# The Energy Impact on Mobile Clients of TCP Pacing During Video Streaming

MICHAEL WELZL*, Department of Informatics, University of Oslo, Norway and Netflix, Norway
GRENVILLE ARMITAGE, Netflix, Australia
TE-YUAN HUANG, Netflix, USA

Limiting the send rate of video streams via TCP pacing can reduce latency and packet loss for competing traffic without degrading video quality. However, there are concerns that smoothing out packet delivery through TCP pacing may reduce opportunities for wireless interfaces to switch into power-saving sleep modes, and thus increase energy consumption in mobile devices. We present measurements of Streaming Video On Demand (SVOD) playbacks via the Netflix app on millions of Android mobile devices around the globe, and find that pacing indeed costs more power, but its impact is very small: in our large-scale experiment, the median difference between unpaced and paced playback was 24 mW for phones and 21 mW for tablets.

CCS Concepts: • **Networks** → **Transport protocols**; • **Hardware** → **Wireless devices**; **Power and energy**.

Additional Key Words and Phrases: TCP, Pacing, Streaming Video

**ACM Reference Format:**
Michael Welzl, Grenville Armitage, and Te-Yuan Huang. 2026. The Energy Impact on Mobile Clients of TCP Pacing During Video Streaming. *Proc. ACM Netw.* 4, CoNEXT4, Article 63 (December 2026), 16 pages. https://doi.org/10.1145/3856977

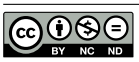
## 1 Introduction

Modern, large-scale streaming services typically utilise DASH or DASH-like protocols for rate-adaptive content delivery over the public internet. Clients retrieve content piecemeal by issuing a succession of GET requests for blocks of bytes representing finite chunks (or segments) of encoded video, spacing those requests out over time at a cadence matching the encoded content's rendering rate. When HTTP relies on a capacity-seeking transport layer like TCP, each GET request results in a burst of data packets on the wire delivering the chunk as fast as the client's network path allows. This is followed by an idle period while waiting for the client to issue its next GET request, giving rise to a familiar on-off pattern on the network.

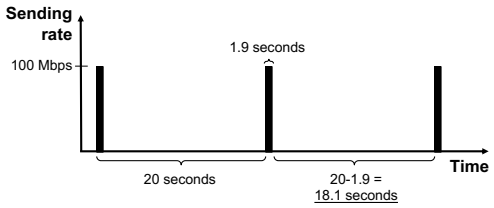
While the long-term average bitrate is proportional to the content's encoded video rate, the instantaneous bitrate during a given HTTP response can spike higher by orders of magnitude [27, 42]. These regular TCP-induced traffic bursts create detrimental side-effects, such as higher queue occupancies and packet loss probabilities, for all traffic sharing on-path bottlenecks with the on-off streaming traffic [50]. Such downsides have driven new interest in the use of server-side TCP pacing to smooth out, and rate limit, such packet-level bursts [2, 19, 20, 41, 51, 53, 54, 60]. Recent work has shown TCP pacing reducing retransmissions by 36% and Round-Trip Times (RTTs) by 14% without detriment to the video stream [42].

*Work done at Netflix.

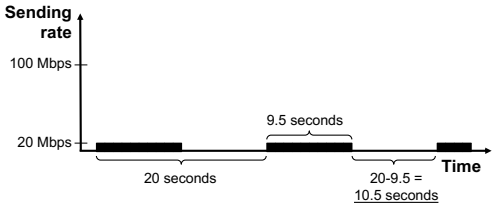
Authors' Contact Information: Michael Welzl, michawe@ifi.uio.no, Department of Informatics, University of Oslo, Norway and Netflix, Norway; Grenville Armitage, Netflix, Melbourne, Australia; Te-Yuan Huang, Netflix, Los Gatos, USA.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2834-5509/2026/12-ART63
https://doi.org/10.1145/3856977




(a) Unpaced: at 100 Mbps, video chunks are transferred within 1.9 seconds, allowing for a pause of 18.1 seconds.




(b) Paced: at 20 Mbps, video chunks are transferred within 9.5 seconds and the pause shrinks down to 10.5 seconds. The rate is significantly smaller, reducing retransmissions and queuing delay.

Fig. 1. Sending 23 MB chunks every 20 seconds to effect a ~3 Mbps video stream.

However, smoothing out the delivery of TCP packets gives rise to a legitimate concern for battery-powered mobile devices[1]. TCP pacing lengthens the period of time the network interface is active per GET, and shortens the idle periods in each on-off cycle. Consequently, mobile devices who leverage idle periods to put their radios into low-power or sleep modes may see increased power consumption, reduced battery life, and degraded user convenience.

Figure 1 illustrates this with a hypothetical scenario where a 3Mbps video stream is retrieved as a GET request for 23Mbytes every 20 seconds, a not-unrealistic cadence for Streaming Video On Demand (SVOD) sessions in steady-state. Figure 1a illustrates how things might be if the path supports 100Mbps—the chunk takes 1.9sec to deliver, leaving the path idle for 18.1sec before the next GET request. Figure 1b illustrates the change if TCP Pacing limits each chunk to 20Mbps—each chunk takes 9.5sec to deliver, and the path goes idle for a shorter 10.5sec.

The power consumption of network hardware may be dwarfed by power consumed by other hardware elements, such as the screen and its backlight [10, 12, 47]. Moreover, the act of video streaming may disable network power management—e.g., the Android developer documentation states that network restrictions will only be activated when the screen is off [5]. Additionally, the radio interface may be prevented from sleeping due to background processes generating their own network traffic during the video streaming app's inter-GET idle periods.

So, we set out to investigate the *real* power impact of applying TCP pacing to SVOD sessions streamed to mobile devices. Our main contributions are:

- The first reported results of the power usage of paced vs. unpaced video streaming from an A/B test with millions of mobile Android devices (Section 2). We find that **pacing's impact on power consumption is very small**. E.g., despite limiting the rate to only 3 Mbps, down from a value that exceeded 20 Mbps for more than half of the monitored playbacks, the median power use for phones only increased from 1.956 W to 1.984 W in our A/B test.
- Hardware-based power measurements of paced vs. unpaced video streaming from mobile Android and iOS devices in a controlled environment (Section 3). These measurements confirm that, **on some devices, power surges coincide with the arrival of data, and their length and/or magnitude is affected by pacing**.

The gap between the paced and unpaced case is slightly larger for playbacks over cellular connections than with Wifi because cellular playbacks require marginally more power than playbacks

[1]For example, the introduction of [45] states that "(..) battery-powered devices perform better with short, bursty exchanges, rather than constant transmission at a fixed rate."

over Wifi; we provide more details on this in Appendix A. The small impact supports the notion that network interface power is swamped by other factors (such as the device screen).

This paper focuses on SVOD sessions, because they are the optimal case for idle-time power savings compared to Live. Power saving becomes more challenging with Live event streaming, as clients request shorter chunks more frequently and idle times will shrink into the range of seconds or sub-seconds. Radios will have fewer opportunities to sleep between chunks, regardless of whether TCP pacing is applied to the 'on' periods.

Combined with the benefits reported in [42] and again confirmed in our A/B test (fewer retransmissions and reduced RTT without degrading video quality), our power consumption results suggest it is reasonable to experiment with TCP pacing for video streaming sessions.

## 2 Large scale measurements

During 20-24th August 2025, we carried out an A/B test involving millions of randomly sampled Android mobile devices (82.3% phones, 17.7% tablets) running the Netflix app[2]. Client devices were split 50:50 between (a) the control group receiving classic, non-paced TCP data transmission, and (b) the test group receiving TCP data transmissions paced to a maximum of 3 Mbps/s. All our results used the Reno congestion control algorithm. We believe that our findings are largely independent of the server's congestion control algorithm: (a) we use pacing to enforce a lower rate than normally calculated by congestion control, (b) the clients are TCP receivers, and we do not expect the server's choice of congestion control to noticeably change receiver-side behavior.

All video content was H.264 or HEVC encoded, and our large sample size ensured a p-value < 0.05 for all considered metrics (i.e. statistically significant differences with a 5% tolerance).

### 2.1 Tracking power consumption

Our streaming app uses the Android API's `BatteryManager`[3] to capture a variety of battery-related metrics at the beginning and the end of every SVOD session. These include the "health" of the battery; the battery type; whether the battery is charging; the present voltage; a "charge counter", representing the present capacity in microampere-hours; and how "full" the battery is, as a percentage. Although battery full-ness is consumer-friendly, it is an imprecise measure here since battery capacities differ across the heterogeneous mix of devices in our experiment. Instead we focus on the time-independent average power $P$ in Watts consumed across SVOD sessions. From the charge counter $C$ and voltage $V$ reported at the start and the end of playback, we calculate the battery's change in energy level in watt-hours, then divide by SVOD playback time $t$ to obtain $P$:

$$P = \frac{C_{start} \cdot V_{start} - C_{end} \cdot V_{end}}{t} \qquad (Watts) \tag{1}$$

### 2.2 Cleaning the data

Since the time gap between video chunks can be in the order of minutes (see, for example, Figures 4 and 5), we limit our analysis to playbacks exceeding 20 minutes. We also filter out cases where:

- the battery was being charged (0.4% of the data),
- the battery was not reported as healthy (2.1% of the data),
- the Android API reported more than one network connectivity type was in use during a playback session (3.4% of the data).

We further cleaned our datasets by filtering out sessions with reported playback errors or obviously wrong or misleading data (such as a battery gaining capacity during a playback without being

[2]No iOS devices because iOS does not expose battery consumption data with sufficient granularity for our study.
[3]https://developer.android.com/reference/android/os/BatteryManager.html

charged). Many of the errors are due to device-specific issues and implementation, and given the diversity of heterogeneous devices in our data set, these issues are inevitable.

### 2.3 Impact of pacing on the network

Consistent with previous findings [42], we observed our paced sessions to have lower retransmission rates and RTT relative to unpaced sessions for equivalent video quality. To confirm the impact of TCP Pacing on idle periods, we define a download time (DLTM) ratio $R_{DLTM}$ as *the time a device spends downloading data divided by the total time of the playback session*, and plot the distribution of $R_{DLTM}$ for our control (unpaced) and test (paced) cohorts in Figure 2. The paced sessions clearly experienced stretched download periods and shortened idle periods. (Other factors such as content bitrates, network congestion, and path capacities are assumed to be equally distributed between cohorts.) For example, 80% of the devices were busy downloading data for less than 15% of the playback time without pacing, compared to more than 20% with pacing. Of course, $R_{DLTM}$ provides no insight into the *absolute* durations of active and idle periods, where the latter directly affects the degree to which radio interfaces can leverage low-power modes.

### 2.4 Impact of pacing on power consumption

Our main finding is captured in Tables 1 and 2: TCP pacing had a very small (albeit non-zero) impact on power consumption between control and test cohorts. With pacing enabled, $R_{DLTM}$ grows by a percentage that is roughly two orders of magnitude larger than the relative growth of power consumption. We also include the battery usage in these tables, noting again that it is a less precise measure than the average power in Watts due to the heterogeneity of client devices (we only know a "filling level" per battery but the total battery capacity is not available to us).

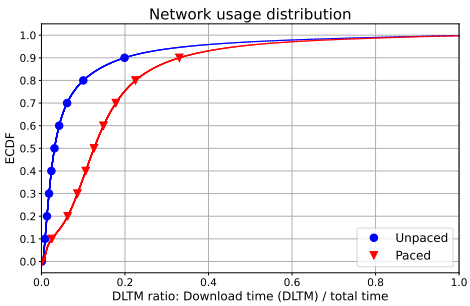



Fig. 2. Distribution of $R_{DLTM}$: Pacing increases the fraction of time spent actively downloading.

We applied two statistical tests from the `SciPy` python library to confirm that the difference in power consumption between unpaced and paced cohorts *is statistically significant*, both for phones and tablets: (a) Kolmogorov-Smirnov (KS), where the KS statistic (i.e., the maximum difference) is 0.0115, and the p-value is 2.06E-121; (b) A two-sample Cramér-von-Mises (CvM) test, where, for the combined dataset, the CvM statistic $T = 61.07$ and p-value is 1.02E-08. With $p << 0.05$ both tests indicate clear statistical significance.

Table 1. **Phones:** Power cost (Watts), Battery cost (% per hour) and $R_{DLTM}$. Percentages in brackets indicate growth from unpaced to paced.

| | Watt | | Battery (% per hour) | | $R_{DLTM}$ | |
|---|---|---|---|---|---|---|
| | Unpaced | Paced | Unpaced | Paced | Unpaced | Paced |
| Mean | 2.552 | 2.607 (2.2%) | 11.921 | 12.046 (1.0%) | 0.085 | 0.166 (95.3%) |
| 10% quantile | 0.983 | 0.997 (1.4%) | 6.083 | 6.232 (2.4%) | 0.009 | 0.026 (188.9%) |
| Median | 1.866 | 1.89 (1.3%) | 10.862 | 10.984 (1.1%) | 0.033 | 0.126 (281.8%) |
| 90% quantile | 3.161 | 3.185 (0.8%) | 18.667 | 18.806 (0.7%) | 0.208 | 0.334 (60.6%) |

Table 2. **Tablets:** Power cost (Watts), Battery cost (% per hour) and $R_{DLTM}$. Percentages in brackets indicate growth from unpaced to paced. Rounded numbers are shown, percentages are calculated with full precision.

| | Watt | | Battery (% per hour) | | $R_{DLTM}$ | |
|---|---|---|---|---|---|---|
| | Unpaced | Paced | Unpaced | Paced | Unpaced | Paced |
| Mean | 3.116 | 3.067 (−1.6%) | 15.217 | 15.332 (0.8%) | 0.066 | 0.16 (142.4%) |
| 10% quantile | 1.051 | 1.067 (1.5%) | 6.944 | 7.073 (1.9%) | 0.008 | 0.021 (162.5%) |
| Median | 2.633 | 2.654 (0.8%) | 13.605 | 13.688 (0.6%) | 0.025 | 0.125 (400.0%) |
| 90% quantile | 5.404 | 5.422 (0.3%) | 25.392 | 25.55 (0.6%) | 0.15 | 0.316 (110.7%) |

Table 3. Power cost (Watts), segmented by network type. Percentages in brackets indicate growth from unpaced to paced.

| | Phones: 3G | | Phones: 4G | | Phones: 5G | |
|---|---|---|---|---|---|---|
| | Unpaced | Paced | Unpaced | Paced | Unpaced | Paced |
| Mean | 2.991 | 2.997 (0.2%) | 2.836 | 2.872 (1.3%) | 2.519 | 2.909 (15.5%) |
| 10% quantile | 1.152 | 1.163 (1.0%) | 1.086 | 1.114 (2.6%) | 0.932 | 0.995 (6.8%) |
| Median | 2.219 | 2.25 (1.4%) | 2.083 | 2.12 (1.8%) | 1.873 | 1.933 (3.2%) |
| 90% quantile | 3.598 | 3.626 (0.8%) | 3.414 | 3.441 (0.8%) | 3.181 | 3.26 (2.5%) |
| | Phones: Wifi | | Tablets: Wifi | | Tablets: 4G | |
| | Unpaced | Paced | Unpaced | Paced | Unpaced | Paced |
| Mean | 2.459 | 2.516 (2.3%) | 3.106 | 3.058 (−1.5%) | 3.515 | 3.397 (−3.4%) |
| 10% quantile | 0.963 | 0.975 (1.2%) | 1.05 | 1.067 (1.6%) | 1.038 | 1.031 (−0.7%) |
| Median | 1.798 | 1.818 (1.1%) | 2.625 | 2.643 (0.7%) | 2.972 | 3.154 (6.1%) |
| 90% quantile | 3.065 | 3.085 (0.7%) | 5.393 | 5.406 (0.2%) | 5.858 | 6.027 (2.9%) |

Unrelated to TCP Pacing, Appendix A captures a separate large-scale data collection exercise we ran to explore the degree to which network connectivity type and battery charge levels influence power consumption for unpaced SVOD sessions. Those results indicate small but real differences linked to network connectivity. Consequently, in Table 3 we explore the power impact of TCP Pacing when segmented by network type. Phone playbacks used WiFi (75.6%), 4G (21.5%), 5G (1.4%), 3G (1.2%), and 2G (0.2%, a number so small that we do not discuss 2G results further). Tablet playbacks used WiFi (97.7%) and 4G (2.1%), with 5G (0.1%), 3G (0.1%) and 2G (0.019%) being inconsequential and not discussed further.

Table 3 does reveal variation linked to network type, but even the most significant differences are small in absolute terms: phones on 5G see a median increase of 60mW (3.2% growth), while tablets on 4G see a median increase of 183mW (6.1%).

## 3 Testbed measurements

To supplement Section 2's revelation of a small power difference between paced and unpaced sessions, we constructed a controlled testbed experiment with specific WiFi-connected clients isolated in a Faraday cage[4] to control for other factors such as device mix, screen brightness, and so forth. Here we report on our testbed findings that support the A/B experiment's observation of

[4]We could not test cellular connectivity due to the lack of an isolated cell tower in our lab's cage.

a real, albeit small, increase in client power consumption when TCP pacing is applied. The change is slightly larger in the testbed due to the more controlled device environment and limited device type mix. TCP pacing does extend the duration of "on" periods and drive up power consumption. But the impact is moderated by the devices being 'chatty' and keeping the network active during the streaming app's "off" (idle) periods, whether pacing or not.

### 3.1 Testbed configuration

The three Android devices and two iOS devices listed in Table 4 were tested one at a time in the testbed shown in Figure 3. We measured each device's energy consumption while playing the same video (encoded at 3.7 Mbps) for one hour without any interruption, no other apps running in the background, and using each device's default configuration. We repeated each test ten times per device, and paced runs were configured for a maximum of 10 Mbps.

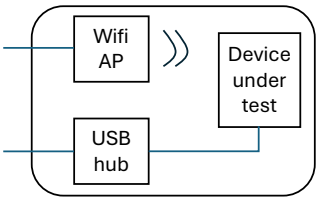



Fig. 3. Testbed: WiFi AP and devices in a Faraday cage. Energy measured by the Acroname USBHub3+ powering each device.

Plugging the devices into an Acroname USBHub3+[5] enabled precise energy consumption measurements while powering each device without charging their batteries. We confirmed expected operation via the Android API, which returned the status `NOT_CHARGING` (rather than `CHARGING` or `DISCHARGING`) while the devices were powered via USB, and separately confirmed that all power consumed during each test came from the Acroname USBHub3+ (none of the batteries lost charge during the experiments).

Table 4. Specific devices used in the Faraday Cage experiments

| Category | OS | Brand | Model | Chipset |
|---|---|---|---|---|
| Smartphone | Android | Samsung | SM-2928B | Qualcomm Snapdragon 8 Gen 3 (SM8650) |
| Smartphone | Android | Google | Pixel 8 | Google Tensor G3 |
| Tablet | Android | Amazon | KFTRWI | Mediatek MT8183 |
| Smartphone | iOS | Apple | iPhone 13 mini | Apple A15 Bionic |
| Tablet | iOS | Apple | iPad 8th gen. | Apple A12 Bionic |

### 3.2 Aggregate energy consumption

We measured energy cumulatively from the beginning of each playback in Watt hours (Wh) and converted to Watts for comparison with Section 2's large-scale measurements. Table 5 shows the median measured power of 10 experiments with and without pacing.

In absolute terms, all of these values are between the 10% quantile and the median in Table 1, both for phones and tablets. We chose these devices due to their relative new-ness and popularity; and since newer devices tend to be more power efficient, it is not unreasonable to see them consume less absolute power than the more diverse mix of devices in Table 1.

The impact of pacing on the median power consumption for phones is 170 mW (Samsung), 112 mW (Google), 101 mW (iPhone); directionally consistent with, albeit larger than, the impact we saw in the wide-area A/B test. These differences show how these two data sets complement each other. While the testbed data allows us to dive deeper under controlled conditions for an individual

[5]https://acroname.com/reference/devices/usbhub3p/index.html

Table 5. Median power use and Inter Quartile Range (IQR) of 10 tests per device in the testbed.

| | Unpaced | | Paced | | | Unpaced | | Paced | |
|---|---|---|---|---|---|---|---|---|---|
| **Phones** | Watt | IQR | Watt | IQR | **Tablets** | Watt | IQR | Watt | IQR |
| Samsung | 1.18 | 0.019 | 1.35 | 0.103 | Amazon | 1.396 | 0.124 | 1.523 | 0.126 |
| Google | 1.624 | 0.061 | 1.736 | 0.029 | iPad | 1.596 | 0.39 | 1.624 | 0.11 |
| iPhone | 1.337 | 0.159 | 1.438 | 0.202 | | | | | |

device, the A/B test data gives us an idea of how power consumption differs in the field between paced and unpaced sessions, while accounting for device mix, playback duration, user interactions, network conditions, background tasks, and other practical factors. Together, these two datasets paint a more complete picture.

For tablets, the differences between the medians are 127 mW (Amazon) and 28 mW (iPad). In all cases, despite the experiments being run under controlled conditions, the spread between the similar 10 experiments is quite large; this is because of outliers. In the following, we will examine the time series to understand the underlying behavior.

### 3.3 Closer inspection of the relationship between Power and Traffic

Although the testbed was isolated, there are always some anomalous events on any given run. We illustrate this for the Google phone with Figure 4a (unpaced) and Figure 4b (paced) showing one line for each of the ten runs of power time series measurements, with per-minute granularity. A single example of the video streaming network traffic associated with each test is shown in Figure 4c and Figure 4d respectively.

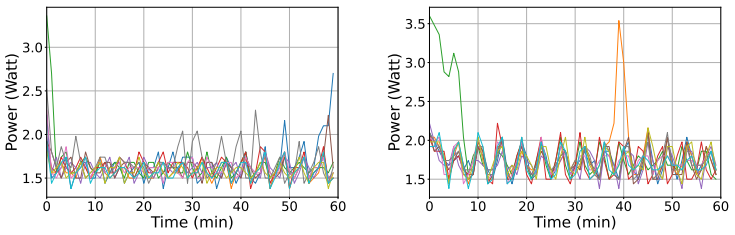
Despite being visually cluttered, we can see a couple of things: (a) The video streaming app issues GETs approximately every 4 minutes (Figure 4c), (b) Many runs have outliers (occasional peaks at different times in each test, rather than systematic behavior), and (c) Pacing does consistently add to energy consumption during the 'on' periods as video chunks are being downloaded (Figure 4b's regular peaks coincide with paced network traffic in Figure 4d).

Outliers were seen with almost all the devices, due to processes unrelated to the video streaming app (e.g. periodic operating systen garbage collection, and so forth). So the power time series plots for remaining devices in Figure 5 use a clearer style focusing on the median and width of the distribution over all ten tests in each case.

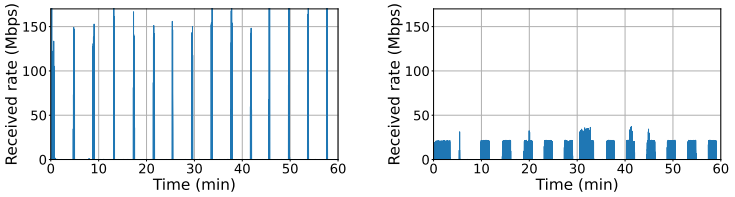
The Samsung phone (Figures 5a, 5b, 5e, and 5f) show similar general behavior to the Google phone, both for unpaced and paced cases: the streaming app is issuing GETs every few minutes, the base power is similar between unpaced and paced, and larger power surges coincide with the longer 'on' period of traffic arrival when pacing (Figure 5f).

The Amazon tablet and iPhone are similar in that their streaming app is issuing GET requests (Figures 5g and 5m) far more frequently than on the Google and Samsung phones. With power readings averaged over one minute intervals, the impact of pacing appears as a level shift rather than distinct power surges (Figures 5c vs 5d for the Amazon tablet, Figures 5i vs 5j for the iPhone).

The iPad is issuing GETs more frequently than the Google and Samsung phones, and less frequently than the iPhone. As with all the other devices there's a clear difference between the resulting unpaced and paced traffic (Figures 5o and 5p). Even allowing for some power consumption outliers, paced sessions exhibit a very small increase in power consumption over unpaced sessions (Figures 5k vs 5l), much smaller than seen with the Samsung and Google phones.




(a) Google phone, unpaced: power. (b) Google phone, paced: power.




(c) Google phone, unpaced: rate. (d) Google phone, paced: rate.

Fig. 4. Time series of Google phone power (10 experiments, one per line) and reception rate (1 experiment). Power spikes in a few experiments are outliers rather than systematic behavior. Reception rate reflects the streaming app issuing GETs approximately every four minutes.

### 3.4 Background chatter

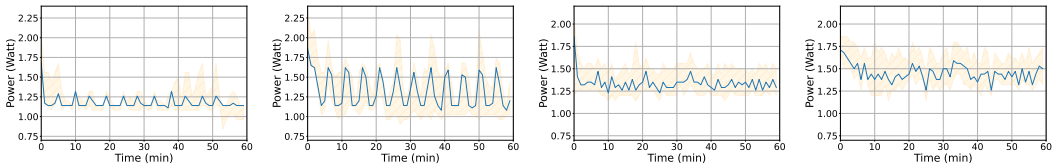
Part of the reason pacing drives a relatively small increase in Table 5's power consumption is that the devices were generally 'chatty' over WiFi. Consequently it becomes less likely the device's network interface goes into a significant sleep mode during the nominally idle periods after responses to the video streaming app's GET requests. Examples of additional observed traffic include:

- TCP keep-alives and associated acknowledgments (ACKs) on the streaming connection.
- Traffic from/to `1e100.net` (internal Google servers), including TCP keep-alives and ACKs.
- Every 15 seconds, the phone sends Address Resolution Protocol (ARP) requests for the gateway's IP address. Less frequently, the gateway ARPs for the phone's IP address.

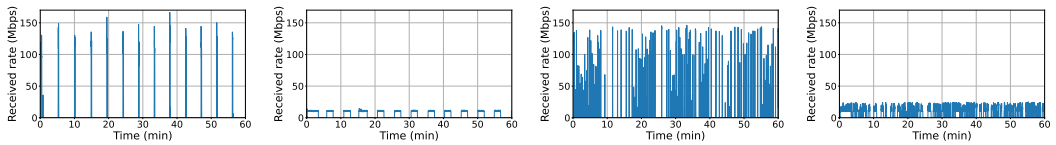
This chatter exists at the OS level, not tied to any specific streaming service. The chatter increases energy consumption in the unpaced case and closes the power gap between paced and unpaced sessions. Hence, while we do not attribute the small difference between the paced and unpaced case to chatter alone (devices typically do not attempt power-saving network behavior when the screen is on [5]), there may be an opportunity for mobile operating systems to better coordinate traffic across the network stack as part of a larger effort to reduce overall power consumption.

## 4 Related work

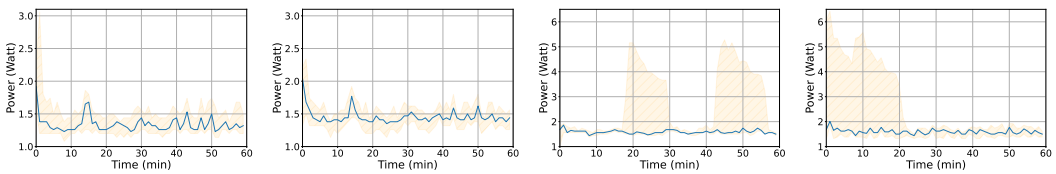
**Mobile device power consumption:** The energy use of mobiles, specifically Smartphones, is a topic of many papers; an early one is by Carroll and Heiser [10] from 2010, showing a break-down per component. Unsurprisingly, this and all of the following studies find that the display backlight is a dominant factor (albeit depending on brightness). The power requirement of Wifi is reported to be relatively small in [10], where, in suspended state, GSM is shown to drain the largest amount of power. In 2014 and 2015, Chen et al. [12, 13] echo this relationship between components, and Wifi appears negligible while the cellular power drain even beats the display in [23].



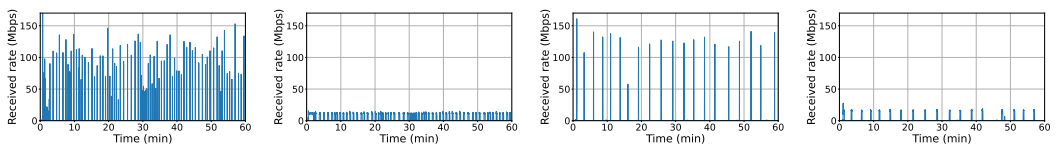

(a) Samsung phone, unpaced: power. (b) Samsung phone, paced: power. (c) Amazon tablet, unpaced: power. (d) Amazon tablet, paced: power.




(e) Samsung phone, unpaced: rate. (f) Samsung phone, paced: rate. (g) Amazon tablet, unpaced: rate. (h) Amazon tablet, paced: rate.




(i) iPhone 13 mini, unpaced: power. (j) iPhone 13 mini, paced: power. (k) iPad, unpaced: power. (l) iPad, paced: power.




(m) iPhone 13 mini, unpaced: rate. (n) iPhone 13 mini, paced: rate. (o) iPad, unpaced: rate. (p) iPad, paced: rate.

Fig. 5. Time series of power (10 experiments) and reception rate (1 experiment) with the Samsung tablet, Amazon tablet, iPhone and iPad. For power, the line shows the per-minute median and the shaded area indicates the minimum / maximum range from all 10 experiments.

Some papers focus on *how* to best measure energy [17, 30, 38, 47, 48], the various ways the phone is used (different apps [32, 57], the browser [28, 49], "dark mode" [18], AI inference [4, 16]), or specific connectivity (5G [37, 56, 59], Wifi 6 [33], 802.11ad [3]). Even the user behavior in response to a draining battery—"Low Battery Anxiety" (LBA)—has been studied [43]. In general, wherever power is broken down by component, Wifi is seen to play a very small role relative to other parts of the device while cellular plays a larger one – corroborating our findings in Appendix A.1.

**Streaming video power consumption:** Some works consider the overall CO2 contribution, taking a holistic view that involves the entire streaming ecosystem from servers via caches to end users [1, 7, 24, 25, 31, 34, 46]. Closer to our investigation, others focus on, and and/or strive to improve, the power consumption at clients [11, 34–36, 44], but we are not aware of any prior study that investigates the influence of the network traffic pattern. Also, to the best of our knowledge,

Herglotz et al. [26] is the only attempt to obtain client-side power use from a large dataset (447k streaming events from YouTube), but power is estimated using a model instead of obtaining it directly, and the dataset is significantly smaller than our datasets in Section 2 and Appendix A.

**TCP power consumption:** Independent of streaming video, the power consumption of TCP has also been investigated. Bolla et al. [8] present a very detailed analysis using an off-the shelf PC, finding that the power impact of even a single TCP connection is significant. It is much smaller in server systems [6, 14], where the impact grows linearly with the number of connections (requests arriving at a web server) [21]. The energy of TCP over Wifi is considered in [52], suggesting that bulk data transmission would cost slightly less energy—but this is done with a model rather than device measurements, and considering energy, not power—conflating the duration of data transfers with their power consumption. For a fair comparison, the idle device energy during the remaining time would also have to be considered.

Some works suggest adapting the traffic pattern or TCP behavior in order to reduce the power consumption of TCP data transfers [6, 15, 55]. Also, reducing the RTT by accessing the edge rather than a remote cloud is found the help [9], and so is proxying with compression [58]. Related to our discussion in Section 3, Pignataro et al. [40] state that the "chattiness" of protocols—e.g. regular heartbeat messages–possibly prevents devices from entering a sleep state, and mobile browsers appear to be particularly chatty [39]. Gunaratne et al. [22] suggest to use a proxy that removes chatter as a way to improve sleep times of a PC.

To summarize, some prior research supports a few of our findings—such as the small influence of Wifi, the larger influence of cellular connectivity, and the high likelihood that chatter prevents components from entering a sleep state—but there is no other client power measurement study at a comparable scale available in the literature. Also, we are the first to investigate the power influence of pacing at the client side.

## 5 Conclusion

This paper aims to understand the implications of TCP pacing on energy consumption for mobile devices. We started with analyzing an A/B test, where SVOD traffic was paced with a maximum rate of 3 Mbps—low enough to cause a significant difference in how much time is spent downloading data. The resulting power difference that Android mobiles reported was very small—generally in the order of tens of milliWatts, which is two orders of magnitude smaller than the overall absolute power used for SVOD by a client device.

Testbed experiments where power measurements were polled from a USB hub (a) confirmed that pacing increases the active periods and decreases idle periods of network activity, (b) illustrated that correlations between pacing's more sustained traffic arrivals and power consumption peaks are common, (c) showed that the behavior is quite diverse across different devices, and (d) confirmed that the power consumption difference between unpaced and paced traffic is small.

TCP pacing is an effective approach to minimize packet loss and reduce queuing delays along shared network paths, and it is becoming more widely used—e.g., [54] describes current deployments in Linux, FreeBSD, Apple OSes and within some implementations of QUIC BBR. We regard pacing as an important element along the way towards a lower-latency Internet for everyone—but the concern of increased energy cost in mobile clients can slow down its deployment. By clarifying that median effect of pacing on energy is small under the tested workloads and current mobile environments, we hope that our work encourages wider deployments of pacing.

We see a number of valuable next steps that should be taken in future work:

- In routers and switches, the general impact of the rate of traffic has been shown to be unsubstantial [29], but the server-side power influence is unknown.

- We have evaluated two specific pacing configurations: 3 Mbps as an upper limit in the large-scale experiment, and 10 Mbps in the testbed. It would be useful to carry out a sensitivity analysis over a larger variety of pacing rates.
- We have focused on realistic usage of our app on mobile devices without carrying out a deeper analysis of the various hardware elements involved. This includes the screen, which typically consumes the most power, being always on, but without considerations of screen brightness, for example.

## Ethics

This paper discusses data from Internet users executing SVOD playbacks from a large streaming service. No personally identifiable information was used; the datasets will not be publicly released. Users have given their consent to participate in A/B experiments. The A/B experiment was carefully designed to avoid harm to users, and succeeded at that: the small power cost was offset by a significant reduction of the RTT and packet loss, and video QoE was not affected.

## References


[1] Samira Afzal, Narges Mehran, Zoha Azimi Ourimi, Farzad Tashtarian, Hadi Amirpour, Radu Prodan, and Christian Timmerer. 2024. A Survey on Energy Consumption and Environmental Impact of Video Streaming. arXiv:2401.09854 [cs.MM] https://arxiv.org/abs/2401.09854

[2] A. Aggarwal, S. Savage, and T. Anderson. 2000. Understanding the performance of TCP pacing. In *IEEE INFOCOM 2000*, Vol. 3. 1157–1165 vol.3. doi:10.1109/INFCOM.2000.832483

[3] Shivang Aggarwal, Moinak Ghoshal, Piyali Banerjee, Dimitrios Koutsonikolas, and Joerg Widmer. 2021. 802.11ad in Smartphones: Energy Efficiency, Spatial Reuse, and Impact on Applications. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. 1–10. doi:10.1109/INFOCOM42981.2021.9488763

[4] Mario Almeida, Stefanos Laskaridis, Abhinav Mehrotra, Lukasz Dudziak, Ilias Leontiadis, and Nicholas D. Lane. 2021. Smart at what cost? characterising mobile deep neural networks in the wild. In *Proceedings of the 21st ACM Internet Measurement Conference* (Virtual Event) *(IMC '21)*. Association for Computing Machinery, New York, NY, USA, 658–672. doi:10.1145/3487552.3487863

[5] Android Open Source Project. 2026. *Platform Power Management.* https://source.android.com/docs/core/power/platform_mgmt Accessed: May 18, 2026.

[6] Serhat Arslan, Sundararajan Renganathan, and Bruce Spang. 2023. Green With Envy: Unfair Congestion Control Algorithms Can Be More Energy Efficient. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks* (Cambridge, MA, USA) *(HotNets '23)*. Association for Computing Machinery, New York, NY, USA, 220–228. doi:10.1145/3626111.3628200

[7] Youssef Badra and Razvan Stanica. 2025. Experimental Analysis of Energy Consumption in Video Streaming Services. In *2025 IEEE 26th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE Computer Society, Los Alamitos, CA, USA, 11–20. doi:10.1109/WoWMoM65615.2025.00016

[8] Raffaele Bolla, Roberto Bruschi, Olga M. Jaramillo Ortiz, and Paolo Lago. 2015. An Experimental Evaluation of the TCP Energy Consumption. *IEEE Journal on Selected Areas in Communications* 33, 12 (2015), 2761–2773. doi:10.1109/JSAC.2015.2482038

[9] Chiara Caiazza, Valerio Luconi, and Alessio Vecchio. 2022. Saving Energy on Smartphones through Edge Computing: An Experimental Evaluation. In *Proceedings of the ACM SIGCOMM Workshop on Networked Sensing Systems for a Sustainable Society* (Amsterdam, Netherlands) *(NET4us '22)*. Association for Computing Machinery, New York, NY, USA, 20–25. doi:10.1145/3538393.3544935

[10] Aaron Carroll and Gernot Heiser. 2010. An Analysis of Power Consumption in a Smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference* (Boston, MA) *(USENIXATC'10)*. USENIX Association, USA, 21.

[11] Xianda Chen and Guohong Cao. 2023. Energy-Efficient 360-Degree Video Streaming on Multicore-Based Mobile Devices. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 1–10. doi:10.1109/INFOCOM53939.2023.10228863

[12] Xiang Chen, Yiran Chen, Mian Dong, and Charlie Zhang. 2014. Demystifying energy usage in smartphones. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–5. doi:10.1109/DAC.2014.6881397

[13] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y. Charlie Hu, Maruti Gupta, and Rath Vannithamby. 2015. Smartphone Energy Drain in the Wild: Analysis and Implications. In *Proceedings of the 2015 ACM SIGMETRICS International*

*Conference on Measurement and Modeling of Computer Systems* (Portland, Oregon, USA) *(SIGMETRICS '15)*. Association for Computing Machinery, New York, NY, USA, 151–164. doi:10.1145/2745844.2745875

[14] Kristjon Ciko, Michael Welzl, and Peyman Teymoori. 2023. Going Dark: A Software "Light Switch" for Internet Servers. In *2023 IEEE 29th International Symposium on Local and Metropolitan Area Networks (LANMAN)*. 1–6. doi:10.1109/LANMAN58293.2023.10189419

[15] Alexander Clemm, Dirk Kutscher, Michael Welzl, Cedric Westphal, Noa Zilberman, and Simone Ferlin-Reiter. 2025. Greening Networking: Toward a Net Zero Internet (Dagstuhl Seminar 24402). *Dagstuhl Reports* 14, 9 (2025), 167–192. doi:10.4230/DagRep.14.9.167

[16] Abhishek Dandekar, Ashrafur Rahman, and Julius Schulz-Zander. 2025. E2E energy monitoring for AI inference in mobile networks. In *Proceedings of the 1st Workshop on Next-Generation Network Observability* (Coimbra, Portugal) *(NGNO '25)*. Association for Computing Machinery, New York, NY, USA, 37–43. doi:10.1145/3748496.3748995

[17] Abhishek Dandekar, Johannes Wegener, Ashrafur Rahman, and Julius Schulz-Zander. 2024. Towards Application Level Energy Monitoring for Green 6G Networks. In *Proceedings of the 4th ACM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases* (Los Angeles, CA, USA) *(5G-MeMU '24)*. Association for Computing Machinery, New York, NY, USA, 8–13. doi:10.1145/3694810.3700158

[18] Pranab Dash and Y. Charlie Hu. 2021. How much battery does dark mode save? an accurate OLED display power profiler for modern smartphones. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 323–335. doi:10.1145/3458864.3467682

[19] Nandita Dukkipati, Yuchung Cheng, and Amin Vahdat. 2018. Research Impacting the Practice of Congestion Control. *SIGCOMM Comput. Commun. Rev.* 46, 3, Article 14 (July 2018), 2 pages. doi:10.1145/3243157.3243171

[20] David Fridovich-Keil, Nathan Hanford, Margaret P. Chapman, Claire J. Tomlin, Matthew K. Farrens, and Dipak Ghosal. 2017. A model predictive control approach to flow pacing for TCP. In *2017 55th Annual Allerton Conf. on Communication, Control, and Computing (Allerton)*. doi:10.1109/ALLERTON.2017.8262845

[21] Loïc Guibert, Sébastien Reynaud, Olivier Weppe, Kristjon Ciko, Michael Welzl, and Sébastien Rumley. 2025. TRAFFIC: Testbed foR Assessing energy eFFiciency In throughput Computing. In *Proceedings of the 22nd ACM International Conference on Computing Frontiers: Workshops and Special Sessions (CF '25 Companion)*. Association for Computing Machinery, New York, NY, USA, 99–107. doi:10.1145/3706594.3727961

[22] Chamara Gunaratne, Ken Christensen, and Bruce Nordman. 2005. Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. *International Journal of Network Management* 15, 5 (2005), 297–310. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.565 doi:10.1002/nem.565

[23] Agrim Gupta, Adel Heidari, Avyakta Kalipattapu, Ish Kumar Jain, and Dinesh Bharadia. 2024. 3 W's of smartphone power consumption: Who, Where and How much is draining my battery?. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 2248–2250. doi:10.1145/3636534.3695905

[24] Christian Herglotz, Matthias Kränzler, Robert Schober, and André Kaup. 2023. Sweet Streams Are Made of This: The System Engineer's View on Energy Efficiency in Video Communications [Feature]. *IEEE Circuits and Systems Magazine* 23, 1 (2023), 57–77. doi:10.1109/MCAS.2023.3234739

[25] Christian Herglotz, Werner Robitza, Matthias Kränzler, Andre Kaup, and Alexander Raake. 2022. Modeling of Energy Consumption and Streaming Video QoE using a Crowdsourcing Dataset. In *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. doi:10.1109/QoMEX55416.2022.9900886

[26] Christian Herglotz, Werner Robitza, Matthias Kränzler, Andre Kaup, and Alexander Raake. 2022. Modeling of Energy Consumption and Streaming Video QoE using a Crowdsourcing Dataset. In *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. doi:10.1109/QoMEX55416.2022.9900886

[27] Te-Yuan Huang. 2025. Adaptive Bitrate & Congestion Control: Together, they make the Internet a friendlier place. Presentation to the Internet Congestion Control Research Group (ICCRG). https://datatracker.ietf.org/meeting/123/materials/slides-123-iccrg-adaptive-bitrate-congestion-control-together-they-make-the-internet-a-friendlier-place-01 Presented at IETF 123.

[28] Stefan Huber, Lukas Demetz, and Michael Felderer. 2022. A comparative study on the energy consumption of Progressive Web Apps. *Information Systems* 108 (2022), 102017. doi:10.1016/j.is.2022.102017

[29] Romain Jacob, Lukas Röllin, Jackie Lim, Jonathan Chung, Maurice Béhanzin, Weiran Wang, Andreas Hunziker, Theodor Moroianu, Seyedali Tabaeiaghdaei, Adrian Perrig, and Laurent Vanbever. 2025. Fantastic Joules and Where to Find Them. Modeling and Optimizing Router Energy Demand. In *25th ACM Internet Measurement Conference (IMC 2025)*. doi:10.1145/3730567.3732920

[30] Abhilash Jindal and Y. Charlie Hu. 2021. Experience: developing a usable battery drain testing and diagnostic tool for the mobile industry. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (New Orleans, Louisiana) *(MobiCom '21)*. Association for Computing Machinery, New York, NY, USA,

804–815. doi:10.1145/3447993.3483269

[31] Ryan Hardesty Lewis and Alex Conway. 2025. Streaming the Future of Sustainability. *SIGENERGY Energy Inform. Rev.* 4, 5 (April 2025), 166–171. doi:10.1145/3727200.3727225

[32] Tong Li, Tong Xia, Huandong Wang, Zhen Tu, Sasu Tarkoma, Zhu Han, and Pan Hui. 2022. Smartphone App Usage Analysis: Datasets, Methods, and Applications. *IEEE Communications Surveys & Tutorials* 24, 2 (2022), 937–966. doi:10.1109/COMST.2022.3163176

[33] Ruofeng Liu and Nakjung Choi. 2023. A First Look at Wi-Fi 6 in Action: Throughput, Latency, Energy Efficiency, and Security. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 1, Article 25 (March 2023), 25 pages. doi:10.1145/3579451

[34] Daniele Lorenzi, Minh Nguyen, Farzad Tashtarian, and Christian Timmerer. 2024. E-WISH: An Energy-aware ABR Algorithm For Green HTTP Adaptive Video Streaming. In *Proceedings of the 3rd Mile-High Video Conference* (Denver, CO, USA) *(MHV '24)*. Association for Computing Machinery, New York, NY, USA, 28–33. doi:10.1145/3638036.3640802

[35] Daniele Lorenzi, Farzad Tashtarian, and Christian Timmerer. 2025. Adaptive Quality and Energy Enhancement in Video Streaming with RecABR. In *Proceedings of the 4th Mile-High Video Conference* (Denver, CO, USA) *(MHV '25)*. Association for Computing Machinery, New York, NY, USA, 67–68. doi:10.1145/3715675.3715804

[36] Abhijit Mondal, Basabdatta Palit, Somesh Khandelia, Nibir Pal, Jay Jayatheerthan, Krishna Paul, Niloy Ganguly, and Sandip Chakraborty. 2020. EnDASH - A Mobility Adapted Energy Efficient ABR Video Streaming for Cellular Networks. In *2020 IFIP Networking Conference (Networking)*. 127–135.

[37] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, Feng Qian, and Zhi-Li Zhang. 2021. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA) *(SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 610–625. doi:10.1145/3452296.3472923

[38] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang. 2012. Where is the energy spent inside my app? fine grained energy accounting on smartphones with Eprof. In *Proceedings of the 7th ACM European Conference on Computer Systems* (Bern, Switzerland) *(EuroSys '12)*. Association for Computing Machinery, New York, NY, USA, 29–42. doi:10.1145/2168836.2168841

[39] John Pegioudis, Emmanouil Papadogiannakis, Nicolas Kourtellis, Evangelos P. Markatos, and Panagiotis Papadopoulos. 2023. Not only E.T. Phones Home: Analysing the Native User Tracking of Mobile Browsers. In *Proceedings of the 2023 ACM on Internet Measurement Conference* (Montreal QC, Canada) *(IMC '23)*. Association for Computing Machinery, New York, NY, USA, 181–187. doi:10.1145/3618257.3624842

[40] Carlos Pignataro, Ali Rezaki, Jari Arkko, Alexander Clemm, Hesham ElBakoury, and Shailesh Prabhu. 2025. *Sustainability Considerations for Networking Protocols and Applications*. Internet-Draft draft-pignataro-enviro-sustainability-consid-03. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-pignataro-enviro-sustainability-consid/03/ Work in Progress.

[41] Andreas Schmidt, Stefan Reif, Pablo Gil Pereira, Timo Honig, Thorsten Herfet, and Wolfgang Schroder-Preikschat. 2019. Cross-Layer Pacing for Predictably Low Latency. In *IEEE INFOCOM 2019 Workshops (WKSHPS)*. doi:10.1109/INFOCOMWKSHPS47286.2019.9093750

[42] Bruce Spang, Shravya Kunamalla, Renata Teixeira, Te-Yuan Huang, Grenville Armitage, Ramesh Johari, and Nick McKeown. 2023. Sammy: smoothing video traffic to be a friendly internet neighbor. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 754–768. doi:10.1145/3603269.3604839

[43] Guoming Tang, Kui Wu, Yangjing Wu, Huan Wang, and Guangwu Qian. 2022. Modeling and Alleviating Low-Battery Anxiety for Mobile Users in Video Streaming Services. *IEEE Internet of Things Journal* 9, 7 (2022), 5065–5079. doi:10.1109/JIOT.2021.3107728

[44] Farzad Tashtarian, Daniele Lorenzi, Hadi Amirpour, Samira Afzal, and Christian Timmerer. 2024. COCONUT: Content Consumption Energy Measurement Dataset for Adaptive Video Streaming. In *Proceedings of the 15th ACM Multimedia Systems Conference* (Bari, Italy) *(MMSys '24)*. Association for Computing Machinery, New York, NY, USA, 346–352. doi:10.1145/3625468.3652179

[45] Martin Thomson, Christian Huitema, Kazuho Oku, Matt Joras, and Marcus Ihlar. 2026. *Standard Communication with Network Elements (SCONE) Protocol*. Internet-Draft draft-ietf-scone-protocol-09. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-scone-protocol/09/ Work in Progress.

[46] Ramona Trestian, Arghir-Nicolae Moldovan, Olga Ormond, and Gabriel-Miro Muntean. 2012. Energy consumption analysis of video streaming to Android mobile devices. In *2012 IEEE Network Operations and Management Symposium*. 444–452. doi:10.1109/NOMS.2012.6211929

[47] Matteo Varvello, Kleomenis Katevas, Mihai Plesa, Hamed Haddadi, Fabian Bustamante, and Ben Livshits. 2022. BatteryLab: A Collaborative Platform for Power Monitoring. In *Passive and Active Measurement*, Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser (Eds.). Springer International Publishing, Cham, 97–121.

[48] Matteo Varvello, Kleomenis Katevas, Mihai Plesa, Hamed Haddadi, and Benjamin Livshits. 2019. BatteryLab, A Distributed Power Monitoring Platform For Mobile Devices. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (Princeton, NJ, USA) *(HotNets '19)*. Association for Computing Machinery, New York, NY, USA, 101–108. doi:10.1145/3365609.3365852

[49] Matteo Varvello and Benjamin Livshits. 2021. Shedding (Some) Light on Mobile Browsers Energy Consumption. In *5th Network Traffic Measurement and Analysis Conference, TMA 2021, Virtual Event, September 14-15, 2021*, Vaibhav Bajpai, Hamed Haddadi, and Oliver Hohlfeld (Eds.). IFIP. http://dl.ifip.org/db/conf/tma/tma2021/tma2021-paper18.pdf

[50] Ranysha Ware, Matthew K. Mukerjee, Srinivasan Seshan, and Justine Sherry. 2019. Beyond Jain's Fairness Index: Setting the Bar For The Deployment of Congestion Control Algorithms. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (Princeton, NJ, USA) *(HotNets '19)*. Association for Computing Machinery, New York, NY, USA, 17–24. doi:10.1145/3365609.3365855

[51] David Wei, Pei Cao, Steven Low, and Caltech EAS. 2006. TCP pacing revisited. In *Proceedings of IEEE INFOCOM*, Vol. 2. 3.

[52] Michael Welzl. 2022. Not a Trade-Off: On the Wi-Fi Energy Efficiency of Effective Internet Congestion Control. In *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*. 1–4. doi:10.23919/WONS54113.2022.9764413

[53] Michael Welzl. 2025. TCP Pacing in the Linux Kernel. In *2025 International Conference on Computing, Networking and Communications (ICNC)*. 133–139. doi:10.1109/ICNC64010.2025.10994011

[54] Michael Welzl, Wesley Eddy, Vidhi Goel, and Michael Tüxen. 2026. *Pacing in Transport Protocols*. Internet-Draft draft-irtf-iccrg-pacing-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-irtf-iccrg-pacing/01/ Work in Progress.

[55] Michael Welzl, Emile Stephan, Eve Schooler, Sébastien Rumley, Ali Rezaki, Jukka Manner, Carlos Pignataro, Marisol Palmero, Jan Lindblad, Suresh Krishnan, Ari Keränen, Hesham ElBakoury, Luis M. Contreras, Alexander Clemm, and Jari Arkko. 2025. *Architectural Considerations for Environmentally Sustainable Internet Technology*. Internet-Draft draft-various-eimpact-arch-considerations-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-various-eimpact-arch-considerations/01/ Work in Progress.

[56] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 479–494. doi:10.1145/3387514.3405882

[57] Ming Yan, Chien Aun Chan, André F. Gygax, Jinyao Yan, Leith Campbell, Ampalavanapillai Nirmalathas, and Christopher Leckie. 2019. Modeling the Total Energy Consumption of Mobile Network Services and Applications. *Energies* 12, 1 (2019). doi:10.3390/en12010184

[58] Bin Yu, Le Wang, and Jukka Manner. 2010. Energy-Efficient Web Access on Mobile Devices. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications and Int'l Conference on Cyber, Physical and Social Computing*. 442–447. doi:10.1109/GreenCom-CPSCom.2010.128

[59] Xinjie Yuan, Mingzhou Wu, Zhi Wang, Yifei Zhu, Ming Ma, Junjian Guo, Zhi-Li Zhang, and Wenwu Zhu. 2022. Understanding 5G performance for real-world services: a content provider's perspective. In *Proceedings of the ACM SIGCOMM 2022 Conference* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 101–113. doi:10.1145/3544216.3544219

[60] Lixia Zhang, Scott Shenker, and Daivd D. Clark. 1991. Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-way Traffic. In *Proceedings of the Conf. on Communications Architecture & Protocols* (Zurich, Switzerland) *(SIGCOMM '91)*. 133–147. doi:10.1145/115992.116006

## A The influence of network connectivity or battery charge level on power use

Here, we analyze battery data reported by the Android API, using a large dataset of tens of millions of samples from randomly chosen SVOD playbacks from 2025. We filter the data in the same way as in Section 2. We examine the type of network connectivity, the type of battery and how much the battery is charged; these results have informed our analysis in Section 2, since factors that magnify the power use have the potential to also increase the gap between the paced and unpaced case.

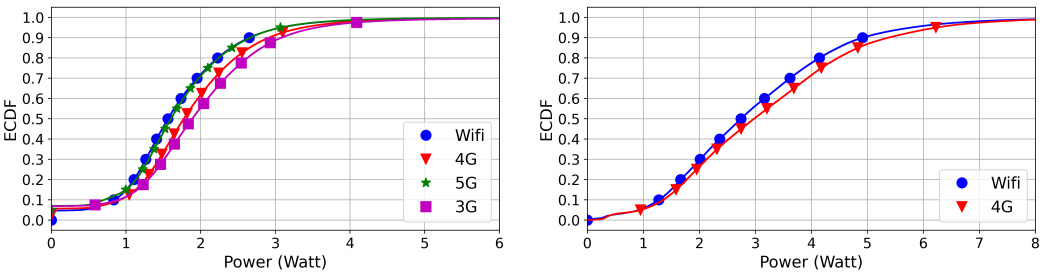
### A.1 Network Connectivity Type

In our larger dataset, the break-down of network usage is:

**Phones:** Wifi 77.1%, 4G 19.6%, 5G 2.3%, 3G 0.8%, 2G 0.2%
**Tablets:** Wifi 97.8%, 4G 1.7%, 5G 0.4%, 3G 0.04%, 2G 0.01%

Figures 6a and 6b show the power distribution per connectivity type for phones and tablets, respectively. We show the same connectivity types as in Table 3.

There is a significant level of noise in these data: e.g., for Wifi, the interquartile range is 57.1% of the median for phones and 73.7% of the median for tablets. This is expected, as battery usage is influenced by screen brightness, background processes and other settings; devices can differ greatly, and so can battery quality (and "age", in terms of usage time). These playbacks cover a wide range of movies, played for different durations.




(a) Phones (b) Tablets

Fig. 6. SVOD power consumption per connection type.

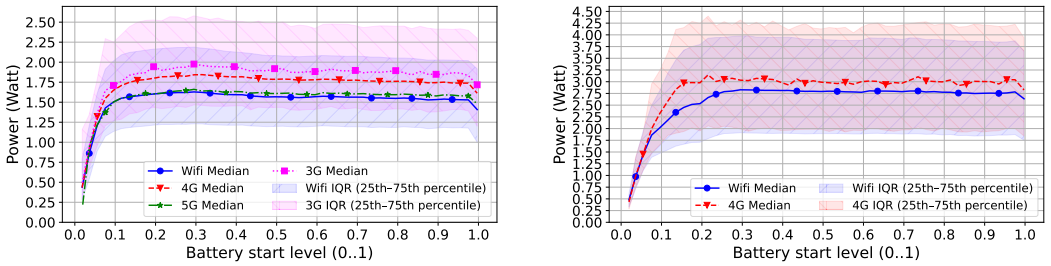
The gap between different types of network usage is quite small, both for phones and tablets; we attribute this to the fact that the largest share of power consumption is in fact caused by other factors (video decoding, screen, backlight, etc.). With phones, in line with earlier findings [59], Wifi consumes the least amount of power. Contradicting the findings in [59] however, the phone power consumption of 5G is smaller than the other cellular cases. We lack the necessary information on the cellular connectivity (e.g., non-standalone vs. standalone mode use of 5G) that would allow us to really understand what may have caused this. We attribute it to the more advanced power saving capabilities of 5G, which appear to be better than 3G or 4G at reducing the power use during periods of quiescence.

It is clear that the connectivity type plays a role, albeit not a huge one. We therefore also analyzed our A/B test data accordingly (see Section 2), and found that 5G for phones played out differently there—the 5G gap between paced and unpaced data transfer was the largest.

## A.2 Battery Level

We use the term "battery level" to refer to the charging level from 0 to 1, where 0 is empty and 1 is "fully charged". We consider the battery level at the beginning of a playback.

Figures 7a and 7b show the SVOD power consumption of phones and tablets as a function of the initial battery level. The distributions were calculated by binning x-axis values (battery start level) into 50 bins; lines represent the median, and the IQR is shown for the two cases that are furthest apart. Generally, all distributions follow a similar trend. The stronger fluctuations in case of 4G for tablets are likely due to the smaller amount of data in this category.




(a) Phones (b) Tablets

Fig. 7. SVOD power consumption depending on the initial battery level.

We make the following observations:

- In the approximate battery level range from 0.2 to 0.95, power usage is quite flat.
- All devices exhibit a roughly concave growth up to an approximate filling level of approximately 0.2, where the curve plateaus. This is likely caused by power saving modes, e.g., reducing screen brightness (either automated or manually) when the battery is low. In case of WiFi and tablets, there is a dip in the curve at 0.2, i.e. the concave increase stops there, only to increase with concave growth until approximately 0.25. Since this is supported by a large data volume, it indicates a systematic behavior of these devices at such battery levels.
- Power use drops off for both phones and the cellular cases of tablets at a very high filling level (above 0.95). Our data does not offer an explanation for this; it could, e.g., be caused by the Power Management Integrated Circuits (PMICs) and system software executing a power management routine to protect the battery from high-voltage stress.

Generally, we find that power consumption does not depend on the battery level except for very low and high levels. Accordingly, an investigation of the A/B test data shows that the difference between the unpaced and the paced cases also does not significantly depend on the initial battery level. Filtering the A/B test data to exclude these ranges did not significantly change the overall statistical behavior.

Received 5 June 2026; accepted 10 September 2026