WireGuard10G-IP Demo Instruction

WireGuard10G-IP Demo Instruction

Rev1.00 8-May-2026

This document provides a comprehensive guide for demonstrating the WireGuard10G IP core designed for 10 Gigabit Ethernet using FPGA evaluation boards. The WireGuard10G-IP core implements the modern and high-performance WireGuard VPN protocol to establish fast and secure network tunnels through hardware-based processing. Users will find step-by-step instructions for setting up the necessary software on a PC, configuring the FPGA device, and verifying the tunnel's operation using either a serial console or the provided graphical user interface (GUI) application.

The initial part of the guide focuses on the hardware environment and host PC preparation, covering essential network configuration details such as speed, duplex settings, and other network properties required for 10 Gbps operation. This is followed by detailed procedures for installing the WireGuard software on a PC and programming the FPGA evaluation board.

The final sections of this document describe the operational commands and demonstration scenarios available for the IP core. It includes a full reference of serial console commands for manual configuration, along with walkthroughs for both point-to-point serial console setups and selective encapsulation using the GUI application. Additionally, the document concludes with performance benchmarks using iperf3 to provide a clear comparison between software-based WireGuard and the hardware-accelerated IP core.

1 Environment Setup

To operate WireGuard10G-IP demo, please prepare the following test environment.

1) FPGA development board.
2) User PC and Endpoint PC with 10 Gigabit Ethernet or connecting with 10 Gigabit Ethernet card.
3) 2x10 Gb Ethernet cable:
   i) 10 Gb SFP+ Passive Direct Attach Cable (DAC) which has 1-m or less length
   ii) 10 Gb SFP+ Active Optical Cable (AOC)
   iii) 2x10 Gb SFP+ transceiver (10G BASE-R) with optical cable (LC to LC, Multimode)
4) Micro USB cable for JTAG connection between FPGA board and User PC.
5) Micro USB cable for UART connection between FPGA board and User PC.
6) Vivado tool for programming FPGA installed on Test PC.
7) Serial console software such as TeraTerm installed on PC. The setting on the console is Baudrate=115200, Data=8-bit, Non-parity and Stop=1.
8) Demo configuration file (To download this file, please visit our web site at www.design-gateway.com).

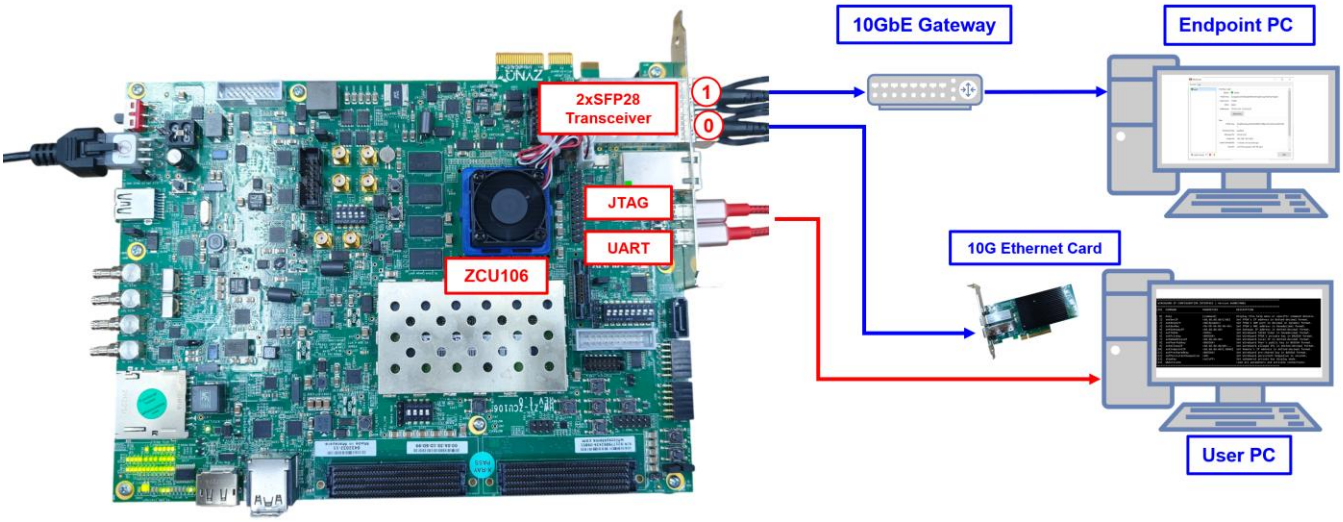

Figure 1 WireGuard10G-IP demo environment on ZCU106 board

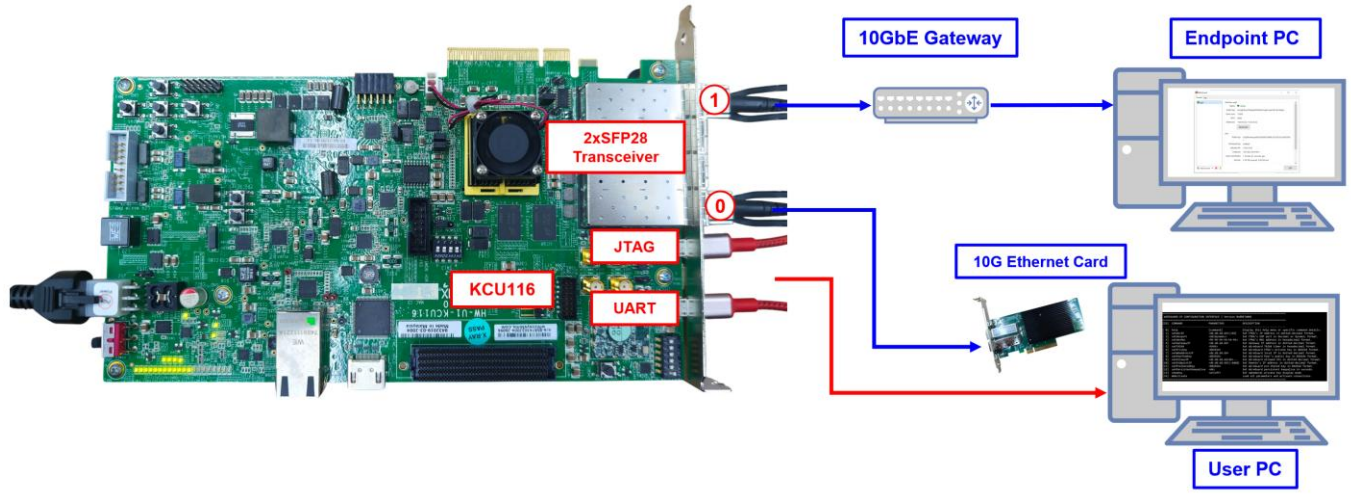

Figure 2 WireGuard10G-IP demo environment on KCU116 board

2 PC Setup

Before running the demo, configure the network settings on the User PC so that its 10 GbE interface is on the same subnet as the FPGA. The example below describes the settings for a 10 GbE connection.

2.1 Speed and Duplex Setting

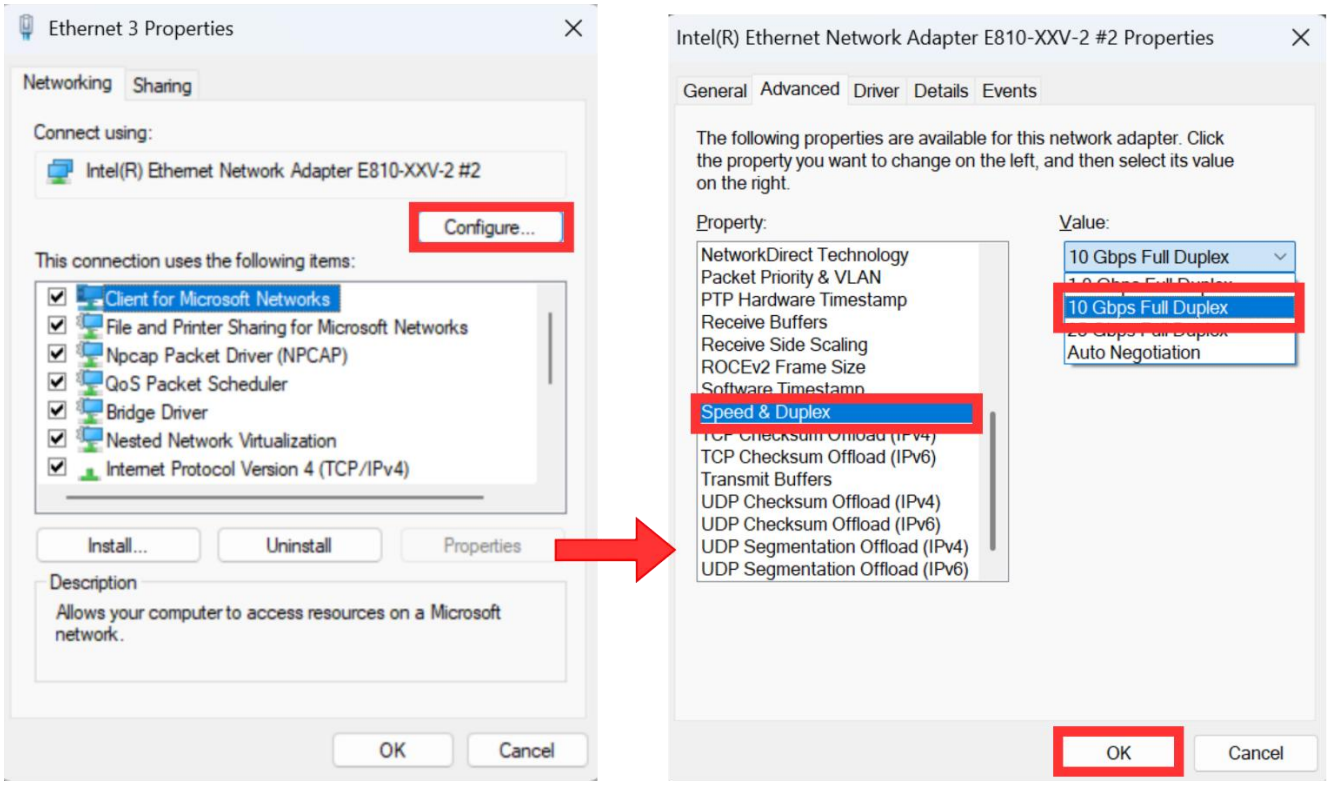

Figure 3 Set Link Speed = 10 Gbps

1) On Local Area Connection Properties window, click “Configure”, as shown in Figure 3.
2) On Advanced Tab, select “Speed and Duplex”. Set the value to “10 Gbps Full Duplex” for running 10 Gigabit transfer test, as shown in Figure 3.

2.2 Network Properties Setting

Some of network parameter setting may affect to network performance. The example of network properties setting as follows.

1) On “Interrupt Moderation” window, select “Disabled” to disable interrupt moderation which would minimize the latency during transferring data, as shown in Figure 4.

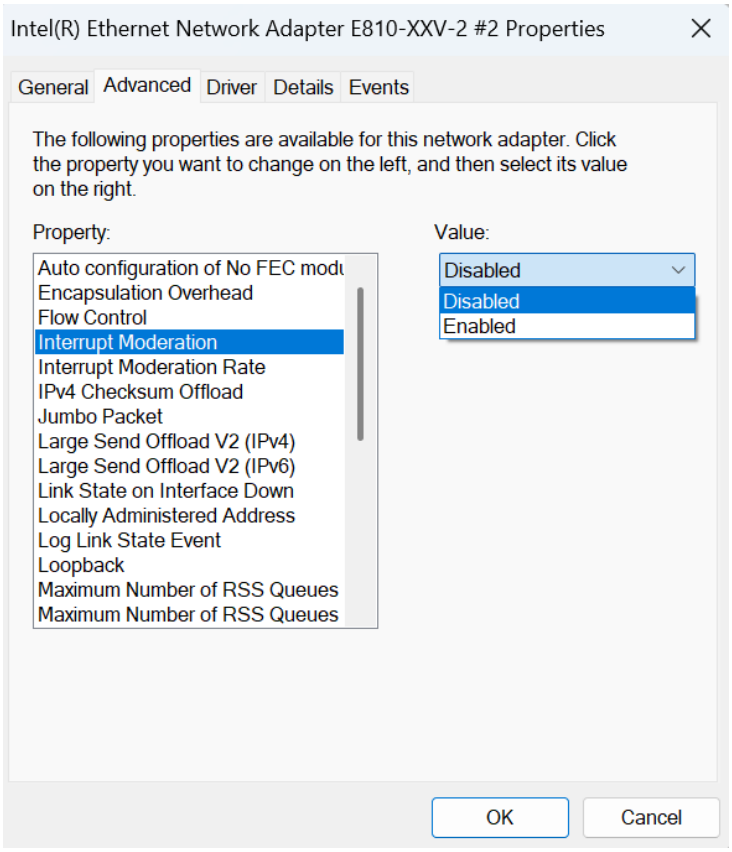

Figure 4 Interrupt Moderation

2) On “Interrupt Moderation Rate” window, set value to “OFF”, as shown in Figure 5.

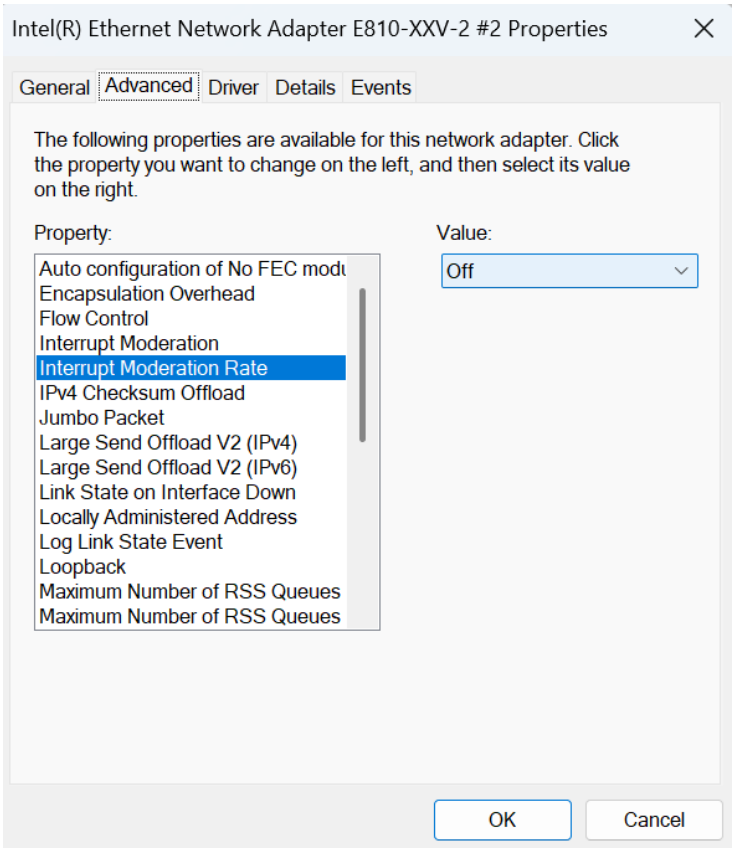

Figure 5 Interrupt Moderation Rate

3) On “Jumbo packet” window, set value to “9014 Bytes”, as shown in Figure 6.

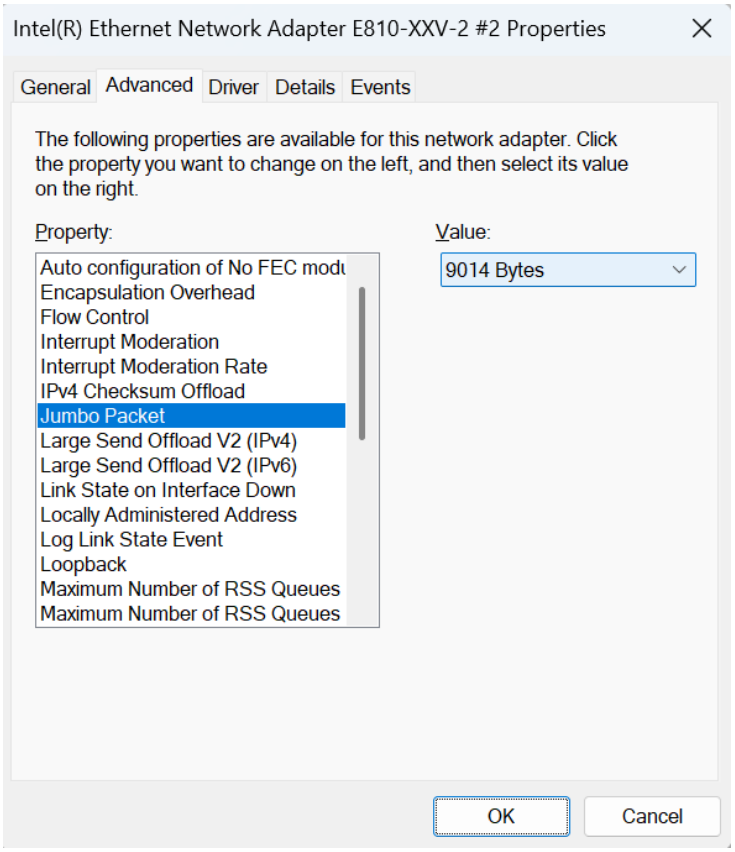

Figure 6 Jumbo packet

4) On “Receive Buffers” window, set value to the maximum value, as shown in Figure 7.

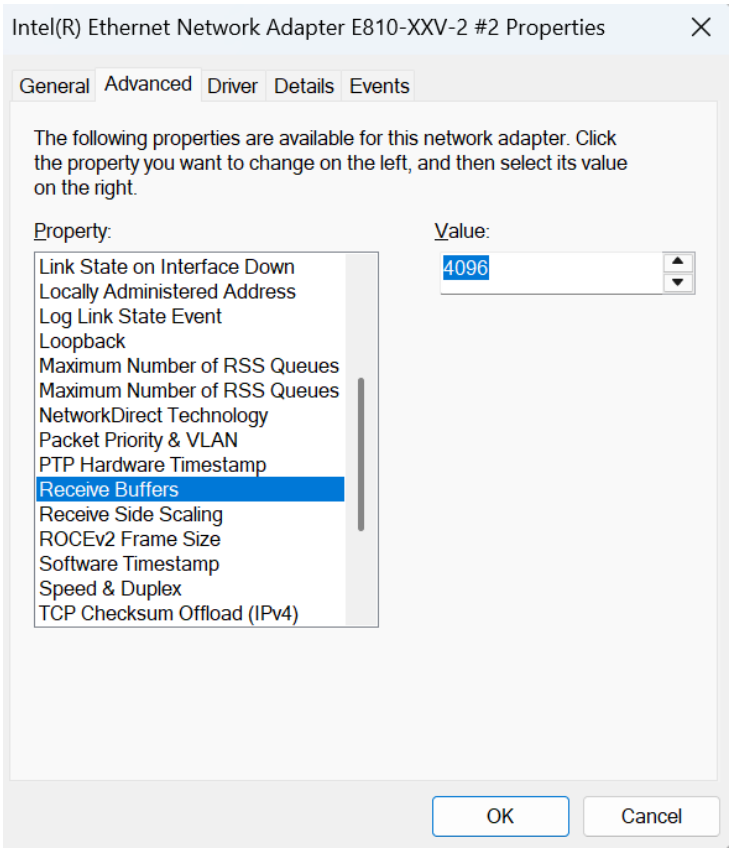

Figure 7 Receive Buffers

5) On “Transmit Buffers” window, set value to the maximum value, as shown in Figure 8.

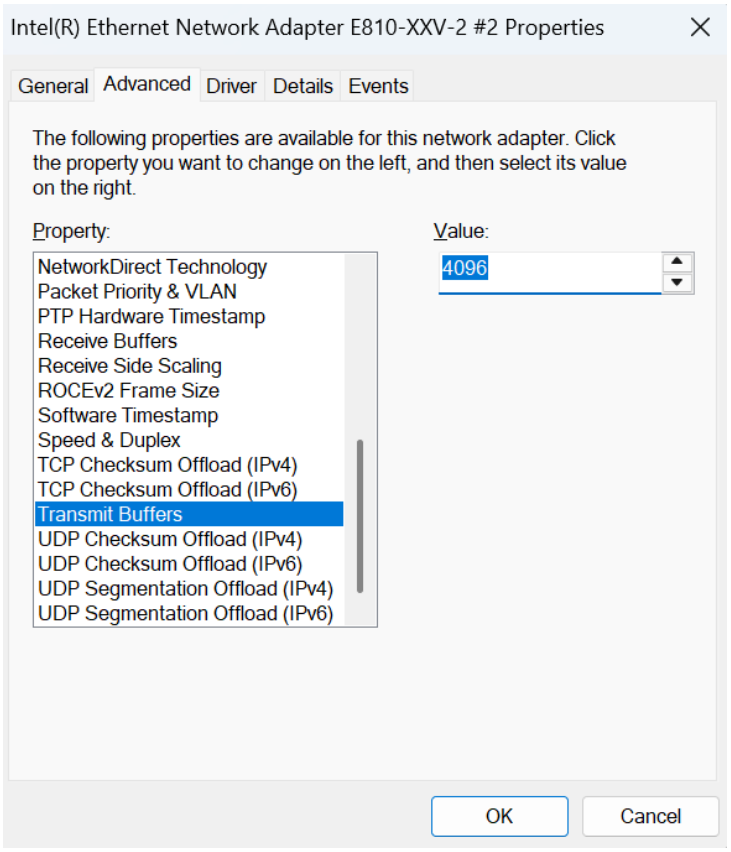

Figure 8 Transmit buffers

3 WireGuard Software Setup

WireGuard is a modern, open-source VPN protocol known for its simplicity, high performance, and strong cryptographic design. This section provides an example of configuring the WireGuard software for use with the demo.

3.1 Install WireGuard

To install WireGuard on the PC:

1) Download the WireGuard Windows installer from https://www.wireguard.com/install.
2) Run the installer and follow the on-screen instructions.
3) Once installed, the WireGuard application will be available in the system tray and Start menu.

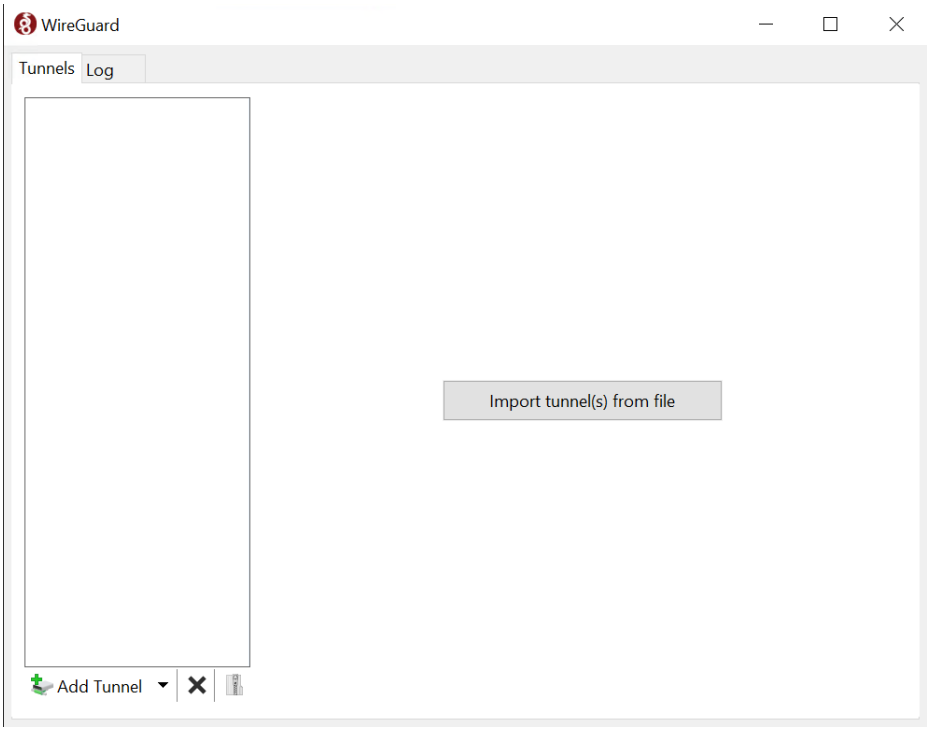

Figure 9 WireGuard application on Windows

3.2 Set up WireGuard

1) Open the WireGuard application and click “Add Tunnel” > “Add empty tunnel...”. A key pair will be automatically generated, with the private key pre-filled in the configuration editor.
2) Copy the displayed public key, as it will be required when configuring the FPGA demo. Complete the configuration by filling in the remaining fields as shown below.

```
[Interface]
PrivateKey = <Device private Key>
Address = <Tunnel IP, e.g. 10.0.0.1/32>
ListenPort = <Optional Listen Port, default: 51820>
MTU = <Optional MTU, default: 1420>

[Peer]
PublicKey = <Peer Public Key>
AllowedIPs = <Allowed IP, e.g. 10.0.0.3/32>
Endpoint = <Optional Peer endpoint, e.g. 192.168.7.25:51820>
PresharedKey = <Optional Preshared Key>
PersistentKeepalive = <Optional Keepalive Interval in Seconds>
```

3) Save the tunnel configuration, then click “Activate” to start the WireGuard interface. The status will change to “Active” once the interface is successfully running.
4) Once the tunnel is active, verify that the connection has been successfully established by checking the “Latest handshake” status. In the tunnel details panel shown in Figure 10

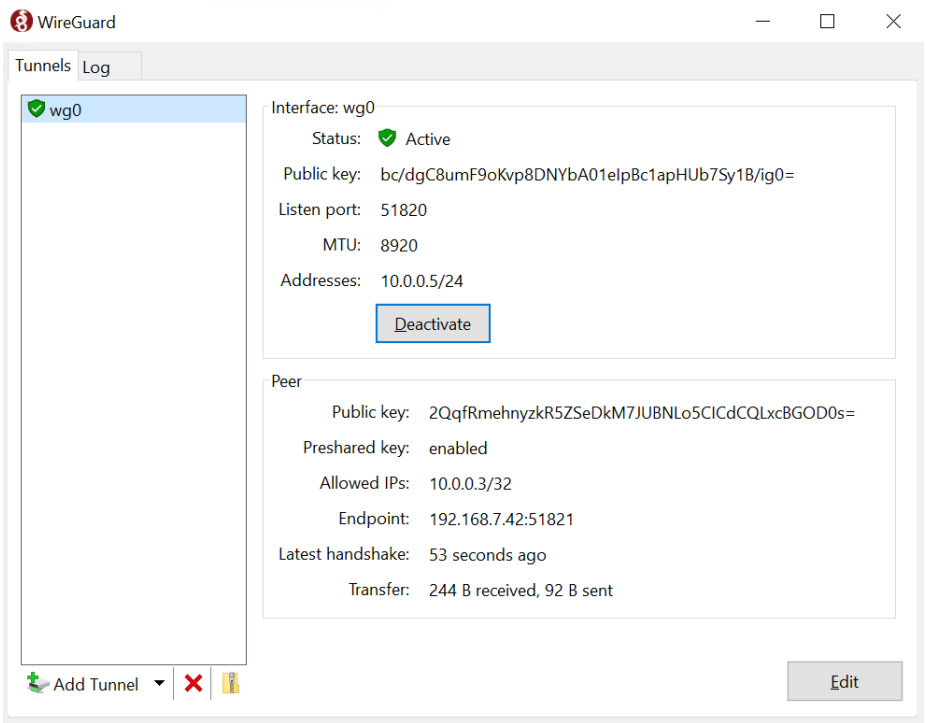

Figure 10 WireGuard Tunnel Handshake Successful

4 FPGA Development Board Setup

4.1 ZCU106 Board Setup

1) Make sure power switch is off and connect power supply to FPGA development board.
2) Power on the system.
3) Download configuration file and firmware to FPGA board by following steps:
   i) Open Vivado TCL shell.
   ii) Change current directory to download folder which includes demo configuration file.
   iii) Run bat script to program bit file, as shown in Figure 11.

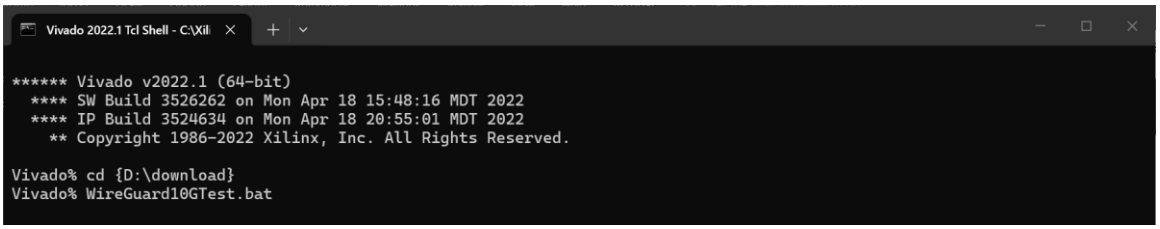

Figure 11 Program Device on ZCU106

4.2 KCU116 Board Setup

1) Make sure the power switch is off and connect the power supply to FPGA development board.
2) Connect USB cable between PC to JTAG micro USB port.
3) Power on the system.
4) Open Vivado Hardware Manager to program FPGA by following steps.
   i) Click open Hardware Manager.
   ii) Open target -> Auto Connect.
   iii) Select FPGA device to program bit file.
   iv) Click Program device.
   v) Click "…" to select program bit file.
   vi) Click Program button to start FPGA Programming.

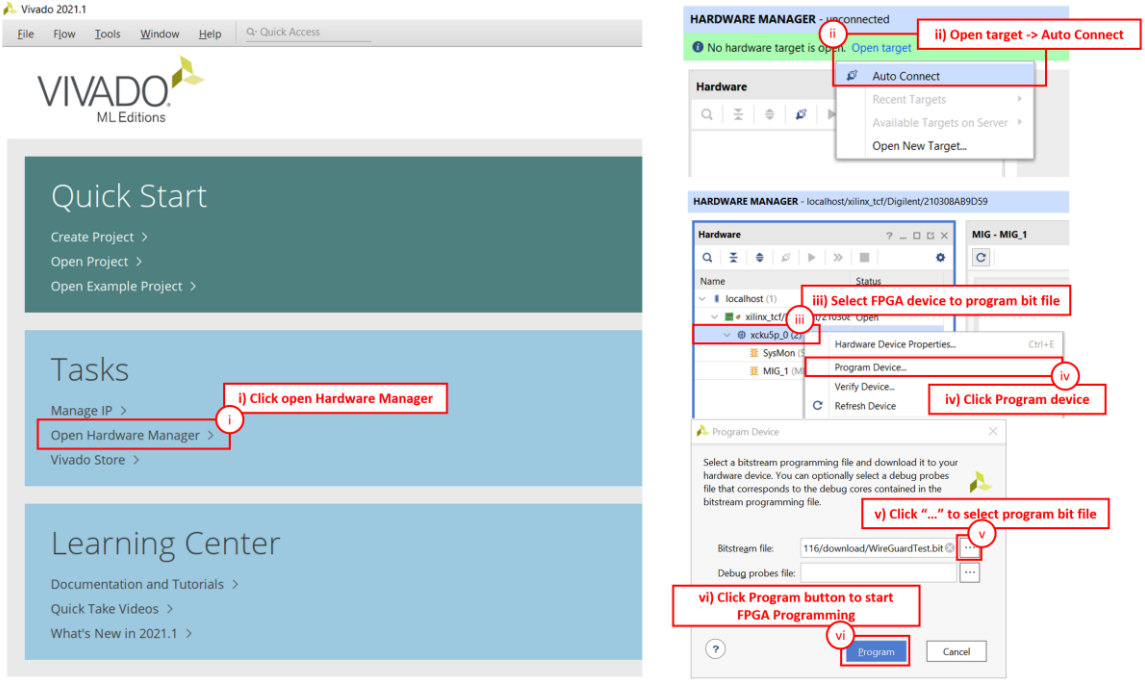

Figure 12 Program Device on KCU116

5 Command details

Users can set WireGuard10G-IP parameters by entering commands in the serial console. When the FPGA boots, the available commands and their usage are displayed, as shown in Figure 13. Detailed information about each command is described below

```
================================================================================
 INITIALIZING NETWORK INTERFACE
================================================================================
[BOOT] System starting...
[BOOT] Waiting for hardware stabilization:  [READY]
[SYSTEM] Checking Ethernet Link Status....
[SUCCESS] All Ethernet links are UP!

================================================================================
 WIREGUARD-IP CONFIGURATION INTERFACE | Version 0x80014440
================================================================================
[ID]  COMMAND                PARAMETERS           DESCRIPTION
--------------------------------------------------------------------------------
[ 0]  help                   [command]            Display this help menu or specific command details.
[ 1]  setdevIP               <dd.dd.dd.dd>[/dd]   Set FPGA's IP address in dotted-decimal format.
[ 2]  setdevport             <dd|dynamic>         Set FPGA's UDP port in decimal or dynamic format.
[ 3]  setdevMac              <hh-hh-hh-hh-hh-hh>  Set FPGA's MAC address in hexadecimal format.
[ 4]  setGatewayIP           <dd.dd.dd.dd>        Set Gateway IP address in dotted-decimal format.
[ 5]  setTAI64               <hhhh>               Set WireGuard TAI64 timer in hexadecimal format.
[ 6]  setPrivKey             <BASE64>             Set WireGuard FPGA's private key in BASE64 format.
[ 7]  setWGaddressIP         <dd.dd.dd.dd>        Set WireGuard local IP in dotted-decimal format.
[ 8]  setPeerPubKey          <BASE64>             Set WireGuard Peer's public key in BASE64 format.
[ 9]  setAllowsIP            <dd.dd.dd.dd/dd>...  Set WireGuard allowed IPs in dotted-decimal format.
[10]  setEndpointIP          <dd.dd.dd.dd>[:dddd] Set Remote's IP address in dotted-decimal format.
[11]  setPresharedKey        <BASE64>             Set WireGuard pre-shared key in BASE64 format.
[12]  setPersistentKeepalive <dd>                 Set WireGuard persistent keepalive in seconds.
[13]  showKey                <on|off>             Set ephemeral private key display mode.
[14]  WGActivate                                  Load all parameters and activate connections.
================================================================================

WG10G >>
```

Figure 13 Serial console

5.1 Set FPGA IP Address

> setdevIP <ddd.ddd.ddd.ddd>[/ddd]

This command sets the FPGA's IP address in dotted-decimal format, with an optional subnet mask in CIDR notation. The default subnet mask is /24. The default IP address is 192.168.7.42. This is the IP address of the physical Ethernet interface, not the WireGuard tunnel address.

5.2 Set FPGA Port Number

> setdevport <ddddd|dynamic>

This command is used to set the FPGA's UDP source port number. If a numeric value (0–65535) is provided, it is used as a fixed static port. If the argument is dynamic, a dynamic port starting from 51821 is automatically assigned and incremented for each new connection. The default mode is dynamic.

5.3 Set FPGA MAC Address

> setdevMac <hh-hh-hh-hh-hh-hh>

This command is used to set the FPGA's Ethernet MAC address in hexadecimal format, with octets separated by hyphens. The default FPGA MAC address is 80-11-22-33-44-55, which is a unicast MAC address.

5.4 Set Gateway IP Address

> setgatewayIP <ddd.ddd.ddd.ddd>

This command sets the network gateway IP address in dotted-decimal format. The default gateway IP address is 0.0.0.0.

5.5 Initialize TAI64 Timestamp

```
> setTAI64 <hhhhhhhhhhhhhhhh>
```

This command initializes the TAI64 timestamp used in the WireGuard handshake initiation message. The value must be a 16-character hexadecimal string representing the current TAI64 timestamp.

* TAI64, a timestamp format that represents time as the number of seconds since January 1, 1970 (TAI), encoded as a 64-bit big-endian integer.

5.6 Set FPGA Private Key

```
> setprivkey <BASE64>
```

This command sets the FPGA's WireGuard private key. The key must be a valid 32-byte X25519 key encoded in Base64 (44 characters ending with '=').

5.7 Set WireGuard Interface Address IP

```
> setWGaddressIP <ddd.ddd.ddd.ddd>
```

This command sets the IP address assigned to the FPGA's WireGuard IP address. This is the IP address used for communication within the VPN tunnel, separate from the physical Ethernet interface IP.

5.8 Set Peer Public Key

```
> setpeerpubkey <BASE64>
```

This command is used to set the FPGA's WireGuard Peer public key. The key must be the Base64-encoded X25519 public key that corresponds to the peer's private key configured in the WireGuard peer's configuration file.

5.9 Set Allowed IPs

```
> setallowsIP <ddd.ddd.ddd.ddd/ddd>...
```

This command sets the list of IP/CIDR pairs that are allowed to be routed through the WireGuard10G-IP (up to 16 entries, space-separated). Each entry may include a CIDR prefix; if omitted, /32 is used by default. Setting 0.0.0.0/0 routes all traffic through the IP.

5.10 Set Endpoint IP Address

```
> setendpointIP <ddd.ddd.ddd.ddd>[:ddddd]
```

This command sets the WireGuard peer's endpoint IP address and port number separated by a colon. If the port is omitted, the default value of 51820 is used. The default IP address is 192.168.7.25. The FPGA will initiate the WireGuard handshake to this endpoint.

5.11 Set Preshared Key

```
> setpresharedkey <BASE64>
```

This optional command sets a 32-byte symmetric pre-shared key (PSK) shared between the FPGA and Peer, encoded as a 44-character Base64 string.

5.12 Set Persistent Keepalive

```
> setpersistentkeepalive <ddddd>
```

The User sets the interval in seconds for persistent keepalive packets to ensure the bidirectional connection remains active through NAT or stateful firewalls. The User must provide a value within the valid 16-bit range, specifically from 0 to 65535 seconds. Setting the value to 0 allows the User to disable this feature, which is the default system state.

5.13 Show Ephemeral Private Key

```
> showkey <on|off>
```

This command enables or disables showkey mode, which controls whether the ephemeral private key is displayed on the serial console. Set the parameter to “on” to enable or “off” to disable. This is disabled by default.

5.14 Activate WireGuard

```
> WGActivate
```

This command applies all previously configured parameters and initiates the WireGuard handshake with the configured Peer endpoint. Once the handshake completes successfully, the IP core enters the monitoring loop and continuously reports status codes and transfer statistics on the serial console. User can press “Ctrl + C” to exit the monitoring loop and deactivate Wireguard.

6 Demo environment

This section describes the hardware and software environments used to demonstrate the WireGuard10G IP core. Two demo configurations are provided, each targeting a different use case.

The first, described in Section 6.1, uses the Serial Console to configure and activate the FPGA directly, demonstrating a point-to-point WireGuard tunnel between the User PC and an Endpoint PC.

The second, described in Section 6.2, uses the provided GUI application to configure the FPGA, demonstrating selective encapsulation where only traffic from a specific IP address is encrypted and tunneled to the Endpoint PC.

Both configurations share the same FPGA board and PC hardware, but differ in network topology and configuration method.

6.1 Point-to-Point Encapsulation using Serial Console

The testing environment for the Serial Console demo is illustrated in Figure 14. The setup consists of three main components: the User PC, the FPGA board running the WireGuard10G IP core, and the Endpoint PC.

The User PC connects directly to the FPGA over a 10 GbE link. Its network interface is configured with a WireGuard tunnel address of 10.0.0.3 and a gateway of 10.0.0.1 as shown in Figure 15. Since all WireGuard tunnel traffic is routed through this gateway, the ARP table on the User PC must be manually configured for the gateway IP 10.0.0.1. Additionally, one active Ethernet interface must be reserved to support the Kill Switch or multiple allowed IPs functionality.

The FPGA is assigned a Network IP address of 192.168.7.42 with a gateway of 192.168.7.1. For outgoing packets sourced from the tunnel IP, it encapsulates and encrypts them into WireGuard packets before forwarding. For incoming packets, it decrypts and decapsulates them before delivering to the User PC. All WireGuard tunnel traffic is routed through the gateway to reach the Endpoint PC.

On the other side, the Endpoint PC is reachable at 203.0.113.5 on the network and is assigned a WireGuard tunnel IP of 10.0.0.5. The tunnel is established between the FPGA and the Endpoint PC through the gateway.

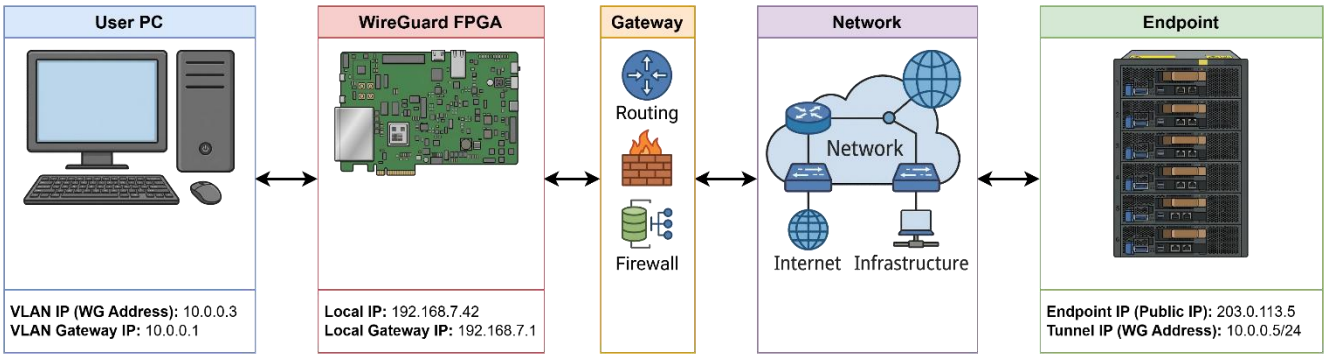

Figure 14 Serial Console testing environment

6.1.1 User PC Setup

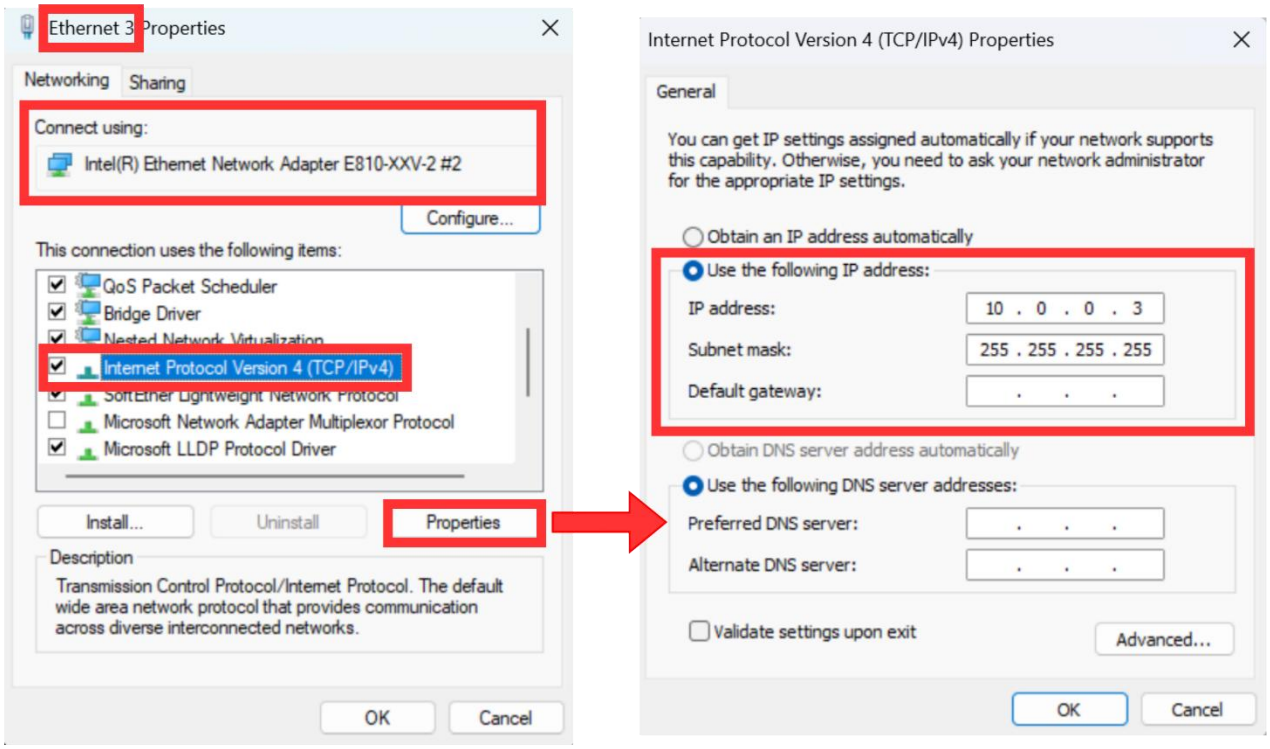

Figure 15 Setting IP address for PC

1) Open Local Area Connection Properties of 10 Gb connection, as shown in the left window of Figure 15.
2) Select “TCP/IPv4” and then click Properties.
3) Set IP address = 10.0.0.3 and Subnet mask = 255.255.255.0 as shown in the right window of Figure 15
4) Add a gateway IP route with highest priority matric using the command below:

```
> netsh interface ipv4 add route 10.0.0.0/24 "Ethernet 3" 10.0.0.1 metric=0
```

5) Open CMD as administrator and assign MAC address for gateway IP using the command below:

```
> netsh interface ipv4 add neighbor "Ethernet 3" 10.0.0.1 80-AB-CD-EF-12-AB
```

6.1.2 WireGuard Peer Configuration

Follow the steps in Section 3.2 to set up WireGuard software, using the configuration below:

```
[Interface]
PrivateKey = YKcfrhW7gOw47WJPEk1RobQPn+c9p0OURJYvZqvAz1g=
Address = 10.0.0.5/24
ListenPort = 51820
MTU = 8920

[Peer]
PublicKey = 2QqfRmehnyzkR5ZSeDkM7JUBNLo5CICdCQLxcBGOD0s=
PresharedKey = 4xZ9Lk2mQvR7wNpT3aBcYdHeJfUgXsOiK1nMqPlV8uE=
AllowedIPs = 10.0.0.3/32
```


6.1.3 WireGuard10G-IP Setup

On the serial console, use all commands below to activate WireGuard10G-IP:

```
WG10G >> setprivkey IEmb7vrj0k0c2x3X/hDlVABYmjm7I2XtwifwlAIFTUs=
Set FPGA's WG Private Key to IEmb7vrj0k0c2x3X/hDlVABYmjm7I2XtwifwlAIFTUs=

WG10G >> setpeerpubkey bc/dgC8umF9oKvp8DNYbA01eIpBc1apHUb7Sy1B/ig0=
Set WG Peer's Public Key to bc/dgC8umF9oKvp8DNYbA01eIpBc1apHUb7Sy1B/ig0=

WG10G >> setpresharedkey 4xZ9Lk2mQvR7wNpT3aBcYdHeJfUgXsOiK1nMqPlV8uE=
Set WG Pre-Shared Key to 4xZ9Lk2mQvR7wNpT3aBcYdHeJfUgXsOiK1nMqPlV8uE=

WG10G >> setdevIP 192.168.7.42
Set FPGA's IP Address to 192.168.7.42/24

WG10G >> setdevport 51821
Set Port number to 51821

WG10G >> setgatewayIP 192.168.7.1
Set Gateway IP Address to 192.168.7.1

WG10G >> setendpointIP 203.0.113.5
Set Endpoint IP Address to 203.0.113.5:51820

WG10G >> setdevmac 98-B7-85-A8-0E-CF
Set FPGA's MAC Address to 98-B7-85-A8-0E-CF

WG10G >> setTAI64 4000000069afdbe1
initial wg TAI64 timer to 4000000069AFDBE1

WG10G >> setWGaddressIP 10.0.0.3
Set WireGuard IP Address to 10.0.0.3

WG10G >> setallowsIP 10.0.0.5/24
Allowed IP 4: 10.0.0.5/24
Total allowed IPs configured: 1

WG10G >> setpersistentkeepalive 25
set wg persistent keepalive to 25 sec
```

Figure 16 Example commands for WireGuard10G-IP on serial console

6.1.4 Verify Connection

After completing the setup, the first WireGuard connection should be established. The WireGuard10G-IP status is displayed on the serial console. Once data is transmitted or received through the tunnel, the transfer and receive sizes are continuously updated on the serial console, as shown in Figure 17.

```
WG10G >> wgactivate
Activate WireGuard
================================================
WireGuard Parameters
FPGA's WG Public Key:  2QqfRmehnyzkR5ZSeDkM7JUBNLo5CICdCQLxcBGOD0s=
Peer WG Public Key:    bc/dgC8umF9oKvp8DNYbA01eIpBc1apHUb7Sy1B/ig0=
Persistent Keepalive: 25

WireGuard Virtual Network
User Address IP: 10.0.0.3
Peer Allowed IP: 10.0.0.0/24

Network Parameters
FPGA's IP Address:   192.168.7.42:51821
Subnet Mask:         255.255.255.0
Gateway IP Address:  192.168.7.1
Endpoint IP Address: 203.0.113.5:51820
FPGA's Mac Address:  98-B7-85-A8-0E-CF
Target Mac Address:  64-9D-99-D0-9E-EE
================================================
WireGuard Log
Status: 0x0031 - Handshake initial successfully generated
Status: 0x0051 - Key pair processed successfully (e.g., new TX/RX keys established)
Status: 0x0011 - Handshake response processed successfully
Press Ctrl+C to exit ...
```

Figure 17 Activated WireGuard10G-IP serial console

```
C:\>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Figure 18 Verifying network connectivity using the ping command

6.2 Selective Encapsulation using GUI

The testing environment for the GUI demo is illustrated in Figure 19, consisting of three main components: the User PC, the FPGA board running the WireGuard10G-IP core, and the Endpoint PC.

The User PC connects directly to the FPGA over a 10 GbE link. Its single network interface is configured with two addresses. The local address 192.168.7.42 with gateway 192.168.7.1 is used for normal network traffic. The WireGuard tunnel address 10.0.0.3 with gateway 10.0.0.1 is used for traffic that should enter the tunnel. Since all WireGuard tunnel traffic is routed through the gateway, no manual ARP entries are required for individual WireGuard destinations.

The FPGA shares the same IP and MAC address as the User PC's, allowing it to transparently intercept traffic. For outgoing packets sourced from the tunnel IP, it encapsulates and encrypts them into WireGuard packets before forwarding. For incoming packets, it decrypts and decapsulates them before delivering to the User PC.

On the other side, the Endpoint PC is reachable at 203.0.113.5 on the network and is assigned a WireGuard tunnel IP of 10.0.0.5. All encrypted WireGuard traffic is routed through the gateway to reach it.

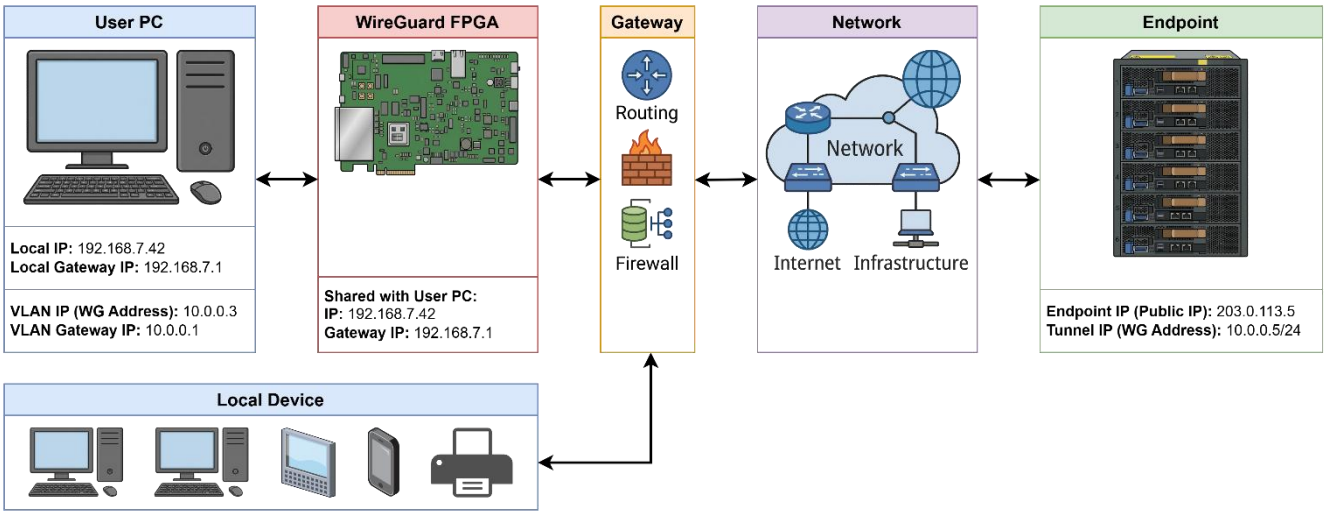

Figure 19 GUI testing environment

6.2.1 Graphical User Interface (GUI)

To simplify the use of the WireGuard10G IP core, a GUI application is provided as WireGuard-Demo-GUI.exe. The GUI allows users to configure the tunnel, connect to the FPGA over a serial port, activate or deactivate the WireGuard tunnel, and monitor real-time transfer statistics.

The GUI window is divided into two tabs: the Tunnel tab (for configuration and status monitoring) and the Log tab (for serial console interaction). The main window is shown in Figure 20.

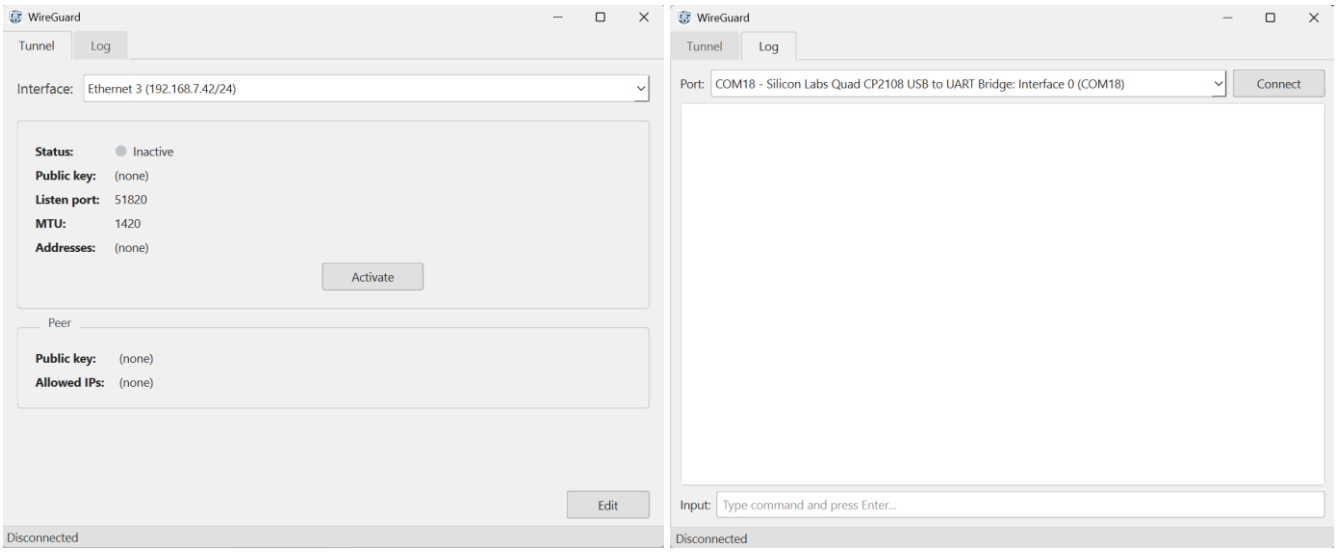

Figure 20 WireGuard Demo GUI window

Note: The WireGuard Demo GUI requires Administrator privileges to modify network interface settings (add IP addresses, configure routes, and manage the Kill Switch via Windows Filtering Platform).

6.2.2 Connect with FPGA device

To establish a connection with the FPGA:

1) Click the COM port drop-down to see available serial ports.
2) Select the COM port corresponding to the FPGA's UART interface.
3) Click "Connect". The status bar at the bottom left of the window will display "Connected" when the connection is established, as shown in Figure 21.
4) Once connected, the button label changes to "Disconnect". Users can send manual commands via the Input field at the bottom of the Log tab.

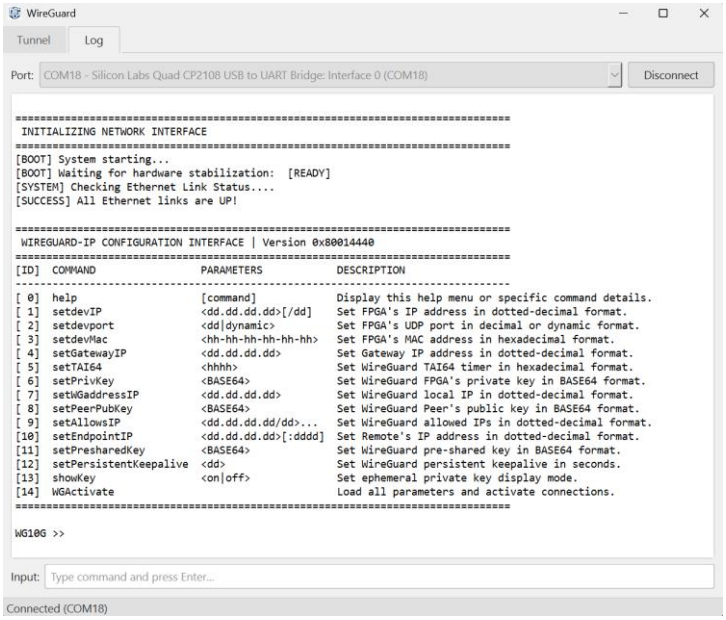

Figure 21 GUI log tab

6.2.3 User PC Setup

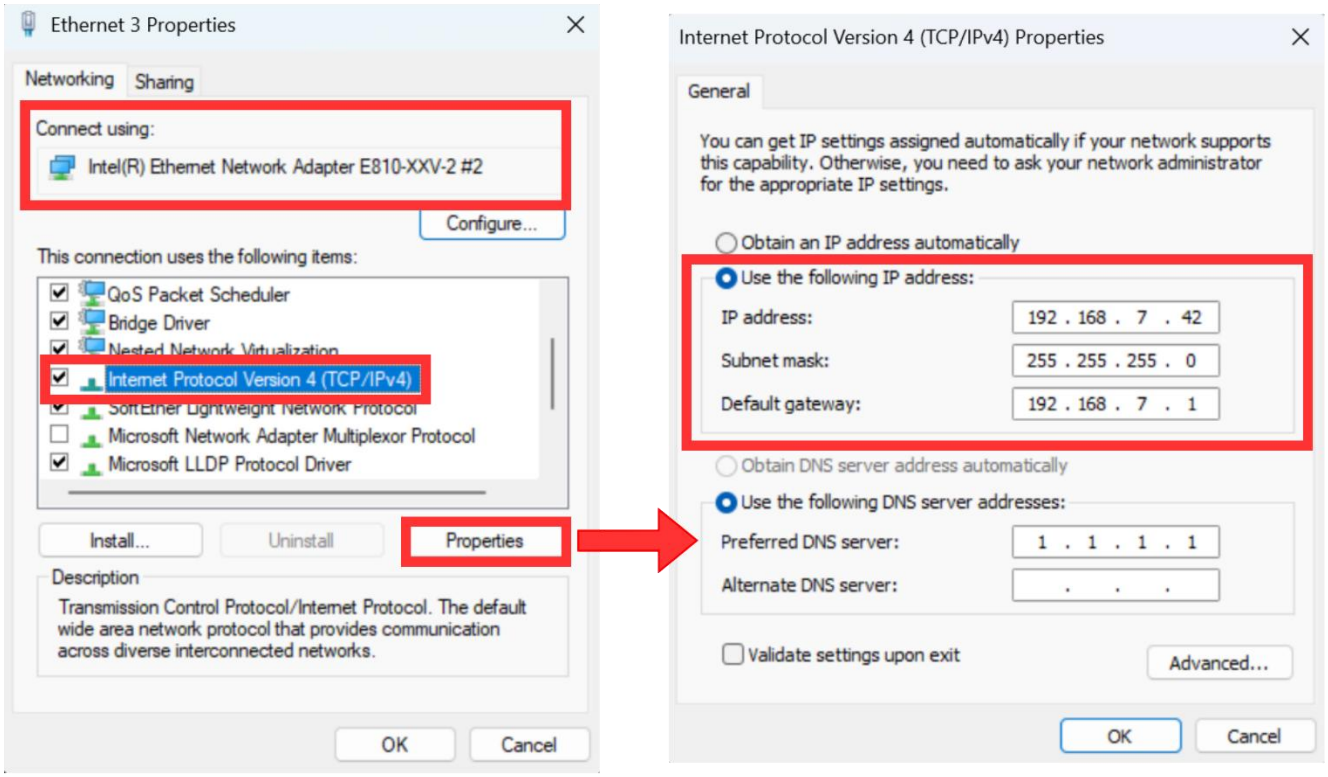

Figure 22 Setting IP address for PC

6) Open Local Area Connection Properties of 10 Gb connection, as shown in the left window of Figure 22.
7) Select “TCP/IPv4” and then click Properties.
8) Set IP address = 192.168.7.42, Subnet mask = 255.255.255.0 and Default gateway = 192.168.7.1 as shown in the right window of Figure 22

6.2.4 WireGuard Peer Configuration

Follow the steps in Section 3.2 to set up WireGuard software, using the configuration below:

```
[Interface]
PrivateKey = YKcfrhW7gOw47WJPEk1RobQPn+c9p0OURJYvZqvAz1g=
ListenPort = 51820
Address = 10.0.0.5/32, 10.90.1.6/32
MTU = 8920
[Peer]
PublicKey = 2QqfRmehnyzkR5ZSeDkM7JUBNLo5CICdCQLxcBGOD0s=
AllowedIPs = 10.0.0.3/32
```


6.2.5 Edit Tunnel Configuration

To configure the WireGuard tunnel parameters, click the “Edit” button on the Tunnel tab. The Edit Tunnel dialog will open, as shown in Figure 23.

- The Public key field at the top is read-only. It is automatically computed from the PrivateKey field.
- The configuration text editor accepts the standard WireGuard INI configuration format. The supported fields are described in Table 1.
- Click “Import” to load an existing WireGuard configuration file (.conf).
- Click “Save” to validate and save the configuration. The configuration is written to tun.conf in the application directory.

Table 1 Supported configuration fields

Field	Section	Description
PrivateKey	[Interface]	FPGA’s X25519 private key (Base64, 44 chars).
ListenPort	[Interface]	FPGA’s UDP listen port. Default: 51820.
Address	[Interface]	WireGuard interface IP.
MTU	[Interface]	Optional physical MTU (1–8920). Default: 1420.
PublicKey	[Peer]	WireGuard peer’s public key (Base64, 44 chars). Required.
AllowedIPs	[Peer]	Subnets routed through the tunnel in CIDR notation. Accepts up to 16 entries separated by space. Use 0.0.0.0/0 to route all traffic through the tunnel.
Endpoint	[Peer]	Peer endpoint in IP:Port format.
PersistentKeepalive	[Peer]	Optional keepalive interval in seconds (0 = disabled). Default: 0.
PresharedKey	[Peer]	Optional symmetric preshared key (Base64, 44 chars).

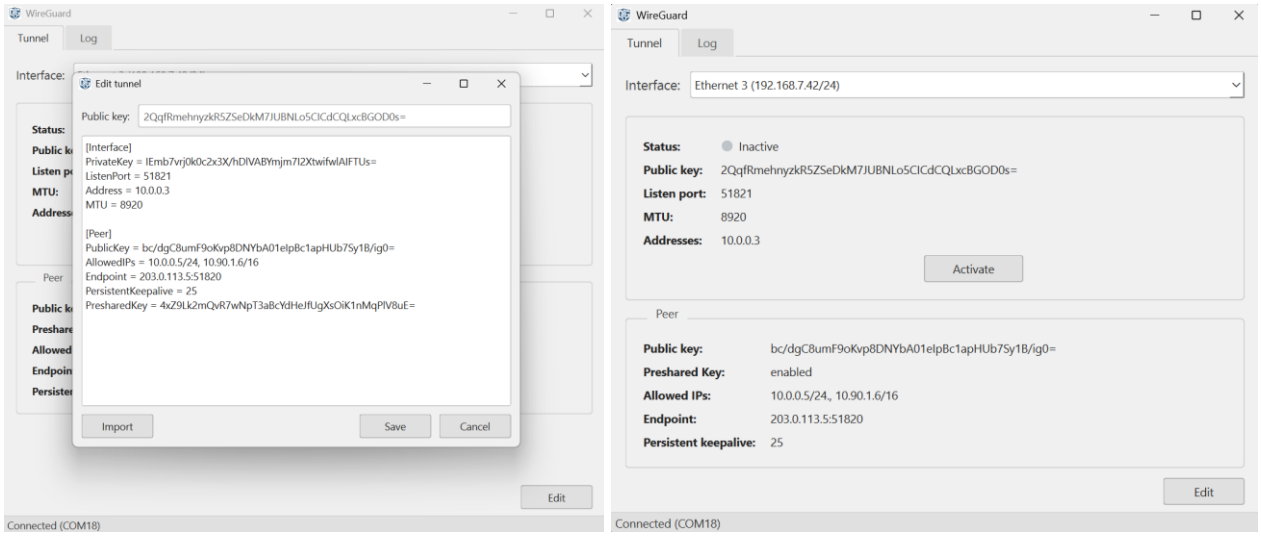

Figure 23 Parameters config

6.2.6 Activate Tunnel

When the user clicks “Activate”, the GUI automatically performs all configuration steps on both the FPGA (via the serial port) and the User PC (via CLI). The following steps detail each action taken and the corresponding commands executed on the PC using the example configuration from Figure 23.

1) Resolve endpoint MAC address.
2) Set new MTU.

```
> netsh interface ipv4 set subinterface “Ethernet 3” mtu=8920 store=active
```

3) Add the tunnel IP address to the network adapter.

```
> netsh interface ipv4 add address “Ethernet 3” 10.0.0.3 255.255.255.255
```

4) Add routes for all Allowed IPs through the tunnel, using 10.0.0.1 as the next-hop (derived from the tunnel Address 10.0.0.3 by replacing the last octet with 1).

```
> netsh interface ipv4 add route 10.0.0.0/24 “Ethernet 3” 10.0.0.1 metric=0
> netsh interface ipv4 add route 10.90.0.0/16 “Ethernet 3” 10.0.0.1 metric=0
```

5) Assign endpoint MAC address to next-hop IP.

```
> netsh interface ipv4 add neighbor “Ethernet 3” 10.0.0.1 98-B7-85-A8-0E-CE
```

6) Send configuration commands set from Figure 23 to the FPGA via serial console.
7) Send the “WGActivate” command to the FPGA to apply all parameters and initiate the handshake. The log serial console will enter monitor mode and print status codes and transfer statistics.

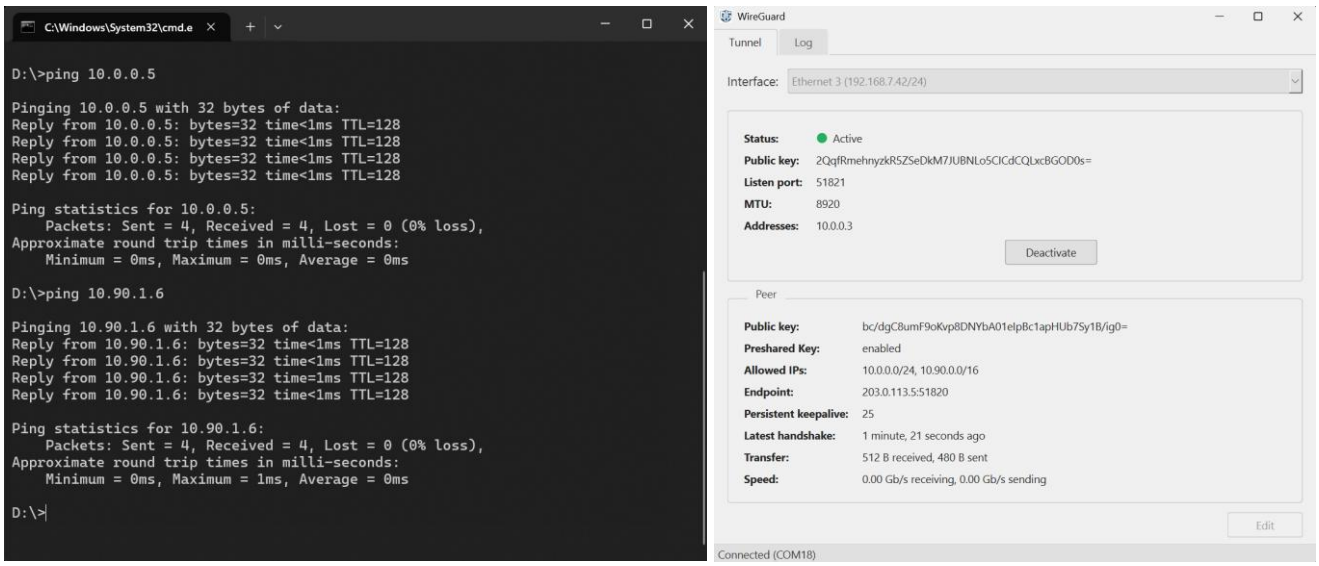

Figure 24 WireGuard tunnel activated

When the tunnel is active, the Tunnel tab in the GUI displays the following live statistics:

- Latest handshake: time elapsed since the last successful WireGuard handshake.
- Transfer: cumulative bytes received and sent through the tunnel.
- Speed: current receive and transmit speeds in Gb/s

To verify functional connectivity through the tunnel, users can use standard network tools such as ping (e.g., ping 10.0.0.5) to confirm end-to-end reachability, or any application that generates traffic over the tunnel IP.

6.2.7 Deactivate Tunnel

When the user clicks “Deactivate”, the GUI reverses all configuration steps applied during activation in Section 6.2.6. The following steps detail each action taken and the corresponding commands executed on the PC.

1) Send “Ctrl + C” to the FPGA via serial console to terminate the WireGuard session.
2) Restore the original MTU of the network adapter.

```
> netsh interface ipv4 set subinterface “Ethernet 3” mtu=9000 store=active
```

3) Remove the tunnel IP address from the network adapter.

```
> netsh interface ipv4 delete address “Ethernet 3” 10.0.0.3
```

4) Remove the routes added for all Allowed IPs.

```
> netsh interface ipv4 delete route 10.0.0.0/24 “Ethernet 3” 10.0.0.1
> netsh interface ipv4 delete route 10.90.0.0/16 “Ethernet 3” 10.0.0.1
```

5) Remove the endpoint ARP neighbor entry.

```
> netsh interface ipv4 delete neighbor “Ethernet 3” 10.0.0.1
```

6) Once deactivated status is received from serial console, the Tunnel tab returns to its inactive state and the live statistics are cleared from the display.

7 Performance Test

This section evaluates the throughput performance of the WireGuard10G IP core using iperf3. Two scenarios are measured and compared: both PCs running WireGuard software directly, and the WireGuard10G-IP environment described in Section 6.2

7.1 PC to PC (WireGuard Software)

In this scenario, both the User PC and the Endpoint PC run the WireGuard software. iperf3 traffic is sent between them over the software-based WireGuard tunnel, as configured in Figure 25.

On the Endpoint PC, run:

```
iperf3 -s
```

On the User PC, run:

```
iperf3 -c 10.0.0.5 --bidir -P 16 -w 512k
```

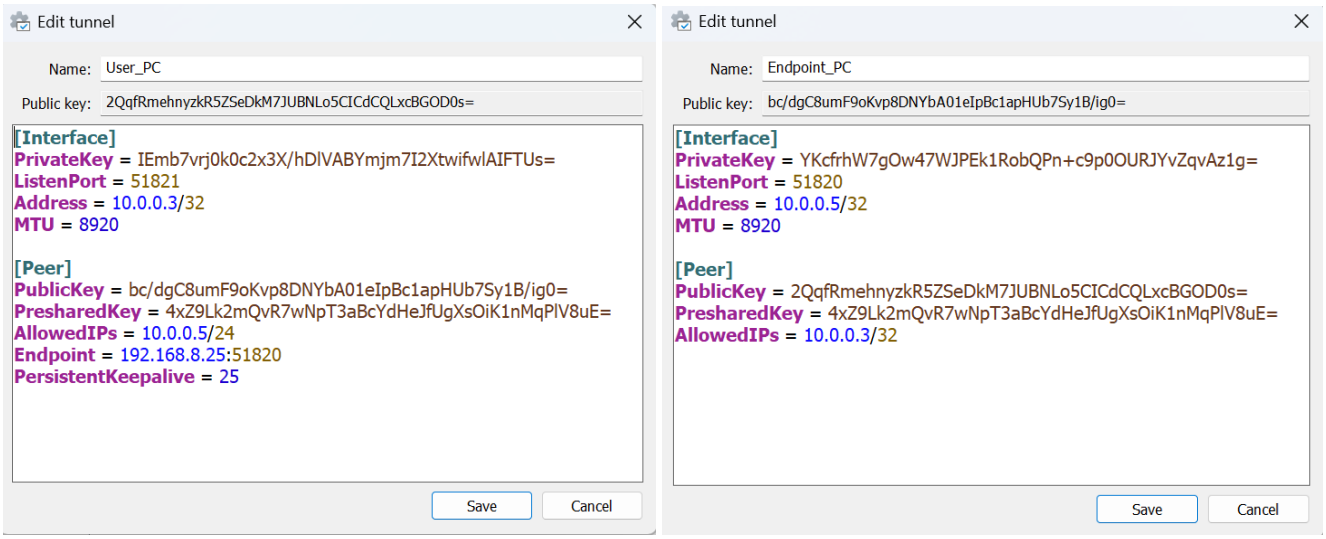

Figure 25 WireGuard software configuration

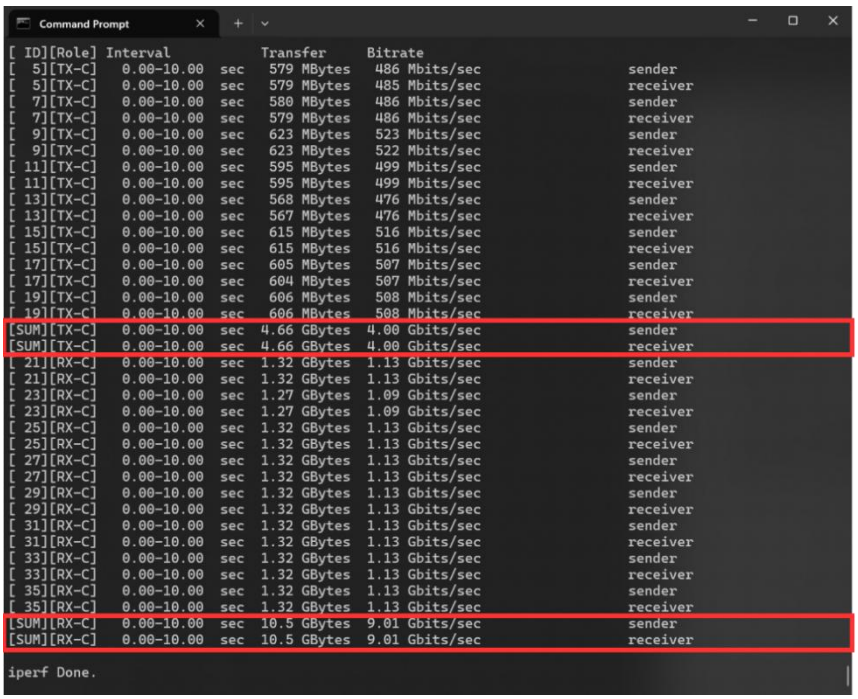

Figure 26 User PC iperf3 result

7.2 PC to PC via WireGuard10G-IP

In this scenario, the environment described in Section 6.2. is used

On the Endpoint PC, run:

```
iperf3 -s
```

On the User PC, run:

```
iperf3 -c 10.0.0.5 --bidir -P 16 -w 512k
```

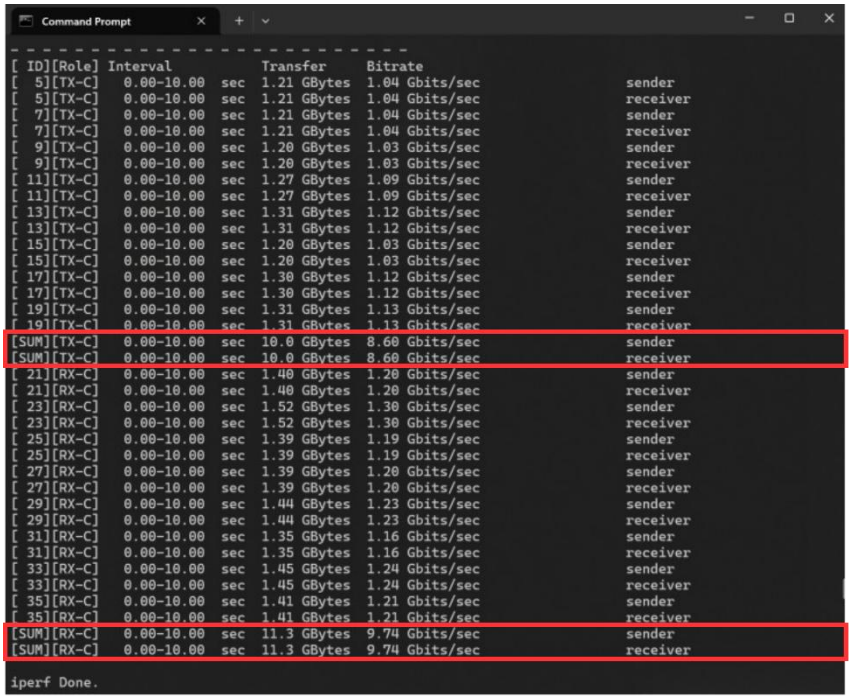

Figure 27 WireGuard10G-IP iperf3 result

7.3 Result Summary

As shown in Figure 26 and Figure 27, the WireGuard10G-IP core achieves significantly higher throughput compared to the WireGuard software implementation, demonstrating the performance advantage of offloading WireGuard encryption to dedicated hardware at 10 Gbps line rate.

8 Revision History

Revision	Date (D-M-Y)	Description
1.00	8-May-26	Initial version release

