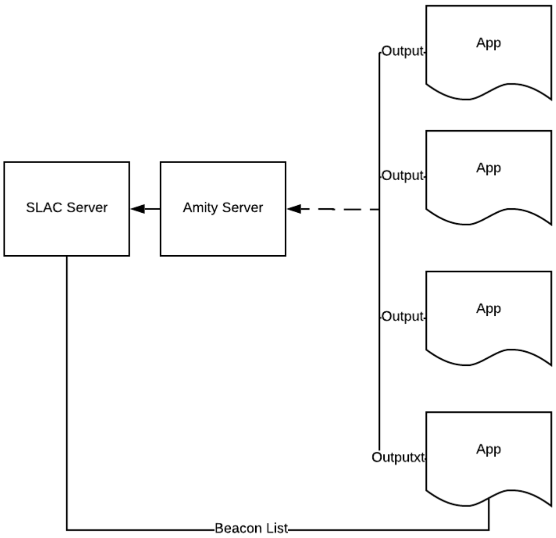
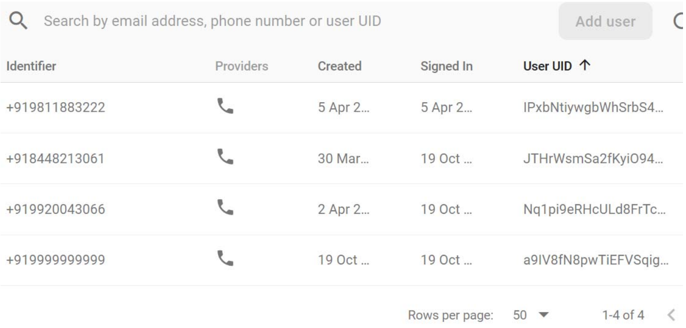


Fig 2. Graphical Representation of Model developed by A. Jain and J. David [11]

IV. PROPOSED MODEL

The model which we propose uses Firebase Realtime database and authentication. The high-level workflow is depicted in Fig. 5:

1. The user enters their mobile number. Firebase Authentication verifies if the number is valid and then adds the number to its database if the user is new. A glimpse of the database is shown in Fig 3.



Identifier	Providers	Created	Signed In	User UID ↑
+919811883222	☎	5 Apr 2...	5 Apr 2...	IPxbNtiywgbWhSrbS4...
+918448213061	☎	30 Mar...	19 Oct ...	JTHrWsmSa2fKyiO94...
+919920043066	☎	2 Apr 2...	19 Oct ...	Nq1pi9eRHcULd8FrTc...
+919999999999	☎	19 Oct ...	19 Oct ...	a9IV8fN8pwTiEFVSqig...

Rows per page: 50 1-4 of 4

Fig 3. Authentication database

2. Firebase Authentication was used to authenticate the user using a mobile number.

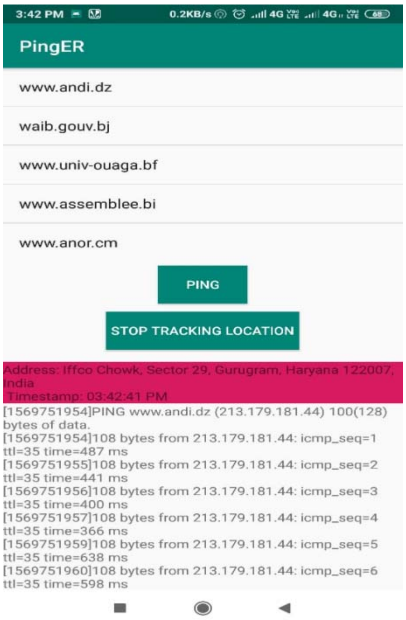
3. A one-Time password (OTP) is generated and a message is sent to the mobile number that the MA entered. The user enters the 6 digit code to proceed to the main application.

4. The main layout of the application consists of 3 sections- a recycler view to display the websites, two buttons- one to ping the hardcoded beacons and the other to track the location of the measuring agent and the third section consists of the box where the location of the user and the output of ping commands were shown.

5. Clicking on the "Ping" button will ping each of the beacons on the list. The output is displayed in a Textview which is a child of the Recyclerview as shown in Fig 4. Also, the output is formatted using Regex to obtain the required parameters. The values are then pushed to the Firebase Realtime Database.

6. Clicking the "Start Tracking Location" button the user is asked for the location permission and after the user allows the permission the app starts tracking the user's real-time location. These values are also inserted into the database so the location from where the MA pinged the beacons is known.

7. The "Start Tracking Location" button changes to "Stop Tracking Location" button thus giving the users the option to not to share the location in case they do not want to share the location. Thus, respecting the users' privacy.



www.andi.dz
waib.gouv.bj
www.univ-ouaga.bf
www.assemblee.bi
www.anor.cm

```
Address: Iffco Chowk, Sector 29, Gurugram, Haryana 122007,
India
 Timestamp: 03:42:41 PM
[1569751954]PING www.andi.dz (213.179.181.44) 100(128)
bytes of data.
[1569751954]108 bytes from 213.179.181.44: icmp_seq=1
ttl=35 time=487 ms
[1569751955]108 bytes from 213.179.181.44: icmp_seq=2
ttl=35 time=441 ms
[1569751956]108 bytes from 213.179.181.44: icmp_seq=3
ttl=35 time=400 ms
[1569751957]108 bytes from 213.179.181.44: icmp_seq=4
ttl=35 time=366 ms
[1569751959]108 bytes from 213.179.181.44: icmp_seq=5
ttl=35 time=638 ms
[1569751960]108 bytes from 213.179.181.44: icmp_seq=6
ttl=35 time=598 ms
```

Fig 4. Ping output displayed in the application

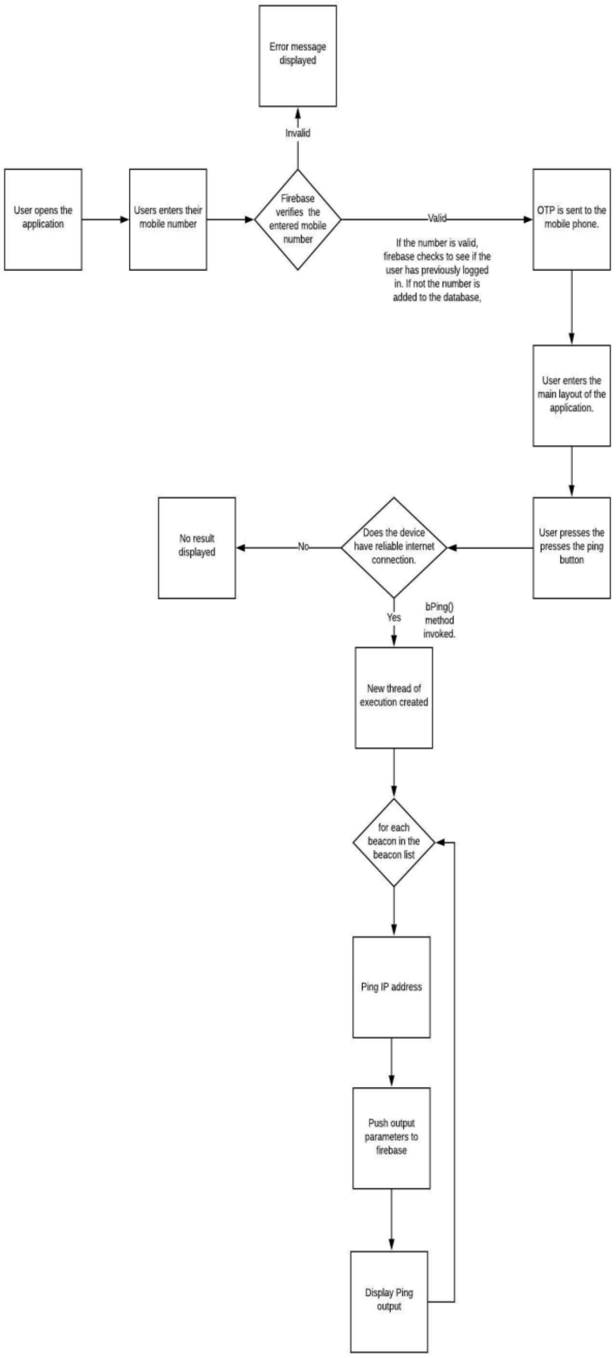

Error message displayed
Invalid
User opens the application
Users enters their mobile number
Firebase verifies the entered mobile number
Valid
If the number is valid, firebase checks to see if the user has previously logged in. If not the number is added to the database,
OTP is sent to the mobile phone.
User enters the main layout of the application.
User presses the presses the ping button
Does the device have reliable internet connection.
No
No result displayed
Yes
bPing() method invoked.
New thread of execution created
for each beacon in the beacon list
Ping IP address
Push output parameters to firebase
Display Ping output

Fig 5. Workflow of New Model

V. COLLECTION OF DATA

Our application pings each of the hardcoded beacons using the command "ping -n -c 10 -w 30 -i 1 -s 100 ". The raw data is collected in a string before regex processing is done to get the required parameters.

Google GeoLocation api is used to get the precise location of the user. With the help of the Geolocation API the location and the accuracy radius, based on information that a mobile client can detect from cell towers and WiFi nodes, can be obtained. Even with GPS and WiFi turned off, the approximate location of the user can be determined as Google uses BSSID information from your WLAN Access Point.

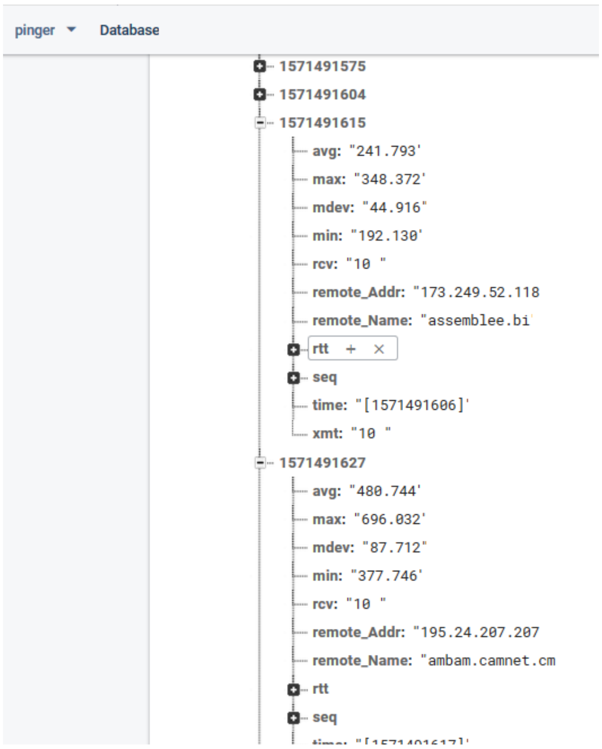
VI. PARSING OF PING DATA

Once the data is collected it needs to be displayed in the correct format. The parameters that need to be recorded by the android application are shown in Table 1

TABLE 1 - PARAMETERS COLLECTED BY THE ANDROID APPLICATION

S.No.	**Parameter**
1	Monitor_Host_Name
2	Monitor_Addr
3	Remote__Name
4	Remote_Addr
5	Bytes
6	Time
7	Xmt
8	Rcv
9	Min
10	Avg
11	Max
12	Seq[1..Rcv]
13	RTT[1..Rcv]

Our application currently collects the following parameters as shown in Fig 6.



```
1571491575
1571491604
1571491615
    avg: "241.793'
    max: "348.372'
    mdev: "44.916"
    min: "192.130'
    rcv: "10 "
    remote_Addr: "173.249.52.118
    remote_Name: "assemblee.bi'
    rtt  +  ×
    seq
    time: "[1571491606]'
    xmt: "10 "
1571491627
    avg: "480.744'
    max: "696.032'
    mdev: "87.712"
    min: "377.746'
    rcv: "10 "
    remote_Addr: "195.24.207.207
    remote_Name: "ambam.camnet.cm
    rtt
    seq
    time: "[1571491617]'
```

Fig 6. Data obtained from pinging beacons as viewed from Firebase

The following functions use a regex pattern to extract the corresponding parameter from the output obtained from pinging the beacon:

1. getRemoteName(): This function returns the name of the remote site. The regex pattern used is "(www.)?([A-Za-z0-9]+\\.)+[A-Za-z]{2,4}"

2. getRemoteAddress(): This function returns the IP address of the remote site. The regex pattern used is "[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}"

3. getNumberOfPacketsSent(String str): The number of packets sent is returned by this function. The regex pattern used is "\\d+ packets"

4. getNumberOfPacketsReceived(String str): This function returns the number of packets received. The regex pattern used is "\\d+ received"

5. getRoundTripTime(String str): This function returns the round trip times of individual responding pings. The regex pattern used is “\[([0-9]{10})\]PING”

6. getMinMaxAverage(String str): This function returns an array containing the minimum, average and maximum response time for packets in ms.

7. countICMP(String str): This function returns an ArrayList containing the sequence number of individual responding pings. The regex pattern used is "icmp_seq=\\d+"

8. timeOfEachICMP(String str): This function returns an ArrayList containing the round trip times of individual responding pings.

VII. SYNCING DATA TO FIREBASE

Once the data is collected and parsed, the next step is to sync it to Firebase Realtime database. The data is appended such that the parent node is the User ID. The User ID is a unique ID given to each user accessing the application. The authentication database in Fig 3. displays this information. Having the User ID as the parent will allow SLAC to easily identify which user has pinged the beacons and from where. The snippet of code shown below illustrates how we implemented this functionality:

```
pingDatabase.child(userId).child(ts).setValue(pingOutputParameters);
```

The parameters acquired after regex processing are stored in an object called pingOutputParameters. When we pass this object in the setValue() method, the data is arranged in key value pairs as shown in Fig. 6.

VIII. CONCLUSION

The aim for developing this application was to simplify the way in which SLAC obtains and analyzes the data. Since the data obtained from MAs around the world would be stored in one place, it should become easier for SLAC to access and process the data. In addition Firebase will make it easy for SLAC to scale in a cost effective manner depending on their needs.

REFERENCES

[1] L. Cottrell, C. Logg and J. Williams, "PingER History and Methodology," in 2003 Round Table on Developing Countries Access to Scientific Knowledge October 23-24, 2003, Trieste, Italy, 2003.

[2] W. Matthews, C. Granieri and L. Cottrell, "International network connectivity and performance, the challenge from high-energy physics," no. SLAC-PUB-8382, March 2000.

[3] Les Cottrell, R., Connie Logg, and Jerrod Williams. "PingER History and Methodology." (2003).

[4] S. M. Khan, L. Cottrell, U. Kalim and A. Ali, "Quantifying the Digital Divide: A Scientific Overview of Network Connectivity and Grid Infrastructure in South Asian Countries," in 16th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2007), Victoria, Canada, 2007.

[5] Cottrell, L. Measuring the digital divide with PingER. No. SLAC-PUB- 10186. Stanford Linear Accelerator Center, Menlo Park, CA (US), 2003.

[6] W. Matthews and L. Cottrell, "The PingER project: active Internet performance monitoring for the HENP community," IEEE Communications Magazine, vol. 38, no. 5, pp. 130-136, May 2000.

[7] L. Cottrell, "How Bad Is Africa's Internet?," IEEE Spectrum, 29 January 2013.

[8] Firebase Realtime Database and Firebase Authentication. Available: https://firebase.google.com/

[9] "Mastering Firebase for Android Development: Build Real-time, Scalable, and Cloud-enabled Android Apps with Firebase" by Ashok Kumar S [Book]

[10] "Firebase Cookbook: Over 70 Recipes to Help You Create Real-time Web and Mobile Applications with Firebase" by Houssem Yahiaoui [Book]

[11] A. Jain, J. David, A. S. Sabitha, L. Cottrell, Bebo White, A. Bansal and R. Bansal, "Extension of the PingER Project onto Mobile Devices using Android Applications" in 9th International Conference on Cloud Computing, Data Science & Engineering - Confluence, Amity University, Noida, India, 2019

[12] Beacon List provided by SLAC. Available: https://www-iepm.slac.stanford.edu/pinger/pinger.xml