ePingER on Android Native - Amity project

Idea from Topher White

So, the big idea at this point is to carefully describe the entire pinger workflow. I imagine it's not that complex, but I do not know how it works, so this is a very important step for figuring out what existing android code/libraries can be used to accelerate (and stabilize) the port process. It will also act as an informal behavior-driven-development definition of the pinger client application. If further development occurs on the existing app code-base (i.e. not the android version) then the workflow definition should be updated too, in order to guide the parallel development across the two platforms.

It could be as simple as this (which I'm totally making up, of course, cause I don't know how pinger currently works)

1. Initialize pinger app

2. Check for cached list of ping targets

   2a. If empty, send GET https request to https://something.pinger.com/ips.csv

   2b. Include client ID in http header

   2c. Include access token in http header

   2d. Receive CSV list of IPs

3. Iterate through fresh list of IPs.

   3a. Send ping

   3b. Cache latency [however it's done]

Discussion on porting PingER to Android

Implementation

Amity Android implementation

Using the Android/PingER to make a low cost monitoring device that can easily be used in developing countries is very noble goal; and very much attainable. We are more than halfway there to deploy apps on low cost monitoring devices in remote regions.

Progress towards that goal; July 2018, from Aayush Jain

1. The Android Team at Amity already has the android app in place. The app is capable of parsing the beacon list available at SLAC, and using it to ping all the beacons. The ping command's outputs are then stored in a txt file in a specific directory.What remains is to establish a way by which these generated txt files are sent to the server at Amity in a secure fashion. Les> Agreed
2. Additional requirements on top of standard PingER (e.g. proxy, new implementation of MA and why do not use perl and simply port existing MA pinger2.pl <http://pinger2.pl>);
    a. Our new suggested approach now involves replacing the perl scripts (executing on fixed line machines), with android apps on new portable MAs (ie, android phones). This would result in more diverse data to SLAC for analysis.
    b. The older approach as outlined in Rohan Sampson and Shiv Rajappa's paper involved the use of the pinger2.pl <http://pinger2.pl> perl script and porting it to android. However, the downside of that approach is that the script could only be run using an emulator on a rooted android phone (equivalent of jailbreak on iOS). Rooting a non-rooted phone is not only illegal in Google's eyes, but also requires a lot of technical setup, which is something not-so-foolproof to be used in developing countries. The rate of failure to even get the app running in such a scenario is very high, as compared to our new developed solution. Les> Agreed.
    c. Our solution doesn't require any form of rooting, or using any emulator to run a script. The code to ping a beacon, parse the output, and also to update the list has been made from scratch. Our new existing setup involves a user to only install the android application like any other application (like whatsapp) on any android device irrespective of whether it is rooted or not.
    d. Les> Excellent
    e. This approach thus deviates from using a singular perl script, but rather adopting a modular approach that can be scaled accordingly. To achieve this vision, we suggest sunsetting the existing pinger2.pl <http://pinger2.pl> services from the servers at Amity. Rather, the server should only act as a proxy/medium/middleman between the Android MAs and SLAC.
    f. Les> Personally I would retain the existing wired pinger2.pl MA so you can compare etc with the new Android/wireless version.
3. how results may be affected by wireless access versus wired PingER MA's;
   The ping-command outputs from a wired MA will always be more predictable over a period of time. Fixed lines usually aren't affected by factors such as weather, geographical location, etc. A portable wireless MA allows a user to gather data from different locations, and over multiple transmission mediums (cellular, wifi, etc.). Such diverse data results in high quality datasets for analysis by SLAC. The app can also be tweaked in the future to generate a few more data points (like GPS location, and connection type) that currently arent being used by PingER. This can aid in a deeper analysis.
4. Benefits such as low power, space;
   Earlier the servers at Amity were the MAs, and did not produce much amount of data, and neither was it so diverse.
   Now, in our proposed model, the servers at Amity act only as a proxy. The Android Apps are the new MAs.
   This new model now provides multiple advantages advantages, such as:
    a. more quantity data accumulated
    b. more diverse data accumulated
    c. failure of one MA does not affect the functioning of other MAs
        i. Les> that is also true with the wired MAs
    d. Android devices are hugely popular. Deployment shouldn't be a problem.

e. The application can run in the background without affecting the user's activities
    i. Les> Also true of the wired MAs, they do not have to be dedicated
f. Mobile devices consume considerably less power than PCs
g. Mobile devices are portable, and can be taken to more places for data collection.The Android Application takes up only a few MBs to install, as compared to a full server in Linux and its dependencies.
    i. Les> Thie second sentence may need some clarification. pinger2.pl takes only 64KB, it needs a perl interpreter, temporary storage space and an OS (Linux). I assume the Android provides an API, temporary storage space, and the Android OS. Thus at the top level they have a lot of similarities.

5. use cases
   If deployed correctly, using the Android Device as an MA gives rise to whole multitude of possibilities. Only a single MA in an isolated/sparsely populated region can provide sufficient data for IEPM. Places with bad weather, or ones struck with natural calamities, or in remote regions such as rainforests or deserts can bring ground level data for analysis. Since the app stores data locally as well, thus in the absence of internet connection, the app will cache it until a stable internet connection is found. A wired MA cannot provide such dynamic flexibility to the PingER /IEPM project.
   Les> well made points

I hope this clears your reservations.

All we need your guidance now is on how to connect the app to the server.
Les>Please see https://confluence.slac.stanford.edu/display/IEPM/Proxy+support+for+PingER for a mechanism for using a proxy anonymous inbound FTP server to get the data to SLAC. Something similar could also be set up at Amity.

And also, on how can existing services on the PingER Amity server be shutdown/transferred to the app safely.

Les> As I mentioned above I recommend retaining the existing wired server for comparisons etc. If you have to shut it down for some reason let me know and I will stop gathering data from the Amity wired MA. Personally I think shutting it down currently is a bad idea. SLAC can gather the data from both the existing wired Amity MA and future wireless MAs.