APPENDIX

A. CODE AND DATA

To make our results reproducible, we have created a public repository of the code and data used in the project. As described in the paper, the experimental setups for this work have been non-trivial. On the client side, we needed a laptop, tethered on Wi-Fi to a mobile phone, that acted as a Wi-Fi hotspot. This mobile had a SIM card with Free Basics support for that country and also had credit balance to conduct the non Free Basics normal cellular connection experiments. We had these client setups in LUMS, Pakistan and UCT, South Africa. In Pakistan, we conducted experiments using SIM cards from two cellular service providers: Zong and Telenor.

In addition to the client side measurement setup, we implemented and controlled two web servers for Bugle News and Learn Basics, hosted at MPI-SWS in Germany. We registered these services to be part of Free Basics service list, going through Facebook's online application and approval process. We also moved the physical hosting of the Learn Basics server across Virginia, Sao Paolo, Mumbai, Sydney and Tokyo, using Amazon EC2 instances, for deconstructing the proxy architecture and network path inflation details, as described in Section 4.

If someone wants to conduct the experiments again, please get in touch with us. Both our web services are live on Free Basics, continuously accumulating visits from the ever growing number of Free Basics countries. Our collaborators in Pakistan and South Africa, who helped us conduct the client side measurements, are our co-authors in this paper. Thus they can help to conduct similar client side measurements in future, in these two countries.

In the public repository [5], we make available all the data collected in our experiments. We also include the code used to run th experiments, and the analysis code used in processing the data and generating the results and the graphs. Below we give a description of the repository, in connection to the different sections in the paper.

A.1 Network Characterization

The code and data for the discussions in Section 4 are in two folders: *4_ec2_experiments* and *5_throttling_pakistan*.

Path latencies and inflation. The *4_ec2_experiments* folder has a python script frb_crawl.py (with comments describing what it does). This script is run on the client side laptop, to fetch content from our Learn Basics server. The client side laptop further runs mitmproxy, to record the requests and the responses and the IP addresses with which it communicates. The server side runs tcpdump to store all incoming requests in pcap files, from which the requests coming from our clients are filtered using the string "Amreesh" in the user-agent field.

The *4_ec2_experiments* folder has two subfolders, one for Pakistan and the other for South Africa. Within each subfolder are three subfolders *clientside/* (contains the client side logs from mitmproxy), *serverside/* (contains the severside pcaps using tcpdump) and *server-client-matched/* (where our client outgoing requests are matched with server side incoming requests). Both the *clientside/* and *serverside/* folders have python scripts to process the mitmproxy and tcpdump outputs and generate text files. The *server-client-matched/* folder has a script times.sh, which computes the ping latency from mobile to C-Proxy and that between S-Proxy and our web server. The cdf.pl scipt computes the CDF of these latencies and cdf.gnu script plots the graphs (Fig. 2 and Fig. 3).

Throttling Policies. *5_throttling_pakistan* contains similar python crawler scripts frb_crawl.py and nfrb_crawl.py, to fetch a large HTML file from the Learn Basics server, over Free Basics connection and normal paid cellular connection respectively. The script trace.py computes throughputs from pcap files generated using tcpdump at the client and the server sides. The Telenor and Zong folders contain the CDF of the throughputs, which the plot.gnu script plots into graphs in Fig. 4 and Fig. 5. We do not upload all pcap files because of their large sizes. In case someone needs the raw pcap files for regenerating our throughput numbers or for conducting some other data analysis, please contact any of the authors for an immediate data exchange.

A.2 Device Characterization

The code and data for the discussions in Section 5 are in three folders: *1_pcaps_to_useragents*, *2_useragents_to_devices* and *3_devices_to_capabilities*.

The script country_useragents.py in *1_pcaps_to_useragents* takes the raw pcap files (example file trace_20160802.pcap) as input, to extract the country and the user agents and generate learnbasics.txt and newsbugle.txt as output.

The user agents are taken in *2_useragents_to_devices*, to produce the output sorted-freq-mobile-devices.txt, after mapping the user agents into mobile devices and sorting and enumerating unique mobile devices. The auto_device_detect.py script inside each subfolder for all and Pakistan specific user agents, processes the user-agents.txt files and maps the user agents to mobile devices using the Scientia Mobile website[6].

In *3_devices_to_capabilities* folder, list contains the names of 34 mobile device vendors, predominant among the mapped devices. The *links/* subfolder contains 34 text files, one for each vendor. These vendorlinks.txt files were created using https://magic.import.io/. We manually gave import.io any website (e.g. http://www.imei.info/phonedatabase/2-phones-alcatel/). import.io automatically crawled all pages under that page to get all the links, which we saved in the text file.

The script fetcher.py takes each link in vendorlinks.txt and downloads the webpage in a temporary folder. fetcher.sh calls fetcher.py for each vendor. The script filler.py takes each webpage (contains details for a particular device model) and extracts specified capability features. Sample outputs are in the *device_capability/* subfolder. Filler.sh calls filler.py for each vendor. The graphs in Section 5 are drawn using the capabilities information of the devices thus extracted.

[5]https://bitbucket.org/rijurekha/freebasics_ccr

[6]http://tools.scientiamobile.com/