Noname manuscript No.
(will be inserted by the editor)

Power Consumption Analysis of Video Streaming in 4G LTE Networks

Jingyu Zhang[1,4] · **Zhi-Jie Wang**[2,1,3] · **Song Guo**[3] · **Dingyu Yang**[5,1] · **Gan Fang**[4] · **Chunyi Peng**[4] · **Minyi Guo**[1]

Received: date / Accepted: date

Abstract Video streaming, one of the most popular technologies for online video playback, has already been applied to 4G LTE networks. Previous work has been devoted to understanding the power consumption in general 4G LTE networks, while it is still unclear how the online video streaming makes impact on the power performance of mobile devices. Inspired by this, this paper investigates the relationship between the mobile device's power performance characteristics and the behaviours of video streaming in 4G LTE networks. There are many natural issues/questions that are clearly interesting and important, while it is non-trivial to answer these issues/questions exactly (e.g., where is the energy saving room? how much is it?). To address a series of issues like the above, we formulate our energy models together with an algorithm that can assist our analysis. Particularly, we design a systematic platform, and conduct a comprehensive and also deep analysis on the power consumption of video streaming in 4G LTE networks. Our experiments reveal us a series of valuable findings — the saving room in the network part is large (from 41.86% to 69.62%), the number of RRC tails and the transmission pattern could be promising for optimizing the power consumption, for example.

Keywords Video streaming · 4G LTE networks · Power consumption

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China
[3] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[4] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
[5] School of Electronics and Information, Shanghai Dian Ji University, China

1 Introduction

In recent years, the mobile industry has been growing rapidly, and there seems to be no any "stop sign". The mobile industry generated 3.8% of global gross domestic product (GDP), it amounts to over 3 trillion U.S. dollars in 2014 [1]. Meanwhile, we are witnessing a rapid technology migration — the increased adoption of higher speed mobile broadband networks (e.g., 4G LTE) and various mobile devices (e.g., smart phones). In 2015 the global monthly mobile traffic is near to 3,500 PetaBytes, while *video data* is around 45% of mobile data traffic [2]. It is reported that *mobile video* grows by around 55% annually through to 2020 [2]. These evidences show us that, in the higher speed mobile broadband networks such as 4G LTE, mobile video will be dominant in mobile data traffic in the near future.

Video streaming [3], one of the most popular technologies for online video playback, has already been applied to 4G LTE networks [4]. In the video streaming implementation, the video source files are to be split into many chunks, and encoded to the desired delivery format. This allows clients to download, in a linear fashion, a series of small video segments from an HTTP video server to the local storage, and play back them in sequence [5].

Motivation. While we have witnessed the advantages of video streaming and the rapid development of mobile networks, the advancement of the battery technology, however, is relatively slow [6]. Often the battery drains fast when mobile clients use online video services [7], the battery capacity is naturally concerned by most of mobile users who like to enjoy online videos via 4G LTE networks. Previous work has been devoted to understanding the power consumption in general 4G LTE

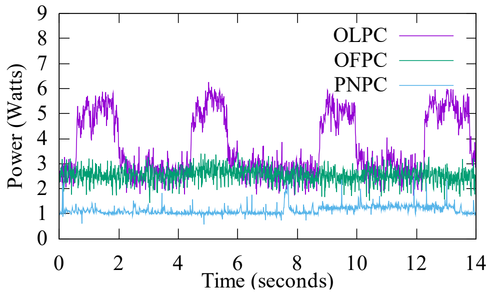


Fig. 1 The distribution of power consumption. The PNPC, OFPC and OLPC denote the power consumption when *no* video is played back on our mobile phone, when the mobile phone plays back the video *offline*, and when the mobile phone plays back the video *online* in 4G LTE networks, respectively.

networks (see, e.g., [8–10]), while it is still unclear how the online video streaming makes impact on the power performance of mobile devices.

Inspired by this, we investigate the relationship between the mobile device's power performance characteristics and the behaviours of video streaming in 4G LTE networks. Our main goal is to make a deep understanding on their relationship. Figure 1 depicts the distribution of power consumption (obtained from our experiments), from which we can get a preliminary intuition on the power consumption of video streaming. Nevertheless, there are still many natural issues/questions that are clearly interesting and important, whereas it is non-trivial to answer these issues/questions exactly (e.g., where is the energy saving room? how much is it? how the online video streaming makes impact on the power consumption of mobile devices?). To address a series of issues like the ones mentioned above, we conduct a comprehensive analysis on the power consumption of video streaming in 4G LTE networks.

Contribution and novelty. To summarize, our major contributions are as follows.

- We point out a series of interesting and important issues related to mobile devices' power consumption and online video streaming in 4G LTE networks. To the best of our knowledge, this is the first paper analysing the relationship between the real world mobile phone power consumption and the real world online video streaming data transmission in 4G LTE networks.
- To understand their relationship from a theoretical viewpoint, we formulate the energy models, which characterize various scenarios and provide us some insights. The models could be of independent interest, and could be used to understand other features related to the energy consumption and video streaming.
- We design a systematic platform that allows us to conduct a deep study on the power performance characteristics of video streaming in 4G LTE networks.
- Based on the energy models, we develop an algorithm that is used to assist us to roughly understand the energy saving room and some other useful information (e.g., which objects or phases could have promising saving room).
- We conduct a comprehensive and also deep analysis based on our platform. Our experiments reveal us a series of valuable findings — the saving room in the network part is large (from 41.86% to 69.62%), the number of *RRC tails*[1] and the transmission pattern could be promising for optimizing the power consumption of video streaming in 4G LTE networks[2], for example.

Roadmap. In the subsequent section, we introduce the preliminaries related to video streaming technology and 4G LTE network characteristics. Section 3 presents our energy models. We describe our systematic platform and an assistant measurement algorithm in Section 4. In Section 5, we answer a series of issues/questions based on the real world experimental results. Section 6 reviews the related work, and Section 7 concludes the paper with the interesting research topic.

2 Preliminary

For ease of understanding the rest of the paper, this section covers the necessary background related to the operational process of video streaming, the 4G LTE radio resource control state transitions, and the timing scheme of video streaming.

2.1 Operational Process of Video Streaming

Figure 2 illustrates a typical network architecture of video streaming [3, 4]. Firstly, the captured video source is to be split into a series of small segments (a.k.a., chunks), which are to be encoded in the *media preparation* phase. Then, the *encoded chunks* along with media presentation description (MPD) file are to be hosted on one or several HTTP web servers. (Note that the MPD file is used to describe the *information of video chunks* such as accessible segments and corresponding timing information.) As such, clients are allowed to exploit *end*

[1] Simply speaking, the RRC tail refers to a network activity state that appears in some time interval. Its meaning will be clear after we introduce the background in Section 2.

[2] Note that, reducing the number of RRC tails does not mean reducing the duration of a single RRC tail.

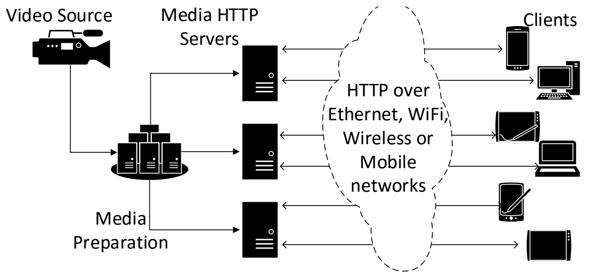


Fig. 2 Example for illustrating the operational process of video streaming.

devices (e.g., smart phones) to request and download the chunks from the web servers. Finally, clients can enjoy the video by playing back the sequence of downloaded video chunks in order.

2.2 4G LTE RRC State

The radio resource control (RRC) protocol belongs to the universal mobile telecommunication system, and it has many functions such as establishing and releasing connections, broadcasting system information, controlling state mode [11]. It is well known that the RRC state changes in the network activities, and different RRC states make different impacts on the instant power [12,13]. In 4G LTE networks, there are two main RRC state modes [13,3]:

– *RRC idle.* In this state mode, there is no mobile data transmission in LTE networks. (Note that, in this mode the common control channel (CCC) is assigned to the user equipment (UE) such as the smart phone, in order to probe the data activity.)
– *RRC connected.* This state mode is activated when the UE is prepared to transfer any application data [14].

Particularly, the RRC connected state mode can be further divided into three sub-modes, which may result in *different* power *consumption* at the UE side [13]:

– *Continuous reception.* When the UE is in the RRC idle state, if the data transmission request is detected in LTE networks, the UE shall switch immediately to this sub-mode (i.e., the continuous reception mode). In this mode, the 4G LTE radio interface is always opened, and the UE can continuously receive the mobile data. Particularly, it transfers the data using a dedicated control channel (DCC), instead of the CCC.
– *Short discontinuous reception (short DRX).* Once the UE is in the continuous reception mode, an timer T_{in} (known as the DRX inactivity timer) is to be started. If there is no data activity but T_{in} gets expired, the UE shall go to the short DRX mode (instead of directly going to the long DRX mode).
– *Long discontinuous reception (long DRX).* Once the UE is in short DRX mode, another timer T_{ins} (known as the short DRX cycle inactivity timer) is to be started, and the UE will probe periodically the data activity. If no any data transmission activity is detected, the UE shall go to the long DRX mode. (Otherwise, it changes back to the continuous reception mode.) We remark that, similar to the short DRX mode, the long DRX mode also employs a timer T_{inl} (known as the long DRX cycle inactivity timer). If T_{inl} expires, the RRC state shall shift to the RRC idle state.

A major feature of the RRC state transition is that, regardless of which RRC mode the UE holds currently, it shall instantly shift to the continuous reception state mode, once the data transmission activity is detected. This implies that the *promotion* of the RRC state is immediate. While the *demotion* of the RCC state is controlled by different "inactivity timers", it takes a long path. That is, the demotion process is not so fast.

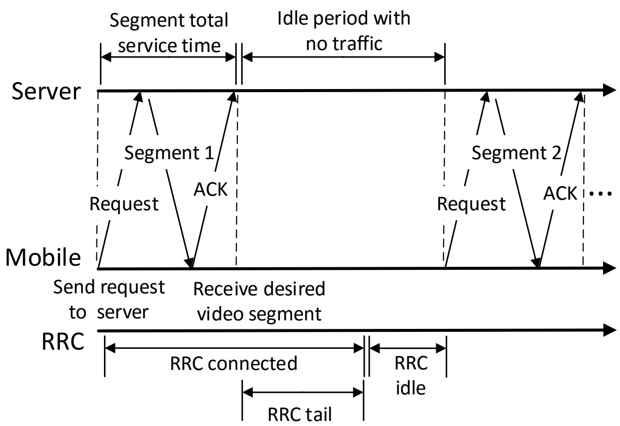


Fig. 3 Example for illustrating the timing scheme

2.3 Timing Scheme

Figure 3 illustrates an overview of timing scheme. (Some terminologies mentioned here will be used extensively when we make the quantitative analysis on the energy/power consumption.) The timing scheme involves two levels [12,13,4]:

– *Video streaming traffic level.* At this level, there is an interactive process between the mobile client and the server (see the upper part of Figure 3). The mobile client sends periodically the *video segment request* to the video server. Once the video server re-

Table 1 Notations used frequently

Notation	Description
T_s	the total service time of a segment
T_i	the idle period with no traffic
T_{rt}	the RRC tail time duration
T_{ri}	the RRC idle state time duration
T_{sd}	the playing back time duration of a video segment
$P_{T_{(\cdot)}}$	the power consumption during $T_{(\cdot)}$

ceives such a request, it sends back the *requested segment* to the mobile client. When the requested segment is transmitted successfully, the mobile client sends the ACK message to the video server, and then enters the idle period, in which no network traffic is produced.

- *RRC level.* At this level (see the bottom of Figure 3), the RRC state is in the connected mode during the segment transmission. Note that, after finishing the segment transmission, even if no new data transmission happens, the mobile client (actually, referred to the UE, recall Section 2.2) still stays at the RRC connected state within a time duration (before entering the RRC idle state mode). The state in such a time interval is referred to as "the RRC tail".

3 Energy Model

In this section, we present mathematical models for the mobile phone energy consumption related to the online video streaming in 4G LTE networks.

Notation. Let E be the total energy consumption of a mobile phone for network activities. Let S and C be the average download speed and the file size (of a *video segment*, i.e., chunk), respectively. Denote by T_s, T_i, T_{rt}, T_{ri} the total service time of a segment, the idle period with no traffic, the RRC tail time duration, the RRC idle state time duration (in each transmission cycle, c.f., Figure 3), respectively. Let $P_{T_{(\cdot)}}$ denote the power consumption of the mobile phone during $T_{(\cdot)}$ (e.g., $P_{T_{ri}}$ denotes the power consumption during the RRC idle state time duration T_{ri}), and P_{off} be the power consumption when playing back the video *offline*. Denote by T_{sd} the playing back time duration of a video segment, and let n be the number of total video segments for an online video streaming transmission. (For ease of reference, some notations used frequently are summarized in Table 1.)

Model. Recall Section 2.3, the timing scheme involves two levels: (i) the video streaming traffic level, and (ii) the RRC level. This implies that we can formulate our energy model from two routines. Specifically, for (i), we can have the following "basic" energy model:

$$E = \sum_{k=1}^{n}(\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t + \int_{0}^{T_i}(P_{T_i} - P_{off})\Delta_t) = \sum_{k=1}^{n}(\frac{C}{S}(P_{T_s} - P_{off}) + (T_{sd} - \frac{C}{S})(P_{T_i} - P_{off})) \quad (1)$$

The basic energy model is composed of two parts: the network transmission energy (see the part before "+" in Equation 1); and the network idle energy (see the part after "+" in Equation 1). Observe that T_i equals the sum of T_{rt} and T_{ri} (c.f., Figure 3). Thus, for (ii), we have the "advanced" model[3] as follows.

$$E = \sum_{k=1}^{n}(\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t + \int_{0}^{T_{rt}}(P_{T_{rt}} - P_{off})\Delta_t + \int_{0}^{T_{ri}}(P_{T_{ri}} - P_{off})\Delta_t) = \sum_{k=1}^{n}(\frac{C}{S}(P_{T_s} - P_{off}) + T_{rt}(P_{T_{rt}} - P_{off}) + (T_{sd} - \frac{C}{S} - T_{rt})(P_{T_{ri}} - P_{off})) \quad (2)$$

From our basic and advanced models above, we can further develop the following more specific models, which characterizes different scenarios:

(1) If $\frac{C}{S} > T_{sd}$ (e.g., the network condition is very weak), users can not watch the online video fluently. In this case, the energy model is formulated as follows.

$$E = \sum_{k=1}^{n}(\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t = \sum_{k=1}^{n}\frac{C}{S}(P_{T_s} - P_{off}) \quad (3)$$

(2) When $\frac{C}{S} = T_{sd}$, the energy model is formulated as follows.

$$E = \sum_{k=1}^{n}\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t = \sum_{k=1}^{n}T_{sd}(P_{T_s} - P_{off}) \quad (4)$$

(3) In the case that $\frac{C}{S} + T_{rt} = T_{sd}$, the energy model changes as follows. This model implies that, if there is a gap between the RRC tail power $P_{T_{rt}}$ and the RRC idle power $P_{T_{ri}}$, then there could be some power saving room. (Remark: our experiments discussed later shall validate this insight.)

[3] This model characterizes the "idle period with no traffic" in a more refined way.

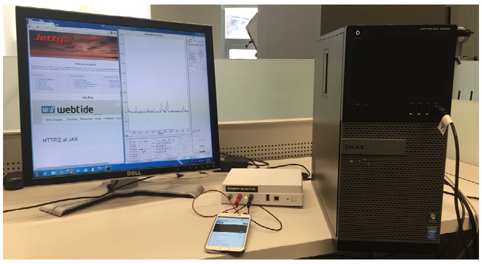


Fig. 4 Our devices

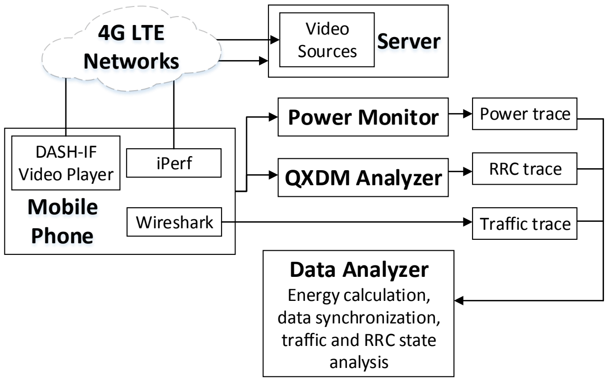


Fig. 5 Platform architecture

$$E = \sum_{k=1}^{n}\left(\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t + \int_{0}^{T_{rt}}(P_{T_{rt}} - P_{off})\Delta_t\right) = \sum_{k=1}^{n}\left(\frac{C}{S}(P_{T_s} - P_{off}) + T_{rt}(P_{T_{rt}} - P_{off})\right) \quad (5)$$

(4) When $0 < T_{sd} - \frac{C}{S} < T_{rt}$ (i.e., the next video segment transmission arrives before the *inactivity timer* gets expired; this could be the most common case of online video streaming in 4G LTE network), the power model changes into the following.

$$E = \sum_{k=1}^{n}\left(\int_{0}^{T_s}(P_{T_s} - P_{off})\Delta_t + \int_{0}^{T_{sd}-T_s}(P_{T_{rt}} - P_{off})\Delta_t\right) = \sum_{k=1}^{n}\left(\frac{C}{S}(P_{T_s} - P_{off}) + (T_{sd} - \frac{C}{S})(P_{T_{rt}} - P_{off})\right) \quad (6)$$

4 Architecture of Our System Platform

To investigate the power performance characteristics of video streaming in 4G LTE networks, we design a systematic platform. Our platform is composed of two major components: (1) *hardware suite*, which is responsible for video streaming system deployment and power data collection (see Figure 4); and (2) *software suite*, which is responsible for online video service, traffic data collection and analysis.

The architecture of our platform is shown in Figure 5. It works as follows. The mobile devices first access video sources on the service, generating three types of traces — the power, RRC, and traffic traces; these traces are collected by various tools such as wireshark, power monitor, and QXDM analyzer. After that, all the collected traces are to be processed by the data analyzer. In what follows, we introduce more details of our hardware and software suites.

• Our hardware suit mainly contains the following elements:

– We deploy a *video web server* that is connected with Internet using Ethernet cable, and is accessible via 4G LTE networks.
– We use an *Android smart phone* — Samsung Galaxy S5 (one of the most popular smart phones) — as our experimental mobile device. And we apply the *T-mobile 4G LTE high speed data plan* (in United States) to the Android smart phone for our experiments.
– We use a professional *mobile power meter* — the Monsoon power monitor (c.f., the white device in Figure 4) — for mobile power data collection. Remark that many power measurement applications in “Android App market” may just use fixed values (pre-defined by smart phone manufacturers) to estimate the power consumption [15]. Those power values could not reflect the exact real power values of smart phones [15]. We here thus use the Monsoon power monitor to extract mobile power data.

• Our software suite mainly includes a video player and some mobile data monitoring and diagnostic tools. Specifically,

– We deploy a *video streaming player* — DASH-IF player (http://dashif.org/software) — on the smart phone. (The DASH-IF player is the official DASH industry forum reference and production video streaming player.)
– Meanwhile, a widely used *mobile data analyzer* — Wireshark (https://www.wireshark.org) — is built on our smart phone, in order to collect the traffic information in 4G LTE networks.
– We use a network bandwidth measurement tool — iPerf (https://iperf.fr) — to support the TCP transmission test. Particularly, it can allow us to send packets to the target smart phone with the customized speed.
– The original *operating system* in our smart phone is Android 4.4.2 (Kitkat). We keep this system clean

without other redundant applications, for the veracity of measurement.

- On the other hand, a servlet engine — Jetty (http://www.eclipse.org/jetty) — is deployed on the web server.
- To record the 4G LTE RRC states, we employ a real-time data collection and diagnostic logging tool – the QUALCOMM eXtensible Diagnostic Monitor (QXDM) (https://www.qualcomm.com).
- To assist our analysis, we also develop an assistant measurement algorithm (detailed in Section 4.1), which is based on the energy model.

4.1 Assistant Measurement Algorithm

Based on our energy models, we propose an algorithm for the power consumption analysis. It will assist us to understand roughly the energy improvement room. At a high level, our algorithm mainly exploits three types of traces (of the current test video) and the *reference traces*[4]. Based on these information, it computes the *total segment service time*, the *power information* during the RRC *idle* (resp., *tail*) state, and two types of *energy consumption*. Finally, it returns some values (which could reflect some energy improvement room).

To understand our algorithm in detail, we need clarify some notations. Let S_t be the file size of the current test video, P_l, T_l and R_l be its power trace, traffic trace, and RRC trace, respectively; and P_{vt} be the power consumption for this video playing. Let RT be a set of m reference traces; and for a reference trace $rt_j \in RT$, let rs_j be its corresponding video file size. For clearness, we denote by RS the set of such type of file sizes, i.e., $RS = \{rs_1, rs_2, ..., rs_m\}$. Let Δ_{op} ($\in (0, 1)$) be a threshold used to determine whether or not a part/phase has the promising saving room. Let I_e be the percentage of the possible energy saving, I_t, I_{rt} and I_{ri} be three indicators showing whether there is some improvement room for the *transmission*, for the *RRC tail*, and for the *RRC idle*, respectively. Algorithm 1 shows the pseudocodes of our algorithm. (Note that, the main notations described in this section are summarized in Table 2, for ease of reference.)

Firstly, our algorithm initializes some values and finds the appropriate reference trace (from a set RT of m reference traces) (Lines 1-5). Then, it analyses traces based on two routines:

- (1) The segment service period. In this phase, (i) it extracts each segment service time $T_l^{(\cdot)}$ from the traffic trace T_l; (ii) it uses T_{sr}^j (the total segment service time of the "selected" jth reference trace) to divide the sum of each segment service time; (iii) it uses the above result to minus the number "1", and then to see whether I_t needs to be updated (Lines 6-9).
- (2) The idle period without no traffic. In this phase, (i) it extracts the power information $P_{T_{ri}^{(\cdot)}}$ (resp., $P_{T_{rt}^{(\cdot)}}$) in each segment's RRC idle time $T_{ri}^{(\cdot)}$ (resp., RRC tail time $T_{rt}^{(\cdot)}$); (ii) it uses P_{vt} to divide the average value of the all segments' RRC idle state (resp., RRC tail state) power consumption; (iii) it computes the subtraction between the result above and the number "1", and then to determine whether I_{ri} (resp., I_{rt}) needs to be updated (Lines 10-17).

Finally, it computes I_e based on E_t and E_r (Lines 18-19), where E_t and E_r refer to the total energy consumption for the test trace and for the reference trace respectively, which can be computed based on the energy models described previously.

Algorithm 1 Power consumption analysis

Input: Power trace P_l, traffic trace T_l, RRC trace R_l, video file size S_t, threshold Δ_{op}, video playing power P_{vt}, a set RT of m reference traces, and the set RS of their corresponding video files sizes
Output: Percentage of possible energy saving I_e, three indicators I_t, I_{rt} and I_{ri}
1: Set $I_e = 0, I_t = 0, I_{rt} = 0, I_{ri} = 0$;
2: **for** each $1 \leq j \leq m$ **do**
3:   **if** $S_t \neq rs_j$ **then**
4:     Continue;
5:   **end if**
6:   Extract $T_l^1, T_l^2, ..., T_l^n$ from T_l;
7:   **if** $\frac{\sum_{k=1}^{n} T_l^k}{T_{sr}^j} - 1 > \Delta_{op}$ **then**
8:     $I_t = 1$;
9:   **end if**
10:   Check RRC states during each $T_i^{(\cdot)}$;
11:   Extract $(P_{T_{rt}^1}, P_{T_{rt}^1}, ..., P_{T_{rt}^n})$ and $(P_{T_{ri}^1}, P_{T_{ri}^1}, ..., P_{T_{ri}^n})$ from P_l;
12:   **if** $\frac{\sum_{k=1}^{n} \int_0^{T_{rt}^k} (P_{T_{rt}^k} - P_{off}) \Delta_t)}{(\sum_{k=1}^{n} T_{rt}^k) \cdot P_{vt}} - 1 > \Delta_{op}$ **then**
13:     $I_{rt} = 1$;
14:   **end if**
15:   **if** $\frac{\sum_{k=1}^{n} \int_0^{T_{ri}^k} (P_{T_{ri}^k} - P_{off}) \Delta_t)}{(\sum_{k=1}^{n} T_{ri}^k) \cdot P_{vt}} - 1 > \Delta_{op}$ **then**
16:     $I_{ri} = 1$;
17:   **end if**
18:   Calculate energy consumption E_t, E_r;
19:   $I_e = \frac{E_t - E_r}{E_t}$;
20: **end for**
21: **Return** I_e, I_t, I_{rt} and I_{ri}

[4] Remark that these reference traces include themselves' power traces and traffic traces. In our experiments the reference traces are obtained by "downloading video segment files".

Table 2 Main notations used in our algorithm

Notation	Description
S_t	the file size of the current test video
P_l, T_l, R_l	the power, traffic, and RRC traces
P_{vt}	the power consumption for video playing
RT	the set of m reference traces
rt_j, rs_j	the reference trace and its file size
RS	the set of file sizes
Δ_{op}	the analysis threshold
I_e	the percentage of the possible energy saving
I_t, I_{rt}, I_{ri}	three indicators for transmission and RRC
$T_l^{(\cdot)}$	the segment service time
T_{sr}^j	the total segment service time of the jth reference trace
E_t/E_r	the test/reference trace's energy consumption

5 Experimental Results & Analyses

In this section, we cover the experimental results and our analyses in detail. Specifically,

- Section 5.1 presents the results related to the *basic* power consumption of our smart phone. This set of results show us some baselines of power consumption, which are helpful for understanding the *additional* power consumption that happens in the subsequent tests.
- Then, to figure out how much energy is consumed by the video streaming, in Section 5.2 we compare the power consumption of offline video playing and online video playing in 4G LTE; also, we cover the results related to *hot points*[5], in order to figure out the energy saving possibility from a general point of view.
- To get a deeper understanding on the power consumption of network activities (sometimes, known as *network part*), Section 5.3 studies the impact of RRC states on the power consumption; and also studies the relationship between the transmission pattern and the power consumption.

5.1 Basic Power Consumption

In this section we cover the results related to the basic power consumption of our mobile phone. These power values will be used as the baselines, which can help us figure out the additional power consumption related to the video streaming.

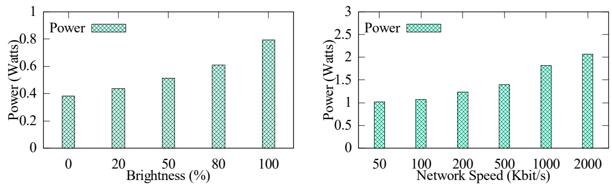
[5] In this paper the hot point refers to a measured point whose power consumption is larger than some preset threshold. This threshold is asked to be larger than the power consumption of most measured points (that are extrated from the power trace of offine video playing).

Table 3 Symbols and descriptions.

Power cons.	Working condition
P_i	*Only* keep the mobile on, anything other is off
P_{iW}	Keep alive in idle state and with WiFi connected
P_{iL}	Keep alive in idle state and with LTE data service
P_{is100}	Keep the screen on and with 100% brightness
P_{basic}	Keep alive in idle state with LTE data service, 100% brightness, and the video player is opened but nothing is playing back

Table 4 Power consumption of different components.

Component	Galaxy S5 (mW)
Standby	20.7
Screen	793.4
WiFi maintenance	14.4
LTE maintenance	3.15
Basic	1060



(a) (b)

Fig. 6 Power consumption under different brightnesses and network conditions: (a) various brightness; (b) network condition.

As we know, some hardware components inside a smart phone can work in different conditions, it is interesting to learn the power consumption when different conditions are employed, and it is also useful for us to better understand the subsequent findings. In our experiments, we measure the power consumption of the smart phone in *different conditions* (shown in Table 3).

Table 4 shows the experimental results which are derived from the collected values in Table 3. (Note that, the basic power consumption refers to the value of P_{basic}, and the "standby" power consumption refers to the value of P_i. The "screen" power consumption is computed through $P_{is100} - P_i$; other two components' power information is obtained similarly.)

From Table 4 we can see that: (i) the power consumption from the network connection maintenance for WiFi or LTE service (during the idle period) is pretty small; and (ii) the power consumption from the screen of the smart phone is non-trivial. Inspired by curiosity, we also study the power consumption when different brightnesses are used. The results are shown in Figure 6(a). As we expected, the *screen* power consumption grows with the increase of the brightness.

Table 5 Power consumption of different parts. In this table the "online, offline, netw, video and basic PCs" refer to P_{on}, P_{off}, P_{netw}, P_{video} and P_{basic}, respectively.

	Online PC (mW)	Offline PC (mW)	Netw PC (mW)	Video PC (mW)	Basic PC (mW)
Case 1	3935(100%)	2533(64.37%)	1402 (35.63%)	1473 (37.43%)	1060 (26.94%)
Case 2	3854(100%)	2533(65.72%)	1321 (34.28%)	1473 (28.22%)	1060 (27.50%)
Case 3	3393(100%)	2533(74.65%)	860 (25.35%)	1473 (43.41%)	1060 (31.24%)
Case 4	3396(100%)	2533(74.58%)	863 (25.42%)	1473 (43.37%)	1060 (31.21%)
Case 5	3534(100%)	2533(71.66%)	1001 (28.34%)	1473 (41.67%)	1060 (29.99%)
Case 6	2916(100%)	2385(81.77%)	531 (18.23%)	1325 (45.42%)	1060 (36.35%)
Case 7	2899(100%)	2385(82.24%)	514 (17.76%)	1325 (45.69%)	1060 (36.56%)
Case 8	2939(100%)	2385(81.13%)	554 (18.87%)	1325 (45.07%)	1060 (36.06%)

5.2 Power Consumption of Video Streaming

5.2.1 Offline and online video playing

This section reports the experimental results related to the power consumption of the *offline video playing* and *online video playing* (in 4G LTE networks). Note that, for the offline video playing, its power consumption P_{off} can be divided into two parts: (i) the basic power consumption P_{basic}; and (ii) the power consumption for playing the video, say P_{video}. That is, $P_{off} = P_{video} + P_{baisc}$. On the other hand, for the online video playing, its power consumption (say P_{on}) can be computed as $P_{on} = P_{off} + P_{netw}$, where P_{netw} denotes the power consumption of *network activities*. Based on the above intuition, we can naturally obtain P_{video} and P_{netw} when P_{on}, P_{off} and P_{basic} are available.

To obtain P_{on} and P_{off}, in our experiments we extract eight typical *power traces* from a large amount of traces, which could allow us to get a general understanding on the power consumption. The videos used in our experiments are obtained from the site (http://www.digitalprimates.net). Table 5 shows the experimental results (notice: Cases 1-8 shown in the table correspond to these eight traces). In this table, we can see that the power consumption percentage of the network part is from 17.76% to 35.63%. It is one of the major contributors for the power consumption of online video playing[6].

On the other hand, we can see that P_{netw} changes a lot for different cases (Cases 1-8). This reminds us that the real-time network circumstance could make impact on the power consumption of the video segment transmission. To further verify it, and figure out what relationship between the *network speed* and the *power consumption of video segment transmission* is, we employ iPerf (c.f., Section 4) to assist us. In brief, we use iPerf send packets to the smart phone using the customized speeds, and extract the power consumption information when different network speeds are used. Note that, in our experiments we send more packets (in terms of the total size) when the customized speeds are higher. This way, it shall be easier for us to extract the power consumption information of the network activities.

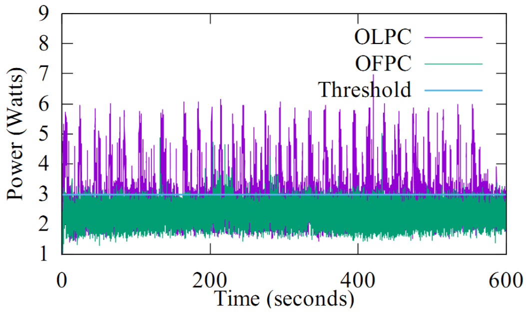


Fig. 7 Hot points analysis.

Figure 6(b) presents the results. At the first sight, it seems that the higher the network speed is, the more the power consumption of network part is. The fact is not so. To understand, let us focus on the growth speed. We can see that, when the network speed increases rapidly (from 50 to 2000 Kbit/s, 40×), the power consumption increases slowly (from 1 to 2 Watts, about 2×). This implies that, given the same size of video file, the power consumption of network part at 2000 Kbit/s is about $\frac{1}{20}$ of the one at 50 Kbit/s. That is, the higher the network speed is, the less the power consumption of network part is.

5.2.2 Hot Points

It is not hard to realize that, for a *traffic point*[7] in current 4G LTE networks, there could be no much chance

[6] Remark that the results in this table are obtained when we set the screen brightness to 100%. It is not hard to know that, if we set it to a lower value, the power consumption percentage of the network part shall increase. We emphasize that in this paper we are mainly interested in the power consumption of network part. The experimental results related to other parts could be mainly used for reference, whereas we believe those results are still useful, and thus we present them here.

[7] A traffic point (resp., an idle point) refers to a measured point that has (resp., no) the data transmission.

Table 6 Hot points and traffic points. The basic measured unit used in our experiment is 0.02 seconds.

	Traffic Points	Hot Points	TP Ratio
Case 1	5806	10249	56.65%
Case 2	5742	9057	63.40%
Case 3	4011	6066	66.12%
Case 4	2977	6081	48.96%
Case 5	2675	6054	44.19%
Case 6	2120	9449	22.44%
Case 7	2203	8992	24.50%
Case 8	2102	10198	20.61%

Table 7 Saving room, indicators I_t, I_{rt} and I_{ri}.

	Streaming Energy(J)	Reference Energy(J)	Saving Room	I_t	I_{rt}	I_{ri}
Case 1	409.2	124.3	69.62%	1	1	0
Case 2	343.5	124.3	63.82%	1	1	0
Case 3	213.8	124.3	41.86%	1	1	0
Case 4	235.5	124.3	47.19%	1	1	0
Case 5	273.6	124.3	54.56%	1	1	0
Case 6	318.9	119.0	62.69%	1	1	0
Case 7	306.5	119.0	61.17%	1	1	0
Case 8	352.6	119.0	66.25%	1	1	0

to reduce its power consumption, since the network connection is necessary in this case. On the other hand, for a non-traffic point, in theory it should consume less power, since no 4G LTE traffic happens. This implies that, if the number of *hot points* (c.f., Footnote 3) matches exactly the number of traffic points, we shall have no much power saving room for the video streaming. In other words, the less the ratio of traffic points and hot points (i.e., $\frac{number\ of\ traffic\ points}{number\ of\ hot\ points}$) is, the more saving room we shall have. In what follows, we analyse the hot points that appeared in our video streaming tests, and to see whether there is some possible saving room.

To determine the threshold, we conservatively set the threshold to 1.5 times of the average power consumption of offline video playing. This way, it could dominate, in terms of power consumption, most of measured points (extracted from the power trace of offline video playing). Figure 7 plots the power consumption of offline video playing and online video playing. We can see that, there are more than 95% of measured points whose power consumption is below our threshold. This implies that the threshold we chose should be meaningful.

Based on this threshold, we extract hot points and traffic points from eight typical power traces. Table 6 shows our results. We can see that the ratio of traffic points and hot points varies from 20.61% to 66%. This means that we still have much power saving room, which comes from those hot points without data traffic.

5.3 A Deep Study

Our experiments discussed before have presented some preliminary results, whereas it is still unclear (1) which specific phase in the network part could be promising, if someone is prepared to optimize the network power consumption; and (2) how much the power saving room could be achieved from the network part. This section attempts to provide some evidences that could help us answer the questions raised above.

5.3.1 Locating Our Object

We here use Algorithm 1 presented in Section 4 to assist us[8] . To compute the extreme energy saving space of online video streaming in 4G LTE networks, we download the video segment files all at once[9], and collect the power traces and the traffic traces, which will be used as the *reference traces* (recall Section 4.1).

Based on our algorithm presented before, we obtain our case study results, which are shown in Table 7. From this table, we can see that, the energy saving room (for online video streaming) varies from 41.86% to 69.62%[10]. It is non-trivial. On the other hand, the results show us that, I_t, I_{rt} and I_{ri} are always 1, 1, and 0, respectively (notice: these notations have already been defined in Section 4.1.). This implies that, there could be some improvement space for the transmission, and for the RRC tail, while it could be not promising for the RRC idle.

While our algorithm helps us point out the possible objects having promising saving room, it is still unclear (i) why the transmission pattern makes impact on the power consumption of video streaming; and (ii) how the RRC tail makes impact on the power consumption. It is clearly interesting and also pretty important to figure out these issues. In what follows, we focus on answering these two questions using abundant evidences together with insightful analysis.

5.3.2 Packet Level Analysis

Recall previous sections, our experiments use two manners to obtain the video segment files, in order to study

[8] In our experiments, Δ_{op} is conservatively set to 5%. In other words, if the potential saving room of a part or phase is less than 5%, we conceive conservatively it is not promising. In the following experiments, we shall not focus on studying such a part/phase.

[9] Remark that it is the extreme case; this way, we can easily find out the extreme power saving room for online video streaming.

[10] The reasons (the saving room varies) could be diversified (e.g., different configurations of network conditions).

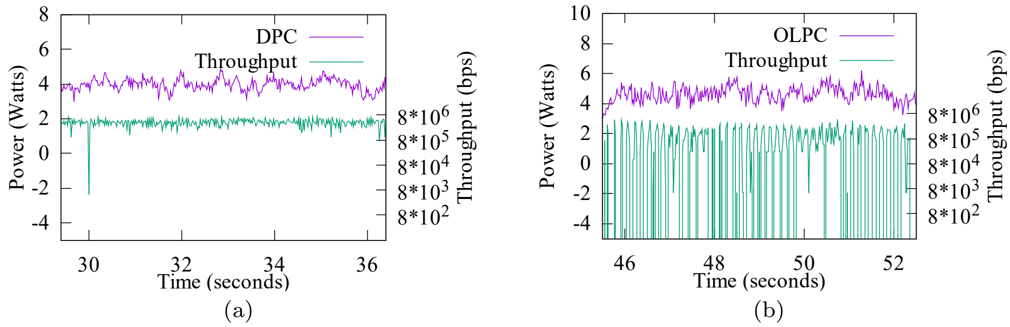


Fig. 8 Zoom in the traffics of different transmission patterns: (a) downloading pattern; (b) video streaming pattern.

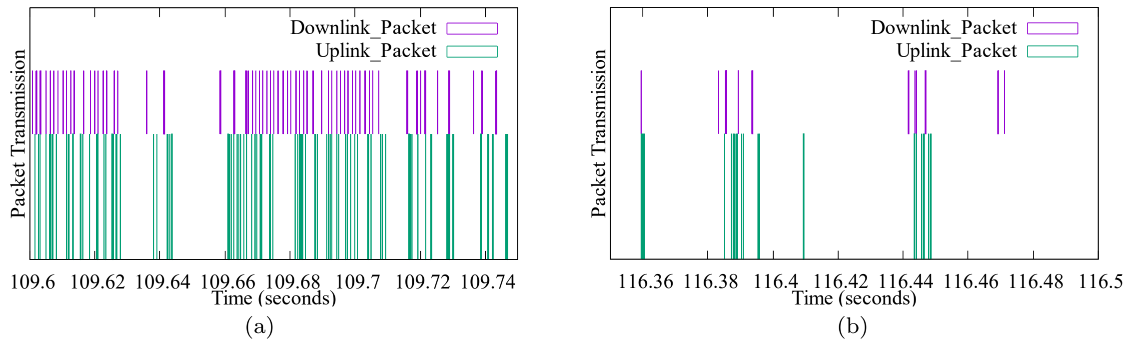


Fig. 9 Packet level transmissions in different transmission pattern: (a) downloading pattern; (b) video streaming pattern.

the power consumption of network part. One is to download video segment files all at once (known as *downloading pattern*), another is to conduct online video playing in 4G LTE networks (known as *video streaming* pattern; in this case video segment files are downloaded in a linear fashion). This section studies these two transmission patterns in detail, and then reveals our findings.

Zoom in Data Transmission. We zoom into the traffic trace of video streaming patten, and of downloading pattern, respectively. We also set the basic measured unit to 0.02 seconds (as the same as the setting when we previously studied hot points).

Figure 8 shows our results. For the downloading pattern (c.f., Figure 8(a)), it is easy to see that (1) the downloading process is continuous; (2) almost no *idle point* (recall Footnote 5) is mixed with traffic points; and (3) a relatively stable and high downloading speed can be kept during the period of video downloading. Overall, almost no unnecessary energy is consumed.

In contrast, for the video streaming patten (see Figure 8(b)), we can see that (1) there are lots of *idle points*; (2) it does not maintain persistently a high downloading speed (naturally, it shall take more time to download all the video segments, compared to the downloading pattern); (3) the downloading progress is discrete; (4) many idle points are mixed up with traffic points; and (5) these idle points (without data transmission) also consume a lot of energy. The above facts reveal that, if these idle points can be eliminated, a large amount of energy could be saved.

To this step, a natural question is: why there are many idle points. A possible answer is that, in the video streaming pattern, the mobile device needs to send the HTTP “get” message to the video server for each segment one by one, this possibly incurs some idle points (naturally it shall incur some delays and also some energy consumption). Another possible reason is that, the video segment transmission is usually finished before the TCP connection reaches a high speed (recall that the TCP connection needs some time to warm up). This renders that the streaming pattern can not achieve a high transmission speed, incurring the idle points. To further understand these reasons, we analyse the packet information related to these two transmission patterns. Our experimental results below further validate our analysis.

Packet Transmission. We extract the packet information from the traffic trace of downloading pattern, and from the one of video streaming pattern, respectively. Figure 9 shows the results. Compare Figures 9(a) and 9(b), we can easily see that (1) in the same time duration, more packet transmissions can be done in the downloading pattern; and (2) the time gaps between two packet deliveries in the downloading pattern are smaller; and (3) the transmission points in the downloading pattern are more dense. The above facts actually explain, from another point of view, why the

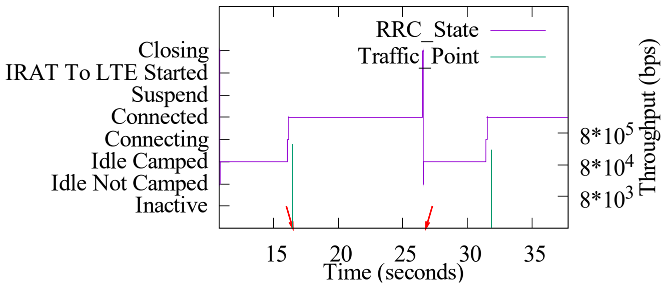


Fig. 10 RRC states in single packet transmission.

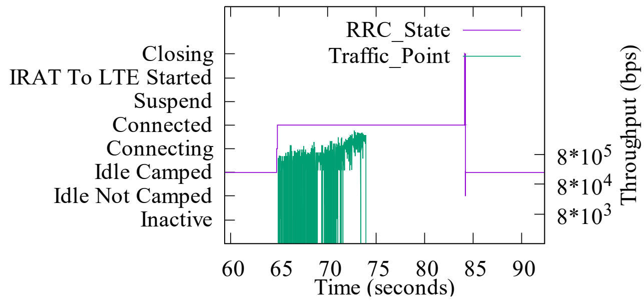


Fig. 11 RRC states when multiple packets are sent at each transmission cycle.

video streaming pattern cannot achieve a stable and high downloading speed, and why it contains many idle points. Also, it reminds us that, if the video streaming pattern can achieve more efficient traffics, a lot of energy could be saved.

5.3.3 RRC State

Recall Section 5.3.1, a natural question has been raised: how the RRC tail makes impact on the power consumption. In this section, we attempt to investigate the power consumption of the RRC stages (especially, the RRC tail).

To study the power consumption of the RRC stages, we establish the TCP connection between the mobile phone and the web server, and send packets from one to another periodically. This way, it allows us to study the RRC states easily[11]. Firstly, we study the RRC states by sending a single packet at each transmission cycle. This could give us some preliminary intuition. In our experiments, the transmission cycle is set to 15 seconds. We use the QXDM tool to collect *local RRC states*[12].

Figure 10 shows the results. From this figure we can see that, the entire RRC state transition cycle follows the following pattern[13]: *Idle camped* -> *Connecting* -> *Connected* -> *Closing* -> *Idle Not Camped* -> *Idle Camped*. On the other hand, we can also see that, the local RRC state transitions shown in this figure are basically consist with the RRC state transitions mentioned in Section 2.2. For example, (1) if any data transmission activity is detected, the local RRC state shall shift to the "Connected" state; (2) the local RRC state still maintains the "Connected" state, although there is no data transmission after finishing a transmission activity (see the *traffic points* shown in this figure). Particularly, we here can observe that the RRC tail lasts about 10 seconds (see the interval between a pair of red arrows in this figure).

Furthermore, we also conduct experiments using multiple packets at each transmission cycle. Compared with the *single packet test*, this could reflect the process of video segment downloading in a more refined manner. (We set the transmission cycle to 60 seconds, and 5000 packets are to be sent each time.) Figure 11 shows us the results. From this figure, we can observe an obvious feature: when a data transmission activity has not finished, another data transmission is coming. That is, multiple data transmission activities are overlapping in some time intervals.

We also plot the power trace and the traffic trace, respectively. The results are shown in Figure 12. By comparing these two sub-figures, we can easily find that, even if the data traffic finishes, the power consumption is not immediately reduced to a very low value. In contrast, it still retains a relatively high value for about 10.3 seconds, as similar as the RRC tail does. Finally, to get an overall and deeper understanding on the RRC state, Table 8 reports the precise RRC profiling parameters. We can see that the transitional states only consume a little time (less than 0.1 seconds in total), while

[11] Note that, we have attempted to observe directly the RRC states based on the process of video downloading. It, however, is not easy to observe the RRC phases (e.g., the RRC tail). In addition, the essence of video downloading is the transmission of packets. Here we thus simulate the process of the video downloading through sending the packets.

[12] Note that, the QXDM tool can record eight local RRC states: Closing, IRAT To LTE Started, Suspend, Connected, Connecting, Idle Camped, Idle Not Camped, and Inactive. We can see that the *local RRC states* mentioned here is similar to, but somewhat different from the *RRC states* mentioned before. The RRC states mentioned before is a standard protocol. The local RRC states introduced by QXDM is used to differentiate the network activities in a more precise manner at the mobile client end.

[13] This pattern involves five instant RRC states, while other three instant RRC states (i.e., *IRAT To LTE Started*, *Suspend* and *Inactive*) are not covered. This is mainly because our experiments are always conducted in 4G LTE networks, those three local RRC states naturally are not appeared in our traces. On the other hand, we can view the five local RRC states as following: the local "Connected" state corresponds to the RRC connected state; the local "Idle camped" state corresponds to the RRC idle state. The other three local states are the transitional states between the RRC connected and idle states.

Table 8 The RRC profile, including the time duration and the power consumption.

RRC parameter	**Value**	**Description**
Connecting	0.074 s	Time duration for changing from the RRC idle to the RRC connected
Closing	0.07 s	Time duration for changing from the RRC connected to the RRC idle
Camping	0.004 s	RRC camping time
Tail	10.3 s	Total time duration of the RRC tail
Aver_addi_tail_power_cons	0.223 w	Average additional power consumption of the entire RRC tail

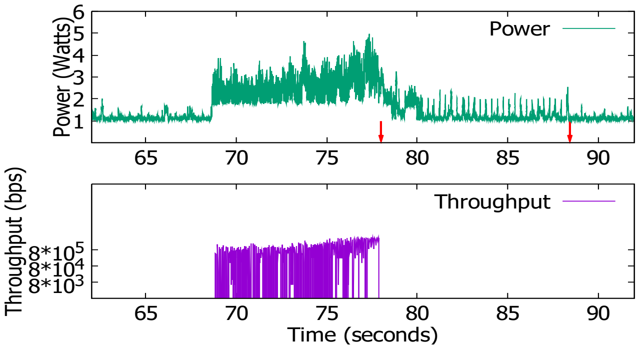


Fig. 12 RRC tail power analysis. The RRC tail is the interval between the pair of red arrows.

the RRC tail lasts for about 10.3 seconds. The above experimental results reveal us that, the RRC tail takes a relatively long time duration, and it consumes a lot of energy while no data traffic happens. This reminds us that, if the number of RRC tails can be reduced, a lot of energy could be saved.

6 Related Work

In this section we review previous work by classifying them into several categories (c.f., Subsections 6.1∼6.3). Note that, to save space we here do not exhaustively enumerate all the related work (most of which could be found in the references therein).

6.1 Evaluation/Analysis of Power Consumption in Wireless Networks

A lot of effort has been made in the literature to analyze/evaluate the energy or power consumption in wireless networks. For example, Wang *et al.* [16] presented an analytical formula for estimating power consumption in large-scale wireless sensor networks. Zhu *et al.* [17] presented the design and implementation of a housebuild experimental platform for the energy management and exploration on wireless sensor networks. Li *et al.* [18] proposed an online algorithm with low computational complexity and deployment overhead in multichannel wireless networks. Xiang *et al.* [19] proposed an energy model for Poisson-Voronoi tessellation cellular networks. Lupia and Rango [20] discussed a mathematical model for power evaluation on mobile ad-hoc networks. Tekkalmaz and Korpeoglu [21] presented a distributed algorithm to increase the lifetime of wireless sensor networks. Narendran *et al.* [22] studied transmission power control algorithms for cellular networks. The energy-efficient architectures of the Internet of Things were discussed in [23–25]. A recent survey [26] covered traffic offloading techniques in wireless networks, and another survey [27] discussed *general* power control issues in wireless sensor networks. Most of prior works in these literature focused on energy consumption of *sensors*, and assumed to be in the general wireless networks environment. We here are interested in digging out energy consumption characteristics of *video streaming* in *4G LTE networks*.

6.2 Power Consumption Related to Video Streaming

There are also many papers investigating the power consumption of video streaming. For example, Sharrab and Sarhan [28] presented models for power consumption of live video streaming systems. Yu *et al.* [29] addressed the problem of resource allocation for video multicast in 4G wireless systems. Sheu *et al.* [30] proposed a resource allocation scheme to support scalable-video multicast for WiMAX relay networks. Go *et al.* [31] presented a seamless high-quality HTTP adaptive streaming algorithm. Ukhanova *et al.* [32] presented an analysis on the power consumption of video data transmission in 3G mobile wireless networks. An online solution, based on users' habit, is proposed to save energy [33]. Many other papers also devoted to optimizing the energy consumption. For example, in [12] the authors made efforts to save the portion of energy consumed by unnecessarily keeping the mobile device's radio in its "Active" mode. Hoque *et al.* [34] proposed a download scheduling algorithm to save energy. Also, Zhang *et al.* [35] proposed a hotword detection system for smart devices, in order to save energy. In these literature, the latest 4G LTE networks environment and the corresponding RRC impact[14] are not covered, while our work

[14] The 4G LTE RRC states are different from those in 3G or previous mobile networks [14].

focuses on power characteristics of video streaming in *4G LTE networks*.

6.3 Power Consumption in 4G LTE Network

We also realize that many researchers studied the power consumption in 4G LTE networks. For example, Huang *et al.* [13] studied the network performance of 4G LTE networks and other types of mobile networks. Imran *et al.* [10] proposed a novel energy efficient MAC scheme for LTE-Advanced networks. Holtkamp *et al.* [36] presented a new radio resource management algorithm. A deployment tool for future green wireless access networks is proposed in [8] for optimizing the power consumption. To reduce the power consumption in LTE data scheduling, Tung *et al.* [37] proposed the Dynamic scheduling with eXtensible allocation and Dispersed offsets (DXD) scheme. These works focused on power consumption in the general 4G LTE networks. They did not cover the specific video scenario, which motivates our work (a preliminary version can be found in [38]).

Others. Besides the works mentioned before, there are many other related works, e.g., *video transmission patterns* [39, 40], *multimedia streaming delivery optimization* [41, 42], *user-perceived Quality-of-Experience in Internet video applications* [43, 44]. These works could be related and complementary to our work, while they are *clearly* different from our work.

7 Conclusion

Motivated by the real world energy consumption issue, we investigated the relationship between mobile device's power performance characteristics and the behaviours of the video streaming in 4G LTE networks. We formulated the energy models together with an algorithm that is used to assist our analysis. Particularly, we designed a systematic platform, based on which we conducted a comprehensive and also deep analysis on the power performance of video streaming in 4G LTE networks. Our experiments revealed us a series of valuable findings, which could be used as the important references for related studies. In the future, we plan to design and implement a system to optimize the network energy consumption of video streaming in 4G LTE networks.

References

1. GSMA global mobile economy report 2015. http://gsmamobileeconomy.com/global/.
2. Ericsson mobility report: on the pulse of the networked society. http://www.ericsson.com/mobility-report, 2015.
3. Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *ACM SIGCOMM Computer Communication Review*, 44(4):187–198, 2015.
4. Thomas Stockhammer. Dynamic adaptive streaming over HTTP –: standards and design principles. In *MMSys*, pages 133–144, 2011.
5. Alex Zambelli. IIS smooth streaming technical overview. *Microsoft Corporation*, 3:40, 2009.
6. Mike Williams. Why are mobile phone batteries still so crap. http://www.techradar.com/news/phone-and-communications/mobile-phones/why-are-mobile-phone-batteries-still-so-crap–1162779.
7. Sahil Suneja, Vishnu Navda, Ramachandran Ramjee, and Eyal de Lara. Envi: energy efficient video player for mobiles. In *CellNet*, pages 25–30, 2013.
8. Margot Deruyck, Emmeric Tanghe, David Plets, Luc Martens, and Wout Joseph. Optimizing LTE wireless access networks towards power consumption and electromagnetic exposure of human beings. *Computer Networks*, 94:29–40, 2015.
9. Opeoluwa Tosin Eluwole and Mahboubeh Lohi. Coordinated multipoint power consumption modeling for energy efficiency assessment in LTE/LTE-advanced cellular networks. In *ICT*, pages 1–6, 2012.
10. Reema Imran, Mutaz Shukair, Nizar Zorba, and Christos Verikoukis. An energy saving strategy for LTE-A multiantenna systems. *Mobile Networks and Applications*, 20(5):692–700, 2015.
11. UMTS RRC Protocol specification (version 12.4.0 Release 12), 2015.
12. Shuo Deng and Hari Balakrishnan. Traffic-aware techniques to reduce 3G/LTE wireless energy consumption. In *CoNEXT*, pages 181–192, 2012.
13. Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *MobiSys*, pages 225–238, 2012.
14. 3GPP TR 25.813: Radio interface protocol aspects (V7.1.0), 2006.
15. Battery doctor: Get your phone lasting longer. http://www.cmcm.com/en-us/battery-doctor/.
16. Hui Wang, H Eduardo Roman, Liyong Yuan, Yongfeng Huang, and Rongli Wang. Connectivity, coverage and power consumption in large-scale wireless sensor networks. *Computer Networks*, 75:212–225, 2014.
17. Nanhao Zhu and Athanasios V Vasilakos. A generic framework for energy evaluation on wireless sensor networks. *Wireless Networks*, 22(4):1199–1220, 2016.
18. Peng Li, Song Guo, and Jiankun Hu. Energy-efficient cooperative communications for multimedia applications in multi-channel wireless networks. *IEEE Transactions on Computers*, 64(6):1670–1679, 2015.
19. Lin Xiang, Xiaohu Ge, Cheng-Xiang Wang, Frank Y Li, and Frank Reichert. Energy efficiency evaluation of cellular networks based on spatial distributions of traffic load and power consumption. *IEEE Transactions on Wireless Communications*, 12(3):961–973, 2013.
20. Andrea Lupia and Floriano De Rango. Evaluation of the energy consumption introduced by a trust management scheme on mobile ad-hoc networks. *Journal of Networks*, 10(4):240–251, 2015.
21. Metin Tekkalmaz and Ibrahim Korpeoglu. Distributed power-source-aware routing in wireless sensor networks. *Wireless Networks*, 22(4):1381–1399, 2016.

22. K. Narendran, R. M. Karthik, and Krishna M. Sivalingam. Iterative power control based admission control for wireless networks. *Wireless Networks*, 22(2):619–633, 2016.
23. K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, and J. Wu. A survey on energy Internet: architecture, approach and emerging technologies. *IEEE System Journal*, 12, 2016. doi: 10.1109/JSYST.2016.2639820.
24. Hui Jiang, Kun Wang, Yihui Wang, and Min Gao. Energy big data: A survey. *IEEE Access*, 4:3844–3861, 2016.
25. Kun Wang, Yihui Wang, Yanfei Sun, Song Guo, and Jinsong Wu. Green industrial Internet of Things architecture: An energy-efficient perspective. *IEEE Communications Magazine*, 54(12):48–54, 2016.
26. Xianfu Chen, Jinsong Wu, Yueming Cai, Honggang Zhang, and Tao Chen. Energy-efficiency oriented traffic offloading in wireless networks: a brief survey and a learning approach for heterogeneous cellular networks. *IEEE Journal on Selected Areas in Communications*, 33(4):627–640, 2015.
27. Nikolaos A. Pantazis and Dimitrios D. Vergados. A survey on power control issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials (COMSUR)*, 9(1-4):86–107, 2007.
28. Yousef O Sharrab and Nabil J Sarhan. Aggregate power consumption modeling of live video streaming systems. In *MMSys*, pages 60–71, 2013.
29. Ya Ju Yu, Pi Cheng Hsiu, and Ai Chun Pang. Energy-efficient video multicast in 4G wireless systems. *IEEE Transactions on Mobile Computing*, 11(10):1508–1522, 2012.
30. Jang Ping Sheu, Chien Chi Kao, Shun Ren Yang, and Lee Fan Chang. A resource allocation scheme for scalable video multicast in WiMAX relay networks. *IEEE Transactions on Mobile Computing*, 12(1):90–104, 2013.
31. Yunmin Go, Oh Chan Kwon, and Hwangjun Song. An energy-efficient HTTP Adaptive Video Streaming with networking cost constraint over heterogeneous wireless networks. *IEEE Transations on Multimedia*, 17(9):1646–1657, 2015.
32. Anna Ukhanova, Evgeny Belyaev, Le Wang, and Søren Forchhammer. Power consumption analysis of constant bit rate video transmission over 3G networks. *Computer Communications*, 35(14):1695–1706, 2012.
33. Wenjie Hu and Guohong Cao. Energy-aware video streaming on smartphones. In *INFOCOM*, pages 1185–1193, 2015.
34. Mohammad Ashraful Hoque, Matti Siekkinen, and Jukka K Nurminen. Using crowd-sourced viewing statistics to save energy in wireless video streaming. In *MOBICOM*, pages 377–388, 2013.
35. Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. Accelword: energy efficient hotword detection through accelerometer. In *MobiSys*, pages 301–315, 2015.
36. Hauke Holtkamp, Gunther Auer, Samer Bazzi, and Harald Haas. Minimizing base station power consumption. *IEEE Journal on Selected Areas in Communications*, 32(2):297–306, 2014.
37. Li-Ping Tung, Ying-Dar Lin, Yu-Hsien Kuo, Yuan-Cheng Lai, and Krishna M Sivalingam. Reducing power consumption in LTE data scheduling with the constraints of channel condition and QoS. *Computer Networks*, 75:149–159, 2014.
38. Jingyu Zhang, Gan Fang, Minyi Guo, and Chunyi Peng. How video streaming consumes power in 4G LTE networks. In *17th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, 2016.
39. Siripuram Aditya and Sachin Katti. Flexcast: graceful wireless video streaming. In *MOBICOM*, pages 277–288, 2011.
40. Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *SIGCOMM*, pages 325–338, 2015.
41. Kun Wang, Jun Mi, Chenhan Xu, Qingquan Zhu, Lei Shu, and Der Jiunn Deng. Real-time load reduction in multimedia big data for mobile Internet. *ACM Transactions on Multimedia Computing Communications & Applications*, 12(5s):76:1–76:20, 2016.
42. Matthew K Mukerjee, David Naylor, Junchen Jiang, Dongsu Han, Srinivasan Seshan, and Hui Zhang. Practical, real-time centralized control for CDN-based live video delivery. In *SIGCOMM*, pages 311–324, 2015.
43. Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.
44. Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. *ACM SIGCOMM Computer Communication Review*, 43(4):339–350, 2013.