

# Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications

Nishad Gothoskar  
ngothosk@andrew.cmu.edu

---

## 1: Summary

---

The main concept discussed and that was attempted to be mitigated in this paper is *tail energy*, the energy consumed by radio that sits in its high power state even after a process/request has been completed. This is a dilemma because it is inefficient for a system to shut off its radios immediately after a process is finished because of concepts of temporal locality and the fact that its likely it will be used again. But also, you don't know if it'll be used again soon and how soon so you unnecessarily drain energy.

But of course, the real issue is that the scheduling problem is NP-hard so we can't get our best case schedule. The best online algorithm can at most be a 1.62 approximation and TailEnd is a 2 approximation algorithm.

The research conducted was very thorough in the variation of parameters: location, time-of-day, mobility, devices. They were also very thorough with 3G vs. GSM vs. WiFi as well as Upload and Download.

Their main strategy was to bunch up the accesses and accumulate them before sending them through so as not to have many intermittent accesses as that would maximize the energy overhead of keeping the radios on. The papers abstracts cites the example that a few hundred bytes transferred intermittently on 3G can consume more energy than transferring a megabyte in one shot.

So, prefetching data may be in the best interest if you could know with high probability that that information will be used. The downside is you can't be sure that information will be needed.

---

## 2: Strengths

---

- The research conducted takes a lot of "not novel" concepts and incorporates them to make a management system that delivers significant results.
- I enjoyed that they discussed the actual algorithm and the mathematics behind it. That is something a lot of papers don't do, they rather just give a brief description of how the algorithm works.
- They actually showed significant improvements and energy savings and significant variation as a result of modified parameters (day vs nights, upload vs download)

---

## 3: Weaknesses

---

- I feel like they underestimated how much delay matters in a user experience
- With mail, they already do something like accumulating before sending

- But with mostly everything else users don't want to wait - Web browsing, uploading pictures, streaming from netflix.

---

#### 4: Future Directions

---

- A clear direction for this work is for us on our PC to be able to specify delay-tolerance of each of our applications. Some of things we do on our PC we can wait for them to happen. I believe that Mobile devices are a different story when it comes to batching.
- Another direction is to give recommendations on better times of doing a certain "job" like do this at night if you don't need it now. Or can you wait till you accumulate a few more downloads before starting these.