What Does It Cost to Run a BTX Node? We Measured.

Mining is famous for burning power. Verifying is not mining. We measured a full BTX node on a Mac mini-class machine: roughly one CPU core during the heaviest catch-up phase, near-idle at the tip, and about as much electricity as a small LED bulb. What the network gets back is harder to price, and worth more.

JULY 11, 2026 · 6 MIN READ · EASYBTX.COM/RESEARCH/BTX-NODE-POWER

ABSTRACT

We ran a full BTX node on an ordinary Mac and measured what it actually uses: about one CPU core while catching up, a fraction of that at the tip, around 150 MB of memory, and 18 GB of disk in July 2026. Here is the data, the method, and what the network gets in return, with a dated postscript on the disk figure.

Everyone knows that mining costs electricity; that is the point of proof-of-work. But a full **node** is not a miner, and mixing the two up makes people think supporting a network is expensive. So we measured it.

The setup: an ordinary Apple-silicon Mac (M2 Pro class), running a full BTX node through easyBTX Node v0.1.1: snapshot fast-start, un-pruned, default settings. Numbers below are from the live process, sampled during the node's first hours. Nothing here is a simulation.

What we measured

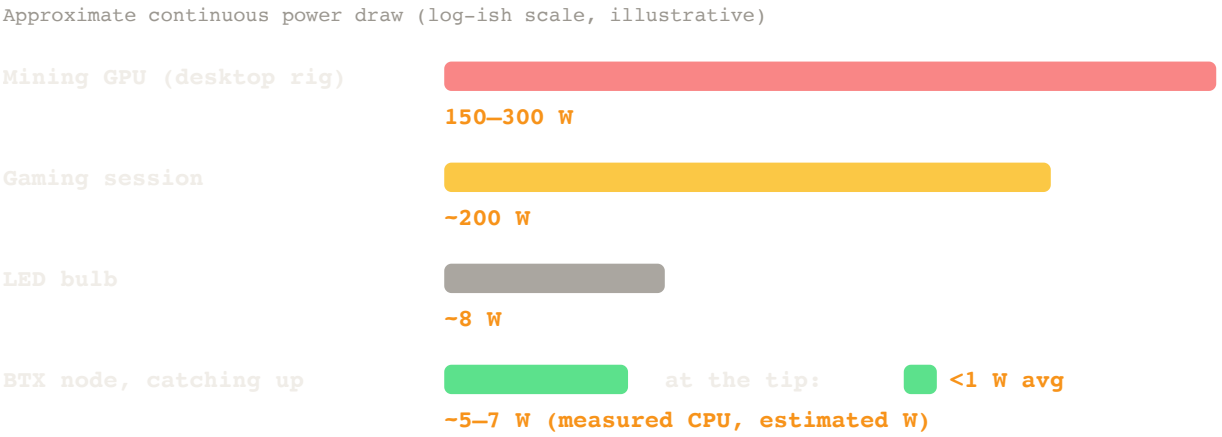
The heaviest thing a node ever does is the initial catch-up: after the fast-start snapshot loads, it downloads and verifies the chain's history in the background. During that phase we sampled the daemon repeatedly:

- **CPU: roughly one core** (samples ranged 80 to 100% of a single core, about 8% of this machine's total CPU). The rest of the machine stayed fully usable.
- **Memory: 100–160 MB** resident. Less than a handful of browser tabs.
- **Network: ~450 MB** for the verified snapshot, then ~7 GB of historical blocks over the first hours. Upload stayed minimal until the node was ready to serve peers.
- **Disk: about 18 GB** for the full, un-pruned chain today, growing with the network.

Once a node reaches the chain tip, the picture changes completely. A new BTX block arrives roughly every 90 seconds; verifying one block takes well under a second of work. In between, the daemon idles. Averaged out, a node at the tip uses only a small fraction of one core.

Power, in plain terms

We estimate electricity from the measured CPU load. A sustained single performance core on this machine draws on the order of 5 watts; spread across the catch-up phase, the node's draw was **roughly an LED bulb**. At the tip, with the CPU idle between blocks, the average falls **well below one watt**, closer to a phone left charging overnight than to any gaming or mining workload.



The asymmetry is the whole story: proof-of-work is expensive to **produce** and cheap to **check**. That is by design: it is what lets thousands of small machines hold a network's producers honest without burning what the producers burn.

What the network gets

The cost side fits in a lightbulb. The value side is harder to price, but on a young network it is unusually concrete:

Independent verification. Every node re-checks every block against the consensus rules. Nobody, not a pool, not an exchange, not us, can slip an invalid block past a network of independent verifiers. Each additional node raises the number of parties that would all have to be fooled at once.

Relay and reach. Nodes pass blocks and transactions peer to peer. More well-connected nodes means blocks propagate faster and small miners are less disadvantaged by network distance.

Bootstrap for newcomers. Here is an honest detail from our own code: BTX is young enough that its DNS seeding is unreliable, so node software ships with a short hand-maintained list of known-good peer addresses to make sure a fresh node can connect at all. On a network where the peer list fits in a paragraph, one more always-on node is not a rounding error. Our fresh node was serving 14 peer connections within its first hour.

A living copy of history. Every un-pruned node keeps the complete chain. Copies held by independent people, in different places, on different hardware, are what "decentralized" physically means.

The honest limits

This is one machine, measured over hours, not a lab study across hardware. Watt figures are estimates derived from measured CPU load and published per-core power characteristics, not wall-meter readings. We say "about an LED bulb", not "exactly 6.2 W". Disk is the one real cost that grows over time, and the node app shows it prominently rather than hiding it. And to repeat the FAQ: BTX has no node rewards today. A node is something you run for the network, which, on a chain this young, is exactly when it matters most.

Measured 2026-07-11 on an M2 Pro Mac running easyBTX Node v0.1.1 (btxd v0.32.12, Metal validation backend, default settings). Methodology: repeated `ps` sampling of the daemon process during background chain download, plus the node's own RPC counters for network totals.

Postscript added 22 August 2026
The cost figures above were taken on 11 July 2026, before BTX changed its proof of work. The CPU, memory and power readings are still the right order of magnitude for what a node costs to run, and we are not restating them from memory. Three things around them have changed enough that a reader planning a node should know, and one figure in the article does not agree with our own companion study.

What validation means changed at block 185,000. Since MatMul v4.7 activated on 10 August 2026, a node with a qualified GPU replays each block's computation itself, while every other node, which is most of them, follows a signed frontier of attestations fanned out by archive nodes. That is a different workload from the one measured here, and it has a failure mode this article could not have covered: when attestations stop flowing, a node freezes in place while looking perfectly healthy. We wrote that up separately in why nodes get stuck and the week the chain froze.

Your node version now matters more than your hardware. BTX Core v0.33.3, published 17 August 2026 after a network-wide stall, introduced a difficulty floor that activates at block 191,714. A node on any older build stops at height 191,713 and stays there. No amount of CPU or disk fixes that. Of 58 community nodes we could see, 29 were running v0.33.3.

easyNode versions, honestly. easyNode is published at 0.6.10 on Mac, 0.6.6 on Windows and 0.6.5 on Linux. The Linux build predates the v0.33.3 engine, so a Linux easyNode parks at 191,713 today. A newer Linux build is written up in the changelog but is not published, and we are not going to link a release that does not exist.

The disk figure needs a caveat. This article measured about 18 GB for the full un-pruned chain on 11 July 2026. Our companion study, why run a BTX node, measured about 105 GB one day later, and about 105 GB is the number to plan around today. We have not reconciled the two readings and are not going to guess at which phase of sync produced the smaller one, so plan for the larger. If disk is the constraint, Keeper is a pruned node at roughly 10 GB that still serves signed block attestations to the rest of the network.

A free way to tell the two apart. Guardian is a standalone script that watches any BTX node and distinguishes "the network is waiting" from "your node is behind", dialing archive peers only in the second case. It never restarts your node. See Guardian.

Frequently asked questions

Does running a BTX node earn me coins?

No. BTX has no node incentive today. Nodes verify and relay, miners earn the block reward. Running a node is a contribution to the network's independence, not an income. If the network ever adds node incentives, that would be a protocol change you would hear about long in advance.

Is running a node the same as mining?

No, and the difference is exactly where the electricity goes. Mining races to solve a proof-of-work puzzle continuously at full power. A node only CHECKS finished work: it verifies each new block once, roughly every 90 seconds, and idles in between. That is why a miner draws hundreds of watts and a node draws about as much as an LED bulb, briefly.

Will a node slow down my computer?

During the initial catch-up the node uses about one CPU core, which a modern multi-core machine barely notices. Once it reaches the chain tip, the work drops to verifying one block every ~90 seconds, near zero in between. Memory stays modest (we measured 100–160 MB). The most noticeable cost is disk. Our July 2026 measurement was about 18 GB, and it grows with the chain. Note added 22 August 2026: our companion study measured a full un-pruned node at about 105 GB a day later, so treat 18 GB as a floor rather than the size to plan for. If disk is the constraint, Keeper is a pruned node at roughly 10 GB.

How much bandwidth does a node use?

Our fresh node downloaded a ~450 MB verified snapshot plus about 7 GB of historical blocks during its first hours, then settled into normal peer-to-peer traffic: new blocks arrive every ~90 seconds and get relayed onward. After the initial sync, bandwidth is closer to background-app territory than to video streaming.

Why does a young network need my node more than Bitcoin needs another one?

Bitcoin has tens of thousands of reachable nodes; one more changes little. BTX is months old and sparse enough that node software ships with a short list of known-good peer addresses to guarantee newcomers can connect at all. On a network that size, each additional always-on node measurably improves how easily new participants join and how hard the chain is to disrupt.

Can I stop or remove the node whenever I want?

Yes. A node is not a commitment: stop it and the network simply has one fewer peer; start it again and it catches up from where it left off. Removing the data returns the disk space. Nothing about running a node locks anything in.

Published by easyBTX, which builds software for BTX and is not the BTX core team. This paper is educational research, not financial advice, and it is not an official BTX document. Read the live version and check the sources at easybtx.com/research/btx-node-power.