

ePingER on Android phone

Introduction

Topher White (son of Bebo White) is convinced that we/he could install the PingER software on the Android phone/OS. He literally has 100's of them that have been donated to his project. The delays will be very different. It will be fascinating to try and understand what parameters determine the delays and losses, quantify the information and compare with wired MAs. Initially the phones would be stationary and use WiFi (thus eliminating roving effects, and cell phone coverage). The power use and availability of smartphones could make it a very interesting platform for PingER.

Measurement Agent

It is basically a cheap, small, low power PingER Measurement Agent (MA) at a fixed location. A difference, however, is that the Android MA would probably not have a fixed wired connection to the network, rather communicating by wireless presumably to some local WiFi router. If the ePingER is to move around it will need to add its location (lat/long) to the data records. This will need to be done in a way that is backward compatible (i.e. the additional can be easily filtered out and ignored by existing applications).

Power

Smart phones are very energy efficient because they are designed to run on a battery for a long period of time. For a smartphone battery with 1650mAh and a voltage of 3.7V (typical cell phone battery voltage 3-5V) the Wattage for an 8 hour talk time on 3G is 0.76 W.

Assuming the smartphone is charged once a day from 0% to 100% [a Galaxy SIII: consumed 12.3 watt-hours to charge, taking 2 hours and 26 minutes](#), the maximum wattage was 6.6 watts, with an average of approximately 5.0 W.

According to [measurements from Lawrence Berkeley National Laboratory](#), the average cell phone draws 3.68 watts of power from the outlet while it's charging and 2.24 watts when charged.

One can [buy off the shelf solar cell phone chargers](#), for example [the SolPro](#) that in direct sunlight can re-charge its battery (a Lithium-Polymer battery with safety circuitry and capacity of 4500mAh enough to keep a smartphone going all day) in 4.5 hours.

Accessing the MA data

This is saved in Firebase. Go to <https://firebase.google.com/> enter the Google account pingeramity123@gmail.com plus the password (Les has the password). Then Go to console, then choose one of the documents (e.g. pinger-7db5c). To the left is a column with items such as Authentication and Database.

- Under Authentication +919999999999 is the guest access.
- The Database provides access to the data, click on ping and the individual records show up. Click on + signs to open each record.

[Video on accessing the data.](#)

This covers accessing the data from Firebase, and how to convert the accessed data from json to text format. The video is [here](#). It is a Windows (.wmv) file. On a Mac you may need to load the vlc app. There is also a copy at https://slac-my.sharepoint.com/personal/cottrell_slac_stanford_edu/_layouts/15/onedrive.aspx?q=Android&id=%2Fpersonal%2Fcottrell%5Fslac%5Fstanford%5Fedu%2FDocuments%2FAttachments%2FPingER%20Android%20Video%2Ewmv&parent=%2Fpersonal%2Fcottrell%5Fslac%5Fstanford%5Fedu%2FDocuments&parentview=7

Format

The format of the PingER text file is described at: [PingER Monitor node format](#). An example of the Amity ePingER format is below. Note it adds a semi-colon (";") followed by the Lat and long space separated. I have modified wrap-analyze-hourly.pl to extract the lat and long and remove the semi colon lat and long (if present) from the records so the rest of the analysis will work for Amity ePingER records.

```

pinger-andl.amity.edu 192.168.0.100 www.andi.dz 213.179.181.44 100 1584526700 10 10 366.348 468.286 1022.335 1
2 3 4 5 6 7 8 9 10 387 410 434 463 380 403 432 1022 366 381 ; 28.5942222 77.29477718
pinger-andl.amity.edu 192.168.0.100 waib.gouv.bj 81.91.232.2 100 1584526739 0 30 ; 28.5942222 77.29477718
pinger-andl.amity.edu 192.168.0.100 www.univ 212.52.131.9 100 1584526739 10 10 342.097 414.403 471.318 1 2 3 4
5 6 7 8 9 10 342 431 458 378 402 423 447 471 389 399 ; 28.5942222 77.29477718
pinger-andl.amity.edu 192.168.0.100 assemblee.bi 173.249.52.118 100 1584526749 10 10 160.730 309.413 998.790 1
2 3 4 5 6 7 8 9 10 160 164 225 227 257 998 206 230 254 368 ; 28.5942222 77.29477718
pinger-andl.amity.edu 192.168.0.100 cmsf.cv 213.150.196.36 100 1584526758 10 10 353.646 530.848 954.487 1 2 3 4
5 6 7 8 9 10 360 695 422 432 353 379 402 924 954 383 ; 28.5942222 77.29477718
pinger-andl.amity.edu 192.168.0.100 www.andi.dz 213.179.181.44 100 1584528880 10 10 354.024 458.614 922.058 1 2
3 4 5 6 7 8 9 10 408 433 456 375 400 922 446 354 376 412 ; 28.5941693 77.2947071
pinger-andl.amity.edu 192.168.0.100 waib.gouv.bj 81.91.232.2 100 1584528920 0 30 ; 28.5941693 77.2947071
pinger-andl.amity.edu 192.168.0.100 www.univ 212.52.131.9 100 1584528920 10 11 358.509 475.244 927.777 1 2 3 4
5 6 8 9 10 11 358 482 402 927 451 472 399 421 458 377 ; 28.5941693 77.2947071
pinger-andl.amity.edu 192.168.0.100 assemblee.bi 173.249.52.118 100 1584528930 10 11 160.074 324.773 967.567 1
2 3 4 5 6 8 9 10 11 251 274 299 219 241 967 298 160 259 276 ; 28.5941693 77.2947071
pinger-andl.amity.edu 192.168.0.100 cmsf.cv 213.150.196.36 100 1584528942 10 10 330.181 489.186 1003.637 1 2 3
4 5 6 7 8 9 10 408 430 351 376 1003 417 330 456 390 725 ; 28.5941693 77.2947071
pinger-andl.amity.edu 192.168.0.100 www.andi.dz 213.179.181.44 100 1584528961 10 10 374.702 423.526 469.632 1 2
3 4 5 6 7 8 9 10 452 374 398 421 445 469 385 400 424 461 ; 28.59418655 77.29478424
pinger-andl.amity.edu 192.168.0.100 waib.gouv.bj 81.91.232.2 100 1584529000 0 30 ; 28.59418655 77.29478424
pinger-andl.amity.edu 192.168.0.100 www.univ 212.52.131.9 100 1584529000 10 13 400.512 463.178 555.022 1 3 4 5
6 7 8 10 12 13 400 464 469 489 425 532 443 420 431 555 ; 28.59418655 77.29478424
pinger-andl.amity.edu 192.168.0.100 assemblee.bi 173.249.52.118 100 1584529013 10 10 155.725 245.308 454.052 1
2 3 4 5 6 7 8 9 10 249 266 291 214 234 155 156 203 226 454 ; 28.59418655 77.29478424
pinger-andl.amity.edu 192.168.0.100 cmsf.cv 213.150.196.36 100 1584529022 10 10 326.824 455.570 1005.206 1 2 3
4 5 6 7 8 9 10 347 370 392 414 433 343 571 1005 326 349 ; 28.59418655 77.29478424

```

Converting json data file to PingER text file

There is a Python script `json_to_dict.py` see below:

```

import json

with open('pinger-7db5c-export.json') as json_file:
    data = json.load(json_file)
    #print(type(data))

    ping_data = data['Ping']
    required_format = ['monitor_Host_Name', 'monitor_Addr', 'remote_Name', 'remote_Addr', 'bytes', 'time',
        'xmt', 'rcv', 'min', 'avg', 'max', 'seq', 'rtt', 'latitude', 'longitude']

    file = open("PingER.txt", "w")
    #file.write("Monitor_Host_Name Monitor_Addr Remote_Name Remote_Addr Bytes Time Xmt Rcv Min Avg Max Seq Rtt
    Latitude Longitude\n\n")

    for keys in ping_data.keys():
        parameters = ping_data[keys]
        #print(parameters)
        #print('\n')

        new_dict={}

        for val in required_format:
            if val in parameters.keys():
                new_dict[val] = parameters[val]

        for value in new_dict:

            if(type(new_dict[value]) is list):
                for v in new_dict[value]:
                    v = str(v)
                    v = v + ' '
                    file.write(v)
            elif(value == 'latitude'):
                value = str(new_dict[value])
                value = ';' + value + ' '
                file.write(value)
            elif(value == 'time'):
                value = str(new_dict[value])
                value = value[1:len(value)-1]
                value = value + ' '
                file.write(value)
            elif(value == 'xmt' or value == 'rcv'):
                value = str(new_dict[value])
                value = value
                file.write(value)
            else:
                value = str(new_dict[value])
                value = value + ' '
                file.write(value)

        file.write('\n')

```

Firestore pricing

Firestore provides two plans to choose from. The one which we are currently on is called the Spark Plan. It is a free plan that provides generous data limits. As our application grows we would need to adopt the Blaze plan. The blaze plan is a “Pay as you go” plan as indicated on the website. It is a post billing service which means that you get billed for your usage at the end of the month.

Our application currently uses two services of Firestore - Authentication and Realtime database. For Authentication, according to the spark plan, we are allowed 10 000 phone verifications for numbers from the US, India and Canada and another 10 000 phone verifications from all other countries. If our application exceeds this amount we would have to pay \$0.01 per verification for all phone numbers from India, the US and Canada and \$0.06 per verification for numbers from all the other countries.

As for Firebase Realtime database, the spark plan allows us to store 1 GB of data and download 10 GB per month. For every additional GB of data stored we would have to pay \$5 whilst for every additional GB of data we download we would have to pay \$1. So to save space and possible charges we would need to limit the amount of data we collect to 1 GB. If we are on the verge of exceeding this limit we can delete a couple of entries from the database.

Details for various pricing services along with a blaze plan calculator is provided on Firebase's website:

<https://firebase.google.com/pricing#blaze-calculator>

Relevant papers

[An analysis of power consumption in a smartphone](#), A. Carrol and G. Heiser, 2009

Meetings

[Discussion on porting ePingER to Android phone](#), March 2016 with Sara Massood.

[20181114 Meeting with Amity](#)

[20190919 Meeting with Amity](#)

[20200115 Meeting with Amity](#)

Implementation

[ePingER on Android scope](#)

[PingER on Android phone two methods](#)

[ePinger on Android Native](#)

[ePinger on Android native, functional prototype](#)

Extension of the PingER project onto Mobile Devices using Android Applications, paper from Amity University Confluence conference January 10, 2019

Instructions from Aayush to load the application are at [ePingER Functional prototype](#)