**Cool Telephony App**

Stage wise development

STAGES:

Stage 1: Simple message exchanges in unsteady environment

1. Basic parameterizations of 2G networks and minimum requirements for voice

**(2G** uses channel **bandwidth** of 200 kHz for voice transmission, and 3G uses 1.25 MHz channel).

Research - 10Kbps bandwidth is enough for voice transmission.

1. Sending simple messages from android app to server: Details
   * 1. First create a simple meser every stage of 10 Kb.
     2. Then, create unique replicates of this message by attaching unique identifiers (like 1001, 1002, 1003, …, 1100,.....) to it (in the form of headers), at the beginning of the message. These will be created on the go - that is these will not be stored.
     3. There will be a cron job running in the background of the Smartphone app, which will do a fake call of “sending these textual messages” at the beginning of every hour for 20 minutes, from 6 am to 8 pm, every.
     4. Essentially, sending these messages will have the following form: A new Unique (next) 10 KB (uniquely identified) package, will be sent to the server every second.
     5. The order in which the packages are received on the server side, on single transmissions, along with the unique IDs of the packages (created above), are sent. The order and the timings of the messages received are stored in the backend, on the server.
     6. At the start of the package transmissions (discussed above, which happens at the start of each hour given above), and after every 2 minutes, there is a complete calibration of the connectivity of the smartphone (this includes calibration of the bandwidth, calibration of packet(s) drop, and so on). At the end of these sessions, all these collected parameters are gathered together, and saved in the backend via another appropriate API call.
2. Sending simple messages from server to android app
3. Exchanging simple messages between two apps
4. Testing and collecting data on 2G networks

**VI.** It’s just doing all work using work manager on the backend. There is no message sending that is driven by front end - in the first phase.

There is a dummy message, of size say 5K - you can load it from somewhere/anywhere. Replicated of this same message, are attached with unique IDs (like increment from some base value), to create new unique messages.

These messages are transmitted to the server at repeated intervals, as specified in the document. The same related details, and other networks parameterisations (evaluates in front end) are recorded/saved on the server using appropriate APIs, from time to time.

**VII.** Varun WebRTC is an excellent recommendation..

You can check SlashRTC, their interface is excellent. Mumbai based guys. As per their sales pitch, Vedantu uses them.

It seems they are making VoIP calls on PSTN (mobile).. is that legal in India

If you want to go international, you can also check out Maqsam. As far as twilio and plivo are concerned, you may need to create a dashboard for the purposes you listed, same goes with Maqsam. Also with plivo and twilio, you would need to integrate a virtual phone.

Haven't tried WebRTC folks. I thought webRTC is a concept/framework for online communications. Didn't know that there is a service provider with that name.

Basically, the App will be installed on the smart-phone of the users.

Stage 3: Integration of voice

Stage 4: Integration for video call

Stage 5: Integration of screen sharing

<http://www.cse.iitd.ernet.in/~aseth/rural_india_2g_3g.pdf>

<https://www.researchgate.net/publication/262355591_2G3G_network_measurements_in_rural_areas_of_India>

<https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2016/07/Unlocking-Rural-Coverage-enablers-for-commercially-sustainable-mobile-network-expansion_English.pdf>

<https://www.opensignal.com/reports/2019/04/india/mobile-network-experience>

<https://www.lightreading.com/jios-dream-of-2g-free-india-is-nightmare-for-telcos/d/d-id/762916>

<http://nuranwireless.com/gsm-ran-literan-2g/>

<https://trai.gov.in/sites/default/files/White_Paper_22022019.pdf>

<https://www.vodafoneidea.com/content/dam/vodafone-microsite/docs/pdf/investor-/investor-presentation/fy-16-17/Investor%20Presentation%20Q2FY17.pdf>

<https://www.ee.iitb.ac.in/~karandi/talks/Affordable_IEEE%20ANTS_Chennai_Dec2013.pdf>

<https://www.vodafoneidea.com/content/dam/vodafone-microsite/docs/pdf/investor-/investor-presentation/fy-16-17/Investor%20Presentation%20Q2FY17.pdf>

<https://www.financialexpress.com/industry/your-phone-is-dead-in-nearly-28000-villages-of-india-these-reasons-to-blame-for-no-mobile-network/1875037/>

<https://www.gsmarena.com/network-bands.php3?sCountry=INDIA>

<https://dl.acm.org/doi/pdf/10.1145/2442882.2442938>

<https://dot.gov.in/sites/default/files/statistical%20Bulletin-2018.pdf>

<https://www.nperf.com/en/map/IN/-/-./signal/>

<https://www.moneylife.in/article/mobile-subscribers-struggle-poor-networks-low-data-speeds-and-services-no-grievance-redress/59257.html>

<https://www.sciencedirect.com/topics/engineering/mobile-network-operator>

<https://www.developingtelecoms.com/telecom-technology/wireless-networks/6693-vodafone-india-and-bsnl-partner-for-2g-network-sharing.html>

<https://www.ias.ac.in/article/fulltext/reso/008/11/0037-0051>

<https://www.avoxi.com/blog/how-much-bandwidth-is-needed-for-voip/>

<https://www.phone.com/much-bandwidth-need-voip/>

<https://www.gl.com/voice-codecs.html>

<https://personal.utdallas.edu/~torlak/courses/ee4367/lectures/lecture2.pdf>

<https://www.nextiva.com/blog/voip-data-usage.html>

<https://personal.utdallas.edu/~torlak/courses/ee4367/lectures/lecture2.pdf>

<https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html>

<http://lehman.edu/faculty/hoffmann/itc/techteach/audio/audiotechinfo.html>

<https://vcc.live/blog/bandwidth-voip-technology-require/>

=======IP Sator=======================

abhinav@ipsator.com

[gaurav@ipsator.com](mailto:gaurav@ipsator.com)

[vaibhavs@ipsator.com](mailto:vaibhavs@ipsator.com)

============================ APIs requested=======================

===========================Architecture=============================