

BURST: Proxy-Assisted Traffic Batching for Energy-Efficient Wi-Fi Bulk Transfers

Vany V. Ingenzi*
UCLouvain, Belgium
vany.ingenzi@uclouvain.be

Louis Navarre
UCLouvain, Belgium
louis.navarre@uclouvain.be

Florentin Rochet
UNamur, Belgium
florentin.rochet@unamur.be

Julien Montavont
University of Strasbourg, France
montavont@unistra.fr

Olivier Bonaventure
UCLouvain & WELRI, Belgium
olivier.bonaventure@uclouvain.be

Abstract

The power consumption of Wi-Fi interfaces is largely dictated by the time the chip spends in an active state. In scenarios where the Wireless Local Area Network (WLAN) offers significantly higher bandwidth capacity than the Wide Area Networks (WANs), this throughput asymmetry creates an opportunity to temporarily buffer traffic and deliver it in high-speed bursts. Hence, the Wi-Fi chip spends more time in idle/low-power state and this reduces the device's battery usage. We propose BURST, an explicit buffering proxy that exploits these opportunities when they occur. Unlike previous proxy-based solutions, BURST is application-agnostic and preserves the end-to-end encrypted TLS session.

ACM Reference Format:

Vany V. Ingenzi, Louis Navarre, Florentin Rochet, Julien Montavont, and Olivier Bonaventure. 2026. BURST: Proxy-Assisted Traffic Batching for Energy-Efficient Wi-Fi Bulk Transfers. In *The 22nd International Conference on emerging Networking EXperiments and Technologies (CoNEXT Companion '26)*, December 07–10, 2026, Utrecht, Netherlands. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3844665.3859459>

1 Introduction

Smartphones and IoT devices have become widely adopted because their increasing computational power enables applications to run with higher throughput and lower latency for increasingly complex tasks. However, this increase in capabilities also increases the device's energy consumption during transmission. To alleviate this, devices are becoming more energy-efficient, i.e., consuming less energy for the

same task. However, the efficiency regime of recent wireless technologies is only available when devices communicate at high throughput [11, 12], which is not always achievable across the Internet [3].

Indeed, Cloudflare's Internet Quality Index (IQI) [3], which represents the measured throughput of real connections, such as web browsing, shows that the median download throughput is 7, 12, 27, and 22 Mbps for Africa, Asia, Europe, and North America, respectively. These throughputs are below modern Wi-Fi networks measured throughputs, e.g. ~1 Gbps [10]. When the WLAN is not the bottleneck, one can exploit this throughput asymmetry to let the device consume less energy per transfer. Deploying a proxy in the *un-bottlenecked* network segment lets it accumulate slow incoming traffic from the WAN before flushing it rapidly over the high-speed LAN.

Several prior works have explored proxy-based approaches to reduce network energy consumption [4–6, 8, 9]. Some of these works are either application-specific [4, 5, 8], while others need access to decrypted traffic or application state [6, 9]. The closest work, Catnap [6], learns transfer size from clear-text headers or server-supplied hints; with 99.4 % of requests in the HTTP archive crawl [1], only the server option remains making it practically hard to deploy. Another similar work is Percy [5], which exploits RTT differences instead of throughput differences. We address the need for a generic TCP-level, application-agnostic mechanism. This work leverage the difference in throughput to offer energy savings, further work will also leverage RTTs asymmetry. Our approach preserves the end-to-end security of TLS connections, *requires no server-side modifications*, and still enables significant energy savings.

In this poster, we show that BURST leverages the existing asymmetry between LAN and WAN throughput to let the client enter idle/low-power periods during long transfers. BURST has a proxy that dynamically buffers data based on observed link capacity, LAN throughput and WAN throughput. As a result, the mobile device consumes less energy. The clients and proxy collaborate to minimize the impact on

*Vany V. Ingenzi is a FRIA grantee of the Fonds de la Recherche Scientifique - FNRS.



This work is licensed under a Creative Commons Attribution 4.0 International License.

CoNEXT Companion '26, Utrecht, Netherlands

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-3006-1/2026/12

<https://doi.org/10.1145/3844665.3859459>

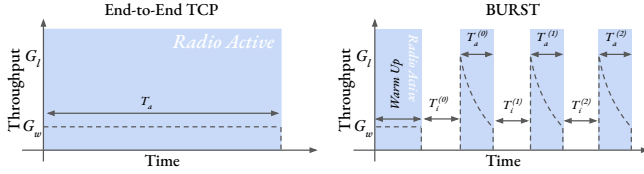


Figure 1: Simplified time diagram comparing an end-to-end connection to a BURST proxied connection, along with the operating throughput towards the client. The superscript denotes the batch number. The light blue zones depict when the WNIC is in an active state.

the transfer's completion time. We focus on bulk transfers, i.e., transfers that can tolerate packet delays such as cloud synchronization, software updates, and file downloads. We implement and evaluate BURST in our campus network with three different smartphones. Our experiments show that in the presence of Wi-Fi Power Save Mode (PSM), BURST can save up to 53 % of the total device energy consumption during screen-off transfers, while increasing the median flow completion time by only 0.45 s.

2 BURST: Design and Architecture

BURST relies on two core insights presented in the literature. First, Wi-Fi interface power consumption is mainly dictated by active time rather than instantaneous bandwidth utilization [6, 10, 12]. Second, per-flow LAN capacity is often significantly higher than WAN capacity [3, 6].

BURST's system includes a proxy that the client explicitly connects to. The proxy runs on the Wi-Fi Access Point (AP) or another LAN host, ideally on-path to ensure minimal latency impact.

Connection Establishment. The client uses the initial byte stream of its connection to the Proxy to exchange control information between the client and the proxy. For this, we use the 0-RTT Convert Protocol [2] and TCP Fast Open option. This control message include target server address and port, along with the maximum batch duration B . After sending the control information all subsequent data is forwarded by the proxy between the client and the server.

Buffering Mechanism. Figure 1 illustrates BURST's behaviour compared to an end-to-end baseline, from a temporal and throughput perspective at the client. The standard end-to-end connection operates continuously at the WAN throughput (G_w). BURST divides a connection into batches of at most B duration, a parameter that the client communicates.

Each batch is made of (i) an idle phase of duration T_i , during which the proxy buffers data for the client in a FIFO queue, followed by (ii) an active phase of duration T_a , in which it drains and forwards the buffered data. BURST dynamically plans each batch based on the difference between the estimated WAN throughput G_w and LAN throughput G_l (see Eq. (2)). As such, the amount of data the proxy buffers during an idle phase is approximately $G_w T_i$. During an active

phase, the proxy flushes buffered data but also *new data* sent by the server during the active phase, of size $G_w T_a$.

The batch total duration, $B = T_a + T_i$, is constrained such that the proxy empties its enqueued data at the end of the active phase. This is only ensured if the throughput between the proxy and the client is sufficiently high to transmit all data during the active phase only:

$$\widehat{G}_l T_a = G_w T_i + G_w T_a = (T_a + T_i) G_w$$

BURST computes these phases duration as

$$T_i = B - T_a, \quad T_a = \frac{G_w}{\widehat{G}_l} B \quad (1)$$

The ratio between WAN and LAN throughput $\beta = G_w / \widehat{G}_l$ governs the duration of the active phase T_a . Buffering is only possible when $\beta < 1$, which falls under the initial hypothesis of BURST [3].

Estimating the achievable LAN throughput. The proxy regularly updates its estimation at batch n , $\widehat{G}_l^n$ using a *moving average* based on previous values G_l^{n-1} , the WAN throughput, and the current Wi-Fi link capacity L :

$$\widehat{G}_l^n = \alpha^n G_l^{n-1} + (1 - \alpha^n) L, \quad \alpha^n = \min\left(1, \frac{G_w^n}{G_l^{n-1}}\right), \quad L = (1 - b) C \quad (2)$$

where b is the fraction of time the channel is busy, and C is the PHY maximum rate. If the proxy runs directly on the access point, it reads b and C directly from the Wi-Fi driver.

3 Evaluation

Environment. We run BURST's proxy on a RTROM01-2G Turrus Omnia 802.11ac access point (5 GHz). The smartphones used are a Pixel 7a and a Fairphone 5 (FP5), both using Android 14, and a Pixel 10 Pro with Android 16. Each device uses the default power-save settings, and we remove other apps. The pixels have On-Device Power Monitor (ODPM) rails that measures the per-component power profile. We use a Monsoon High Voltage Power monitor to measure the total device energy from flow start to completion. We combine both tools because prior work demonstrates that, under heavy workloads, aggregate ODPM rail measurements can diverge significantly from battery counters and total device power measurements [7]. The workload is a download over HTTPS from a server in our university's datacenter. Each individual flow's WAN rate is shaped to 22 Mbps, which is the median per-connection throughput that Cloudflare reports for North America [3]. However the LAN is unconstrained, with a measured throughput of 600 Mbps. The maximum batch duration B is set to 1 s. We report means over 10 runs.

BURST enables WNIC low-power. Figure 2a illustrates the total time the WNIC spends in low-power/idle state under throughput asymmetry. BURST allows the client WNIC to be in low-power 38-51% of the transfer, an improvement over the 1-20% achieved by the end-to-end baseline. The WNIC idle residency depends on the smartphone. For small 1 MB flows, we observe no significant difference; the transfer

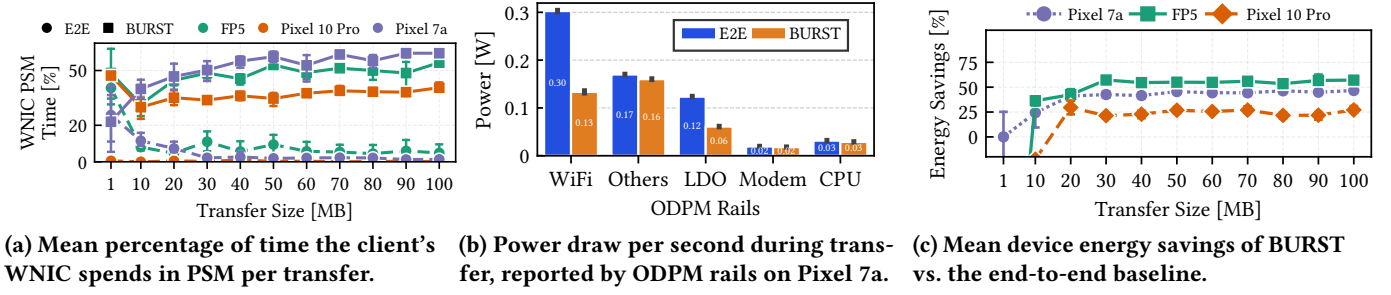


Figure 2: Single-flow evaluation with a batch duration of 1 s. Error bars represent 95 % confidence interval.

completes in under one second, finishing entirely within the initial active phase. Beyond 1 MB, for the baseline, continuous packet arrivals natively force WNIC to remain awake, restricting its average PSM time to 4%. On the contrary, BURST actively paces the flow, enabling the WNIC to spend an average of 48 % of the transfer time in PSM.

BURST reduces WNIC and system power consumption. Figure 2b illustrates the per-second power consumption across hardware components of the Pixel 7a. Compared to the baseline, BURST reduces WNIC power consumption during the transfer by ~56 % (from 0.3 W to 0.13 W). Alongside these Wi-Fi energy savings, we observe a reduction in the power drawn by the Low Dropout (LDO) regulators. The baseline end-to-end transfer necessitates a higher sustained current draw, thereby increasing the regulatory load on the LDOs.

Energy Savings Per Transfer Size. Figure 2c illustrates how total device energy savings evolve alongside transfer size. BURST's relative energy savings increases proportionally with transfer sizes until plateauing at approximately 24, 53, and 45 % for the Pixel 10 Pro, Fairphone 5 and Pixel 7a respectively. For short 1 MB transfers, we observe high variance and a net energy penalty. This is due to the transfer completing during the initial phase and the control information exchange overhead. Results omitted for brevity, show that on a Pixel 7a, BURST yields device energy savings of 20 % to 50 % for WAN throughputs up to 200 Mbps, and 48 % for data volumes up to 1 GB.

Impact of BURST on Flow Completion Time. Across all measurements, BURST introduces a median FCT penalty of 0.45 s, and a 95th percentile of 1.06 s. For small flows under one second, this translates to a 40 % relative increase in transfer time. However, for transfers taking several seconds or longer, this overhead decreases to a negligible fraction of the total FCT: under 8 % from 10 MB and above.

4 Discussion and Future Work

Further work will integrate RTT asymmetry to further improve energy savings of short transfers. Additionally, we aim at performing system-wide power analysis to see the impact of BURST at the AP. Furthermore, we will also shade more light to fairness regarding other LAN users. Finally, we will

evaluate BURST's on trace driven measurements to see its performance under typical mobile workload.

References

- [1] Httparchive. URL <https://httparchive.org/reports/state-of-the-web>.
- [2] O. Bonaventure (Ed.), M. Boucadair (Ed.), S. Gundavelli, S. Seo, and B. Hesmans. 0-RTT TCP Convert Protocol. RFC 8803 (Experimental), July 2020. ISSN 2070-1721. URL <https://www.rfc-editor.org/rfc/rfc8803.txt>.
- [3] Cloudflare. Cloudflare Radar docs — developers.cloudflare.com. <https://developers.cloudflare.com/radar/glossary/#iqi>. [Accessed 25-08-2025].
- [4] A. Y. Ding, B. Han, Y. Xiao, P. Hui, A. Srinivasan, M. Kojo, and S. Tarkoma. Enabling energy-aware collaborative mobile data offloading for smartphones. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 487–495, 2013. doi: 10.1109/SAHCN.2013.6645020.
- [5] N. Ding, A. Pathak, D. Koutsonikolas, C. Shepard, Y. C. Hu, and L. Zhong. Realizing the full potential of psm using proxying. In *2012 Proceedings IEEE INFOCOM*, pages 2821–2825. IEEE, 2012.
- [6] F. R. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 107–122, 2010.
- [7] É. Guégain, R. Raes, N. Chachignot, C. Quinton, and R. Rouvoy. Androwatts: Unpacking the power consumption of mobile device's components. In *2025 IEEE/ACM 12th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pages 71–81. IEEE, 2025.
- [8] N. Jiang, M. C. Vuran, S. Wei, and L. Xu. Qos-aware network energy optimization for danmu video streaming in wifi networks. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pages 1–10. IEEE, 2021.
- [9] M. Jimeno, K. Christensen, and B. Nordman. A network connection proxy to enable hosts to sleep and save energy. In *2008 IEEE International Performance, Computing and Communications Conference*, pages 101–110. IEEE, 2008.
- [10] R. Liu and N. Choi. A first look at wi-fi 6 in action: Throughput, latency, energy efficiency, and security. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–25, 2023.
- [11] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z. M. Mao, et al. A variegated look at 5g in the wild: performance, power, and qoe implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 610–625, 2021.
- [12] S. K. Saha, P. Deshpande, P. P. Inamdhar, R. K. Sheshadri, and D. Koutsonikolas. Power-throughput tradeoffs of 802.11 n/ac in smartphones. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 100–108. IEEE, 2015.