# Test a Minecraft network compression threshold change

Minecraft Java can compress packets above a configured size. Compression trades processing work for smaller transfers, so changing the threshold can affect bandwidth and CPU use differently. It is not a universal fix for low TPS, high ping or a slow internet connection. Start with a clear symptom and measurements you can compare.

## Check the connection path

Record the Paper build, current threshold, typical online count and whether players connect directly or through a proxy. A proxy can apply its own compression policy on another connection leg. Changing a backend alone may not change traffic between players and the public proxy. Coordinate with the network administrator before testing a multi-server setup.

Separate the symptom into useful categories: delayed game ticks, slow chunk delivery, high round-trip latency or bandwidth pressure. A profiler can help identify server-side work, while client observations and network graphs may show a different bottleneck. Do not assume every report of lag comes from packet compression.

## Plan a controlled comparison

Download `server.properties` privately and record the existing `network-compression-threshold`. Paper's reference expresses the threshold in bytes and documents negative values as disabling compression. That explains the setting; it is not advice to disable it. Turning compression off can substantially increase transferred data for some workloads.

Choose a modest, documented test change only after discussing the relevant metric with support if necessary. Keep the same client version, route and general activity for the comparison. Flying through unexplored terrain in one trial and standing in a lobby in the other will not provide a useful result. Do not run a flood or synthetic traffic test against the live service.

# Apply one change

Announce a short restart, stop the server cleanly and edit the existing property. Use a whole numeric value in the format supported by the installed release. Do not alter memory flags, distances, packet limits and compression together. Save, start normally and inspect the log for configuration or connection errors.

Use the documentation for the exact server and proxy versions. Current 26.x Paper settings and an older 1.21.11 deployment may have different surrounding behaviour even when the property name is the same. Avoid copying an optimisation bundle whose other options you cannot explain.

# Evaluate the result

Compare a representative session before and after the change. Record player count, activity, tick performance, CPU observations and network throughput where the panel exposes them. Ask the same test players about chunk delivery and disconnects. Separate a measured change from subjective improvement and avoid attributing a quiet period to the new threshold alone.

If network statistics are unavailable, acknowledge that limitation. A lower CPU graph without bandwidth data is not enough to prove the overall trade-off is better. Likewise, packet size changes cannot remove distance-based latency between a player and the server location.

# Roll back

Restore the previous threshold during a clean stop and restart if the result is worse, unclear or unnecessary. Keep the comparison notes with the build and connection path so the same experiment does not have to be repeated blindly.

If players continue to experience timeouts, inspect their connection errors and the server log separately. Do not disable account authentication, remove proxy protections or broaden unrelated ports as part of a compression experiment.

# Related help

- Diagnose Minecraft lag with TPS, MSPT and spark
- Fix common Minecraft connection errors

- Set up a Velocity network with protected backend servers

# Reference

Paper network compression property. Velocity network compression settings.

---

Revision #1
Created 2026-11-09 10:42:41 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:41 UTC by CloudExa Hosting Admin