Lecture 11 Cellular Networks

Networks and Mobile Systems

April 26, 2018

In this lecture we will talk about Cellular Networks, LTE technology in particular. We will briefly look at the history of Cellular Networks followed by some basic terms and concepts that will appear in the lecture as we proceed. Following that we will look at two research papers, Sprout[1] and LTEye[2], that study cellular networks from two different view points. Sprout looks at a technique to best utilize the link capacity of cellular network for best throughput over delay by working at the transport layer. LTEye on the other hand tries to better understand why LTE networks work the way they do by capturing information at the physical layer.

1 Cellular Networks

Cellular Networks are the underlying technology behind almost all mobile devices today. They allow mobile devices to remain connected to the internet while on the go. In a basic cellular network, a mobile device is connected to a radio tower wirelessly and the radio tower indeed is connected to other towers or a physical network that evetually connects the mobile device to the internet. Typically service providers (like AT&T, Verizon and such) deploy a few hundred of such radio towers across any city to provide cellular connectivity through out the city. Over the years these networks have evolved significantly. Table 1 compares the different cellular technologies over the years in terms of their capacity and functionality.

Table 1: Generations of mobile technology

Generation	Year	Features	Capacity	Applications
1G	1980-1990	Voice Only	-NA- (Analog Signal)	Only phone calls
2G	1991-2000	Voice + Data	<100 Kbps	SMS, RSS Feeds
3G	2000-2010	Voice + Data	<5 Mbps	Basic Internet, EMails, Rich text and images
4G	2010 - 2018	Voice + Data	100Mbps	Video browsing, Conference calls, Multiplayer gaming
5G	2018 (Demo)	Voice + Data	1Gbps	Better support for IoT devices

Each of these generations of mobile technologies used different techniques to improve and scale with the growing demands like GSM, CDMA and LTE.

Following are some commonly used terms and definitions in context of cellular networks.

1. **Cell:** In a cellular network, the geographical area that needs to be serviced by the network is divided into small geographical areas called cells. Based on the size of each cell they are categorized as Microcell, Femtocells, etc.

2. **Base station:** A base station is a wireless transceiver or a wireless router that is located at the center of a cell (on a tower) and connects mobile devices to a physical or a wireless network.

3. **Modulation:** Simply put, modulation is the process of piggybacking data on a carrier signal which is transmitted over a telecommunication medium.

4. **Time Division Multiple Access (TDMA):** One of the improvements that happen across different generations of mobile technology is being able to scale with increasing users. TDMA was technology that allowed multiple users to use the same frequency at the same time. In TDMA each user is allocated a time slot when it can send data over a certain frequency. This way no two users send data over the same frequency at the same time hence leading to better use of the frequency spectrum.

5. **Code Division Multiple Access (CDMA):** CDMA is an improvement over TDMA, wherein multiple users can share the same frequency spectrum at the same time. Each transmitter is assigned a code to avoid unwanted interference among different transmitters.

6. **Global System for Mobile Communication (GSM):** This was the standard used by 2G networks which used a modified version of TDMA. GSM is a circuit switched system

7. **Long Term Evolution (LTE):** LTE is a wireless communications standard used based over GSM. LTE as such does not meet the criteria for 4G, but is marketed as a 4G technology.

2 Long Term Evolution (LTE)

Major telecommunications companies like AT&T, Verizon and such, provide LTE (4G) services to its users and with growing demand these companies are looking to deploy new cells (base stations). However there is not enough insight into how these networks perform in terms of quality of service, utilization of assigned frequency spectrum and more. This leads to inefficient deployments of new cells (base stations), policy makers are unable to propose policies that make better use of the spectrum and researchers are unable to suggest improvements to existing networks and develop new technologies.

To address such issues Kumar et. al. in [2] proposed an open platform, LTEye, that analyses LTE performance at temporal and spatial granularity providing better insights on the performance of LTE. They build a platform by snooping in on the LTE traffic with low cost equipment and without the involvement of the service providers (like AT&T and Verizon).

A major design goal of LTE networks was to provide good quality real-time video browsing and calling capability. However, the existing network transport protocols and congestion control

schemes were not well suited for cellular networks. Cellular networks experience high variation in the link capacity which makes the existing algorithms, protocols inefficient. Wienstein et. al. in [1] propose a new transport protocol that better uses the cellular networks for interactive applications like Skype, Hangouts and Facetime. Their results show significant improvement over existing transport protocols.

Let us look at each of these papers in more detail in the next two sections.

3 LTEye: LTE Radio Analytics Made Easy and Accessible

As mentioned in the last section, LTEye developed an open platform to monitor and analyze LTE radio performance. The main points that this paper addresses are

- Low cost equipment to snoop on LTE radio. Did not require any implementation of the LTE protocol or setup a separate network for measurement
- No dependence on LTE operators or service providers for measuring performance. It makes use of the meta-data from LTE control channel.
- One of the first techniques to measure performance at the physical layer. All other attempts to measure LTE performance were at higher layers of the networking stack

3.1 Some LTE concepts

Before we look at the details of the paper, following are some LTE terms/concepts used in the paper

1. **Resource Allocation:** Each base station in the LTE network, divides the radio resources into multiple frames. Each frame is then divided into multiple subframes along both time a frequency (Figure 1 in [2]). Each of these subframes are called resource elements.
2. **Channels:** LTE networks need to send both control and data information to users. From the grid of resource elements available, base station assigns a group of these elements to each of the different types of information, data and control, and calls it data channels and control channels respectively. LTEye makes use of only the control channel information as it is unencrypted.
3. **LTE protocol fields:**
   (a) **C-RNTI**: Cell Radio Network Temporary Identifier is a unique ID assigned to a user connected to a base station. This ID is assigned for a particular session and for a particular cell(base station) that is serving the user. Base station could assign a new ID to the same user for a new session.
   (b) **Checksum:** This the XOR of C-RNTI and checksum of control message. LTEye uses this field to retrieve C-RNTI (details on this later).

3.2 Setup and Experiment

There are two main components in the LTEye setup: (1) LTE Logger, and (2) Analyzer. LTE logger's work is to sniff the downlink control channel messages and populate the transmission records in the database used by the Analyzer. To sniff the LTE traffic, it uses a passive 2 antenna MIMO receiver. As LTE uses OFDM for sending packets to the user, logger needs to perform demodulation, de-scrambling and other operations before it is able to extract the exact control message that was sent by the base station. Once the logger has extracted the exact control message sent, it retrieves the C-RNTI from the messages and populates the LTEyeDB. Each transmission record in LTEyeDB is tagged with a timestamp. Some other fields in transmission record are: C-RNTI, a bit for Uplink or Downlink, Signal to Noise Ration, Bit-Map of resource block etc. (Refer to Table 2 in [2] for complete list). Transmission records collected over time are then used to analyze the networks performance per user or the network as a whole.

The transmission records collected in the database as such have no information at the granularity of a user. It just holds information about different C-RNTI. As we saw in Section 3.1 a single user could be assigned different C-RNTI across different session. To track the same user across different sessions (new C-RNTIs), for each C-RNTI within a session RF fingerprint is computed. RF fingerprint consists of the user's location, its signal to noise ration (SNR) and the multipath characteristics (discussed later). After assigning each C-RNTI its RF fingerprint, they match the C-RNTI across sessions using the Hungarian Matching problem. Weights of the edge between pair of C-RNTIs is the similarity of RF fingerprints. Nodes of the graph are C-RNTI values and two extra states NEW and EXIT. These are to track users who just joined the network and the users who are no longer part of the network. This enables LTEye to track a particular user across different sessions.

Another challenge faced by LTEye is to actually extract the C-RNTIs from the control messages. In the protocol specification, the C-RNTIs are sent to the users initially when the connection setup happens using protocols at higher level of the networking stack. LTEye may start sniffing the traffic after the initial setup has happened. To still be able to identify the C-RNTIs having missed the initial setup they use the last 16 bit of the packet, which is the checksum. As we saw in Section 3.1, the checksum is the XOR of the C-RNTI and the checksum of the control message, after de-modulation and decoding the packet, they themselves calculate what they think should be the checksum of the control message. After doing that they XOR it with the last 16 bits of the packet. The result of the XOR is the C-RNTI. Now to actually verify if the control message they received was not corrupt in the first place, they perform modulation and coding on the message and compare it to the original message that they received. The tolerate an error of upto 3 bits, beyond which they discard the packet.

Many measurement papers perform such clever tricks to extract the desired parameters or information from packets, network since they are measuring the network as an outsider and they have little to no control over the network.

So far, LTEye is able to extract C-RNTIs, match C-RNTIs across different sessions and track individual users and add all the collected data (transmission records) to the LTEyeDB. Let us now look at some other details in their setup that particularly show how they are able to extract modulation and coding schemes, and finding location of a user.

LTE standard itself allows a wide range of modulation and coding schemes and the particular scheme used by a base station is know to the user through the encrypted data channels. Since LTEye uses only the control channel for its measurement, it does not have access to the scheme sent by the base station. One way to address this problem is to do a brute force search across the space of all possible schemes supported by LTE and identify the scheme for a given packet. This turns out to be a very expensive search to perform in terms of computation. However, it turns out LTE service providers like AT&T and Verizon use a small subset of the available space of schemes. Performing search over a very limited subset of schemes is not very expensive (AT&T and Verizon use around three different schemes only). LTEye can learn the subset of schemes used by a particular service provider in the first few control messages and use that information for subsequent packets.

To find the location of the user, LTEye follows a three step approach. (1) Extract the multipath profile of an LTE signal, (2) Identify the direct path (if any) from the profile and (3) Localize the source of the LTE signal using intersection of multiple direct paths (identified using other receivers). As mentioned earlier, LTEye uses 2 antenna MIMO receiver of which one antenna is static and other antenna moves in circular motion using a rotating arm attached to the sniffing device. They use this two antenna setup to nullify the error that gets accumulated over time due to the Carrier Frequency Offset (CFO) details of which can be found in Section 6 of [2]. Using the signals received by the two antenna, they apply SAR equations to identify the multipath profile of the LTE source (Mathematics behind the exact calculation is omitted. Refer Section 6.1 in [2] for details). Once the multipath profile has been computed, LTEye tries to pick the signal that it thinks is a direct path to the LTE source (to identify direct path they just pick the one with least delay) Using multiple direct paths from other LTEye sniffers, it finds the points of intersection of direct paths. That is how they are able to locate LTE sources.

3.3 Results

LYEye paper measures different metrics both temporally and spatially. Following are their findings.

- **Active Users:** They measured the number of active users per service provider at different times of the day. They observed a peak during afternoon hours. Their hypothesis is that people come out of buildings for lunch.
- **Network Utilization:** This was probably the most important result. They observed that the uplink channels are significantly under used as compared to downlink channels. To give some perspective, Verizon's downlink utilization is around 58.2% and uplink utilization is around 2.6%. This shows that there is a significant skew in the utilization of different channels in the network. Such measurements can help policy makers to allocate frequency spectrum accordingly.
- **Link Quality:** They also report the link quality by measuring the average number of bits transmitted per resource element. They observed that link quality of downlinks was better than uplinks.

- **Control Overhead:** LTEye measurements show that operators provision more control channels than are needed. Once allocated the control channels are not dynamically reduced or increased based on the usage. For example, Verizon has a 21% control overhead (Figure 8 (b) in [2]).
- **Inter-cell reference:** LTEye also showed that multiple base-stations interfere with each other leading to destructive interference leading to poor network performance. Such measurements can help service providers to better place their base stations to avoid inter-cell interference.

4 Sprout: Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks

As mentioned in Section 2, Sprout is a new transport protocol aimed to improve interactive traffic like video conferencing or browsing over cellular networks. Applications like Skype, Hangouts or Facetime do not perform well over cellular networks that use traditional transport protocols and congestion control schemes because of high variation in the the link capacity in cellular networks. Sprout addresses the issue of variable link capacity by modeling the network in terms of how the link capacity varies over time. The results show considerable improvement in throughput and lower delays.

The main idea behind Sprout is to make use of the fact that cellular networks maintain a per-user queue, implying that the delay observed by a particular user is self-inflicting. This happens because the base station schedules data transmission considering per-user fairness and channel quality (similar in spirit to WFQ). It is in the best interest of a user to control the rate at which is sends packet into the network, to avoid delays due to buffer build up. In addition to this, Sprout uses the arrival times of packets at the user to infer future rates as a signal of congestion or link capacity. This is different from existing congestion control algorithms that are reactive in nature. They use packet drops, round trip delays as signal of congestion or link quality.

4.1 Sprout's Model of the Network

As per the experiments performed by the authors, they observed that cellular networks can be approximated as a Poisson process. Sprout's model of the network needs to estimate the link capacity at the present time and also in the future. This is done to predict the number of packets that the user should send to minimize delay due to queue build up.

A Poisson process is used to model packet deliveries. The underlying rate in a Poisson process λ is estimated by using the arrival rate at the user's end. The network is modeled as doubly-stochastic process where λ keeps changing as per Brownian motion with a noise power of σ ($packets\ per\ second\ per\ \sqrt{second}$). The λ gets updated is, if say at t=0 $\lambda = 137$, then at t = 1, λ will follow a normal distribution with mean equal to 137 and standard deviation equal to σ. A larger standard deviation means that the model is less confident about predicting the link rate. To have a model that is as close an approximation to a real cellular network, authors also

model outages. For $\lambda = 0$, it is assumed that the link is experiencing an outage. They also have an outage escape rate that resembles sticky outages (e.g., going through a tunnel).

The evolution of the network happens in three steps,

1. Apply Brownian Motion to all 255 λ values (Sprout maintains probability distribution on λ in 256 floating point values. For $\lambda = 0$, the outage escape rate is taken into account while applying Brownian Motion.

2. Update the probabilities by multiplying it by the likelihood that a certain Poisson Distribution with a certain rate would have produced the observed rate.

3. Normalize the 256 probabilities.

The last part in the Sprout algorithm is to predict the future link rate. For that Sprout picks the 5th percentile of how many bytes will arrive at its receiver during the following 8 ticks (160 ms) by evolving the model 8 times. Sprout is cautious in estimating the future link rate. The sender and receiver exchange information about the forecast and the total number of bytes that the receiver has received so far. This information is then used by the sender to estimate the queue length and make modifications to the window size. Sender ensures that each packet it sends has a probability of atleast 95% that is will clear the queue in the next 100ms. This is exactly how the user is able to control the amount of packets that it sends into the network and thus avoiding self-inflicting delays.

4.2 Experiment and Results

For testing the protocol, Sprout uses a trace-driven simulation. To collect traces, they built an application that they call Saturator, which basically tries its best to keep the queues as filled as possible. They keep the delay high and not too high by keeping the RTT between 750ms and 3000ms. They also use a second phone as a feedback device because if the delay in feedback is too high control the network becomes hard. Once the traces were collected, they used a network emulator, Cellsim, to replay the traces. Cellsim runs on a PC and is connected to two ethernet interfaces. It takes in packets from one ethernet port and adds them to a queue and then later releases them in to the other port using the trace collected earlier. This is how Cellsim is able to emulate a cellular network. Cellsim also takes in to account packet losses to study resilience towards packet drops.

The results show that Sprout outperforms the existing delay based congestion control schemes like TCP Vegas, Cubic and Compound TCP (Figure 7 in [1]). A simpler version of Sprout called Sprout-EWMA also performed as good as Sprout. Sprout-EWMA uses an exponentially weighted moving average of throughput instead of the forecasting method used by Sprout. The reason that Sprout-EWMA works well is because only certain amount of caution is needed on forecasting the link rate. The fact that cellular networks have per user queues already simplifies the problem of congestion control.

5 References

[1] Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks, in the proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2013), Lombard, Ill., April 2013.

[2] LTE Radio Analytics Made Easy and Accessible, Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li, ACM SIGCOMM 2014,Chicago IL