

# Speed Boosting VPN: A Multi-Link Approach to Network Reliability and Performance

*Luka Liluashvili, Victor Zakaraia*

e-mail: viktor.zakaraia743@ens.tsu.edu.ge

Department of computer science, the Faculty of Exact and Natural Sciences, Tbilisi State University, 2  
University St

In the modern digital landscape, reliable and high-speed internet connectivity is essential for remote work, communication, and entertainment. However, users often face connection instability or bandwidth limitations due to environmental factors, despite having access to multiple network interfaces (e.g., Wi-Fi, Ethernet, Cellular). Contemporary devices support various connection methods, yet they typically utilize only one active interface at a time, leaving potential bandwidth unutilized. The primary problem addressed is the inefficiency and instability of single-link connections. When a connection slows down or disconnects, the user is forced to manually seek alternatives, resulting in interrupted sessions.

This paper proposes a "Speed Boosting VPN," a multi-link Virtual Private Network solution designed to aggregate multiple network connections into a single logical tunnel. By utilizing OpenVPN and Linux-based policy routing, the proposed system aims to increase throughput and ensure session persistence during network failures. The solution implements a custom watchdog mechanism for automatic failover, ensuring that if one link fails, traffic is seamlessly rerouted without user intervention. This approach addresses the limitations of standard remote access VPNs, which traditionally rely on a single interface.

The system leverages iproute2 for advanced routing management. We establish Policy Based Routing (PBR) by creating separate routing tables for each provider (e.g., eth0, wlan0). This ensures that packets are directed through the specific interface required by the OpenVPN daemon, preventing routing conflicts. The core functionality relies on modifying OpenVPN configurations (.ovpn files) to utilize remote-random or connection blocks for load balancing. To manage reliability, we employ a custom Python or Bash script serving as a watchdog. This daemon pings a reliable DNS (e.g., 8.8.8.8) every 5 seconds through each interface. Upon detecting a failure, the system automatically reroutes traffic to the remaining active interfaces within 1-2 seconds.

To verify the solution, we define specific performance metrics. Qualitative analysis demonstrates the ability to aggregate bandwidth, where the VPN throughput exceeds that of any single interface. Quantitative analysis utilizes iPerf3 and Speedtest-cli to measure throughput. Reliability is tested via 24-hour ping monitoring, with a target packet loss of <1% even during interface switching. Failover testing confirms that the system recovers the connection in under 2 seconds during physical disconnection simulations. This project demonstrates the viability of a software-based multi-link VPN to enhance internet reliability and speed.