**Author:** Nic Betancourt

**Date & Time:** 6/23/2025, 12:43pm CDT

**Version 1.0**

**Description:**

For this case study, I walk through how I identified and fixed a major speed issue on my home network – specifically when using ProtonVPN’s WireGuard Protocol through an ASUS router’s VPN Fusion feature. By diving into routing and network configuration, I found success with increasing my encrypted download speed by **60-80%.**

**Problem:**

I first noticed a delayed response time from applications and web browsers during system start-up. Web pages would take several seconds to load, and some apps with online components (e.g., Spotify, Google) appeared to hang briefly or failed to connect on the first attempt.

The issue was isolated to my desktop, which is connected via Ethernet and configured to route traffic through ProtonVPN using ASUS VPN Fusion. Other devices on the network — whether using the VPN or not — were unaffected. Speed tests on the desktop confirmed consistently degraded performance, particularly for download speeds, which ranged from **~60–200 Mbps** despite a Gigabit internet connection.

This prompted a deeper investigation into the VPN tunnel’s routing configuration and potential MTU-related inefficiencies.

**Phase 1: Optimizing VPN Routing and Server Selection**

**What I observed:**

* Checked Router speed, showed full bandwidth: **~936 mbps down / ~845 mbps up**
* Desktop routed by VPN showed poor performance: **~60-200 mbps down**
* Desktop speed test without VPN: **~970 mbps down / ~845 mbps up**

**Actions I took:**

1. Ran baseline ping tests to assess response time and DNS resolution:
   1. ping google.com (request timed out)

A blue screen with white text

AI-generated content may be incorrect.

* 1. ping 8.8.8.8 (successful response)
  2. nslookup google.com (successful response, except: Server=unknown)

A screenshot of a computer

AI-generated content may be incorrect.

1. Verified that my desktop was properly assigned to the ProtonVPN client under the router’s VPN Fusion configuration, with NAT enabled
2. Identified the active VPN endpoint and switched to a lower-latency regional WireGuard server by generating a new configuration file through ProtonVPN’s dashboard

A screenshot of a computer

AI-generated content may be incorrect.

1. Disabled lower-priority devices from the VPN Fusion device list to reduce client load

**Result:**

Running ping tests revealed that IP-based pinging (e.g., ping 8.8.8.8) returned successfully, while domain-based pinging (e.g., ping google.com) timed out. The nslookup command returned correct IP addresses but showed the DNS server as "unknown." While this did not appear to cause immediate service issues, it helped confirm that DNS resolution over the VPN was inconsistent or slow.

After the VPN configuration and server switch, desktop download speed improved to **~350 Mbps**, with upload speeds around **~290 Mbps**. This confirmed that server endpoint selection, routing, and client priority were significant contributors to the slowdown.

**Phase 2: Resolving MTU Fragmentation for Better Throughput**

**What I observed:**

* After changing the VPN endpoint, download speeds remained lowand online services showed delays
* Router’s configuration file for VPN Fusion did not have a limit on MTU

**Actions I took:**

* Used powershell to test MTU Discovery
* ping -f -l 1500 8.8.8.8
* Working down from 1500 to find that 1380 was the biggest non-fragmenting load, resulting in a MTU of 1408 to use

A computer screen shot of a blue screen

AI-generated content may be incorrect.

**Result:**

I decided to manually set the MTU to 1400 in the Wireguard configuration file on my router. After making the manual change to the file, I went ahead and rebooted the router. Once the reboot finished, I retested the performance of my network. **The download increased to about ~800 mbps and the upload to around ~350 mbps.** The metrics of the speed test confirmed that packet fragmentation inside the tunnel was the cause of the remaining bottleneck.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Key Takeaways:**

A tuned configuration file and MTU can significantly reduce latency and packet loss, especially in encrypted tunnels such as Wireguard. Since this adjustment aligned the MTU across the local network and VPN tunnel, it allowed for more efficient packet delivery and improved consistency in application response times. Understanding routing logic, client prioritization, and packet-level tuning can significantly improve performance in VPN-secured environments. Further tuning and testing could potentially maximize performance, but utilizing a VPN could be limiting as it tunnels encrypted data.

**Next Steps:**

* **Validate IPv6 performance**: test for IPv6 support with VPN for capability and dual stack readiness
* **Enhance Log Monitoring**: configure captured logs for packets and correlate with firewall rules to understand traffic behavior
* **Assess Multi-Device Performance**: Continue to evaluate how multiple devices route through VPN Fusion effect throughput and DNS performance
* **Monitor Long-Term Stability**: Continue to run speed tests and check system logs to monitor key trends in VPN endpoint performance to help strategize future adjustments