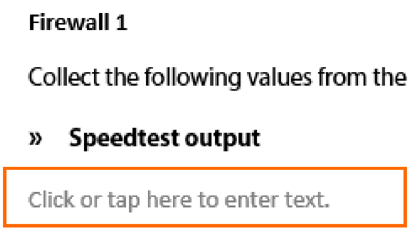
Best Practice - VPN Performance Testing

https://campus.barracuda.com/doc/79462716/

The following VPN performance test method provides a guideline for creating a standardized VPN performance testing environment required by Barracuda Networks Technical Support that allows you to identify potential configuration improvements. Please note that the VPN throughput results can differ from the values published on the datasheet of CloudGen Firewall F models due to varying test methods and equipment used.

Before You Begin

- To collect all relevant information for Barracuda Networks Technical Support, download the Word template form and fill in the required values as you go through the steps. Paste the output from each step into the text form and save the result. Include the completed form when contacting Barracuda Networks Support.



Firewall 1

Collect the following values from the output:

» **Speedtest output**

Click or tap here to enter text.

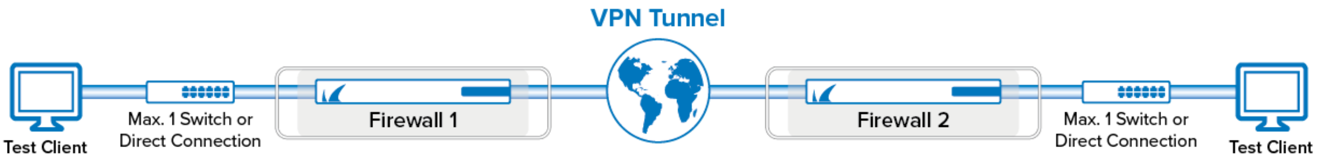
- Make sure that clients driving the test are powerful enough. iperf3 is especially critical in that respect because it is inherently single-threaded.
- Download VPN Performance Testing Form (Microsoft Word).

VPN Performance Test Setup

Before You Begin

To rule out devices in the local and remote networks, as well as side effects of other services on the firewall, create the following setup:

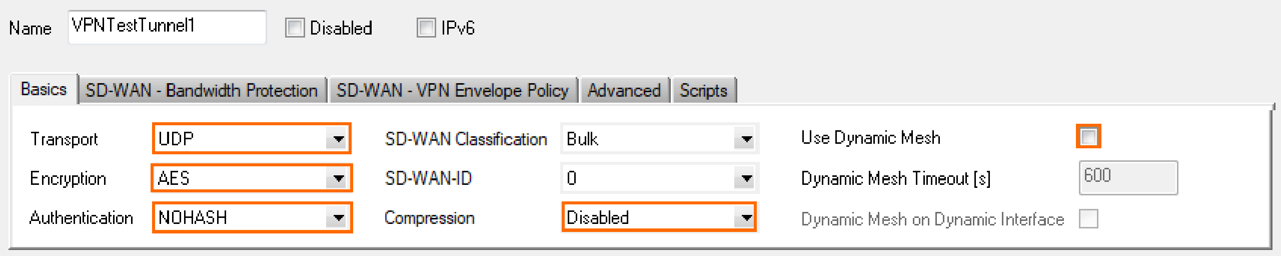
1. Testing must be performed on dedicated clients on both ends. Do not run performance tests directly on the shell of the firewall unless specifically stated otherwise.
2. Connect the test clients directly to the firewall. If that is not possible, up to one switch between the firewall and the test client is acceptable.
3. Do not use wireless network connections to connect to the firewall.


VPN Tunnel
Test Client
Max. 1 Switch or Direct Connection
Firewall 1
Firewall 2
Max. 1 Switch or Direct Connection
Test Client

Step 1. VPN Tunnel Configuration

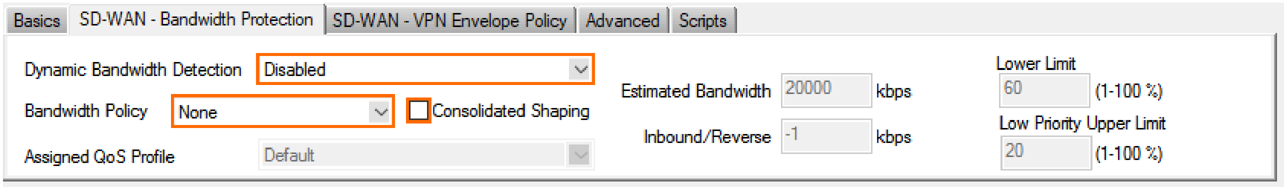
Configure the VPN tunnel on both firewalls with the following settings:

1. Create a TINA site-to-site VPN tunnel with a single transport with the following encryption settings:
2. In the **Basics** tab, configure the following settings:
    - **Transport** – Select **UDP**.
    - **Encryption** – Select **AES 128**.
    - **Authentication** – Select **NOHASH**.
    - **Compression** – Select **Disabled**.
    - **Use Dynamic Mesh** – Clear check box.



Name VPNTestTunnel1 ☐ Disabled ☐ IPv6
Transport UDP
Encryption AES
Authentication NOHASH
SD-WAN Classification Bulk
SD-WAN-ID 0
Compression Disabled
Use Dynamic Mesh ☐
Dynamic Mesh Timeout [s] 600
Dynamic Mesh on Dynamic Interface ☐

3. In the **SD-WAN Bandwidth Protection** tab, configure the following settings:
    - **Dynamic Bandwidth Detection** – Select **Disabled**.
    - **Bandwidth Policy** – Select **None**.
    - **Consolidated Shaping** – Clear the check box.

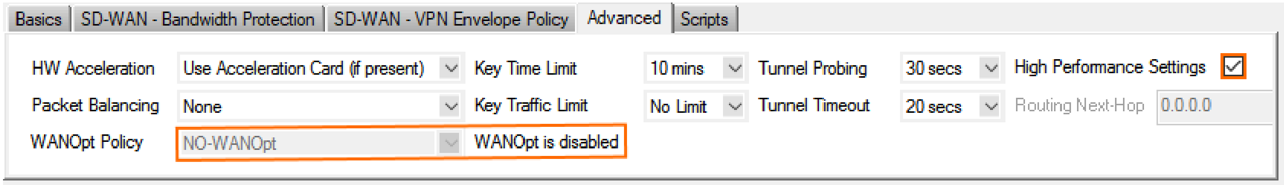


Dynamic Bandwidth Detection Disabled
Bandwidth Policy None ☐ Consolidated Shaping
Assigned QoS Profile Default
Estimated Bandwidth 20000 kbps
Inbound/Reverse -1 kbps
Lower Limit 60 (1-100 %)
Low Priority Upper Limit 20 (1-100 %)

4. In the **Advanced** tab, configure the following settings:
    - **High Performance Settings** (optional) – Select the check box.

      > This setting can be applied only to tunnels that bind to port 691 (UDP) on both sides. Ensure also that the local tunnel IP address is listed in **Service Properties > Service IPs**.
      > When running on VMWare with the vmxnet3 driver, the **High Performance Settings** have no impact.

    - **WANOpt Policy** – Select **NO-WANOpt**.



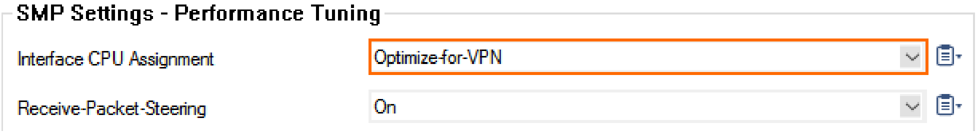
HW Acceleration Use Acceleration Card (if present)
Packet Balancing None
WANOpt Policy NO-WANOpt WANOpt is disabled
Key Time Limit 10 mins
Key Traffic Limit No Limit
Tunnel Probing 30 secs
Tunnel Timeout 20 secs
High Performance Settings ☑
Routing Next-Hop 0.0.0.0

5. Configure the **Local** and **Remote Networks.**
6. Configure the **Local** and **Remote** IP addresses used by the VPN endpoint.

For more information, see How to Create a TINA VPN Tunnel between CloudGen Firewalls.

Step 2. Optimize Performance Settings for VPN

1. Go to **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > System Settings**.
2. From the **Interface CPU Assignment** drop-down list, select **Optimize-for-VPN**.



SMP Settings - Performance Tuning

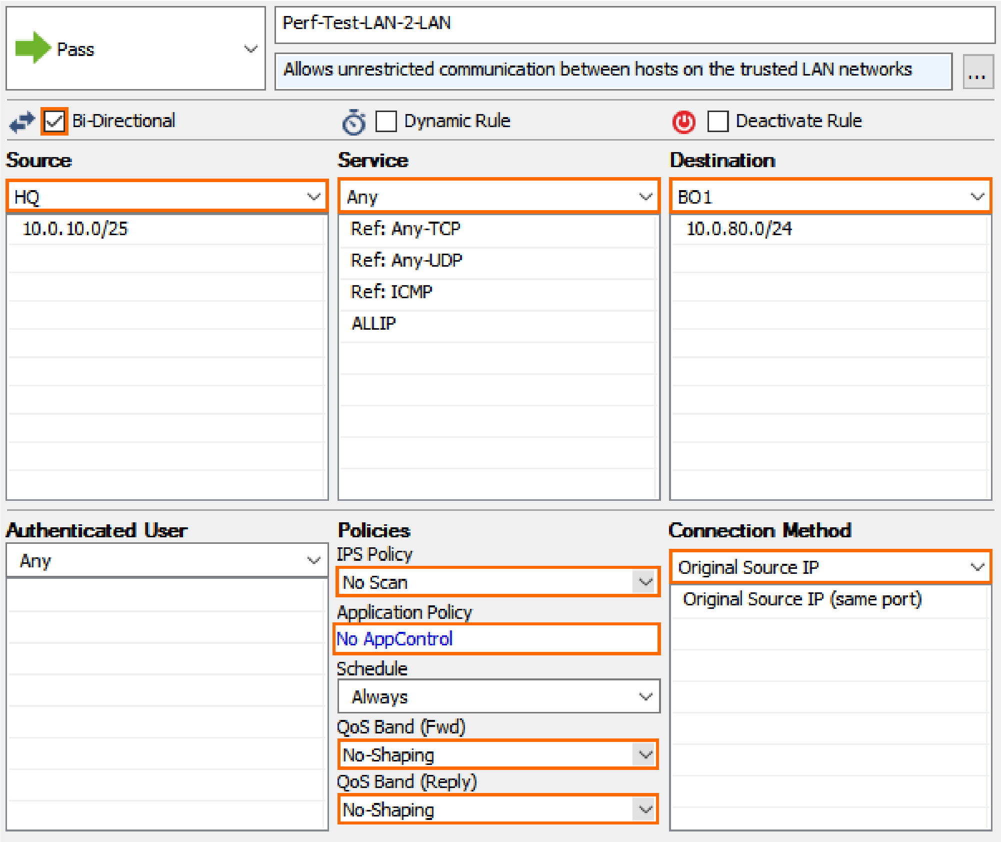
Interface CPU Assignment: Optimize-for-VPN

Receive-Packet-Steering: On

Step 3. Access Rule Configuration

On both firewalls, configure access rules to allow the test traffic through the VPN tunnel.

1. Create a **Pass** access rule:
   - **Source** – Select the IP address of the local test client.
   - **Service** – Select **Any**.
   - **Destination** – Select the IP address of the remote test client.
   - **Bi-Directional** – Select the check box.
   - **IPS Policy** – Select **No Scan**.
   - **Application Policy** – Select **No AppControl**.
   - **QoS Band (Fwd)** – Select **No-Shaping**.
   - **QoS Band (Reply)** – Select **No-Shaping**.



Pass

Perf-Test-LAN-2-LAN

Allows unrestricted communication between hosts on the trusted LAN networks

☑ Bi-Directional ☐ Dynamic Rule ☐ Deactivate Rule

Source: HQ
10.0.10.0/25

Service: Any
Ref: Any-TCP
Ref: Any-UDP
Ref: ICMP
ALLIP

Destination: BO1
10.0.80.0/24

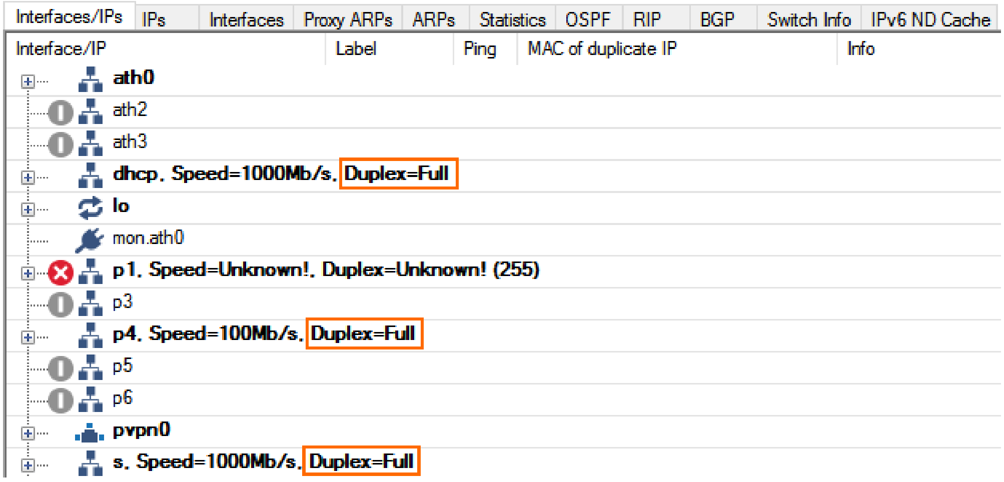
Authenticated User: Any

Policies
IPS Policy: No Scan
Application Policy: No AppControl
Schedule: Always
QoS Band (Fwd): No-Shaping
QoS Band (Reply): No-Shaping

Connection Method: Original Source IP
Original Source IP (same port)

Step 4. Verify the Network Interface Is Using Full Duplex

On the **CONTROL > Network** page of both firewalls: Verify that the network interfaces used by the test client and the VPN tunnel are using full duplex:



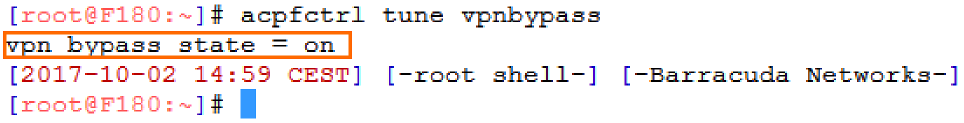
ath0
ath2
ath3
dhcp, Speed=1000Mb/s, Duplex=Full
lo
mon.ath0
p1, Speed=Unknown!, Duplex=Unknown! (255)
p3
p4, Speed=100Mb/s, Duplex=Full
p5
p6
pvpn0
s, Speed=1000Mb/s, Duplex=Full

Step 5. (Multi-Core Firewalls Only) Verify VPN Bypass Is Enabled

VPN bypass is a performance optimization for the VPN device queues on multi-core firewalls. The VPN bypass must be enabled.

1. Log into the firewall via SSH.
2. Enter the following command to check the VPN bypass state:

```
acpfctrl tune vpnbypass
```



```
[root@F180:~]# acpfctrl tune vpnbypass
vpn bypass state = on
[2017-10-02 14:59 CEST] [-root shell-] [-Barracuda Networks-]
[root@F180:~]#
```

3. If it is disabled, enable it with the following command:

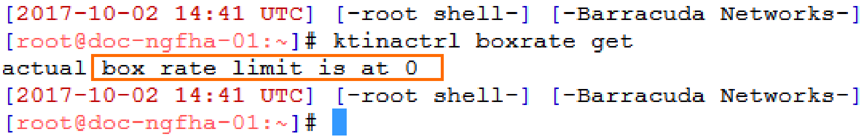
```
acpfctrl tune vpnbypass on
```

Step 6. Verify that the VPN Rate Limit Is Disabled

If a VPN rate limit is set, the VPN throughput is automatically decreased to the configured value.

1. Log into the firewall via SSH.
2. Enter the following command to check the VPN rate limit:

```
ktinactrl boxrate get
```



```
[2017-10-02 14:41 UTC] [-root shell-] [-Barracuda Networks-]
[root@doc-ngfha-01:~]# ktinactrl boxrate get
actual box rate limit is at 0
[2017-10-02 14:41 UTC] [-root shell-] [-Barracuda Networks-]
[root@doc-ngfha-01:~]#
```

If needed, the VPN rate limit can be configured in the **Operational VPN** settings on the **General Firewall Configuration** page.

Performance Testing

Perform the following tests to gather performance data for the VPN tunnel and the ISP connection.

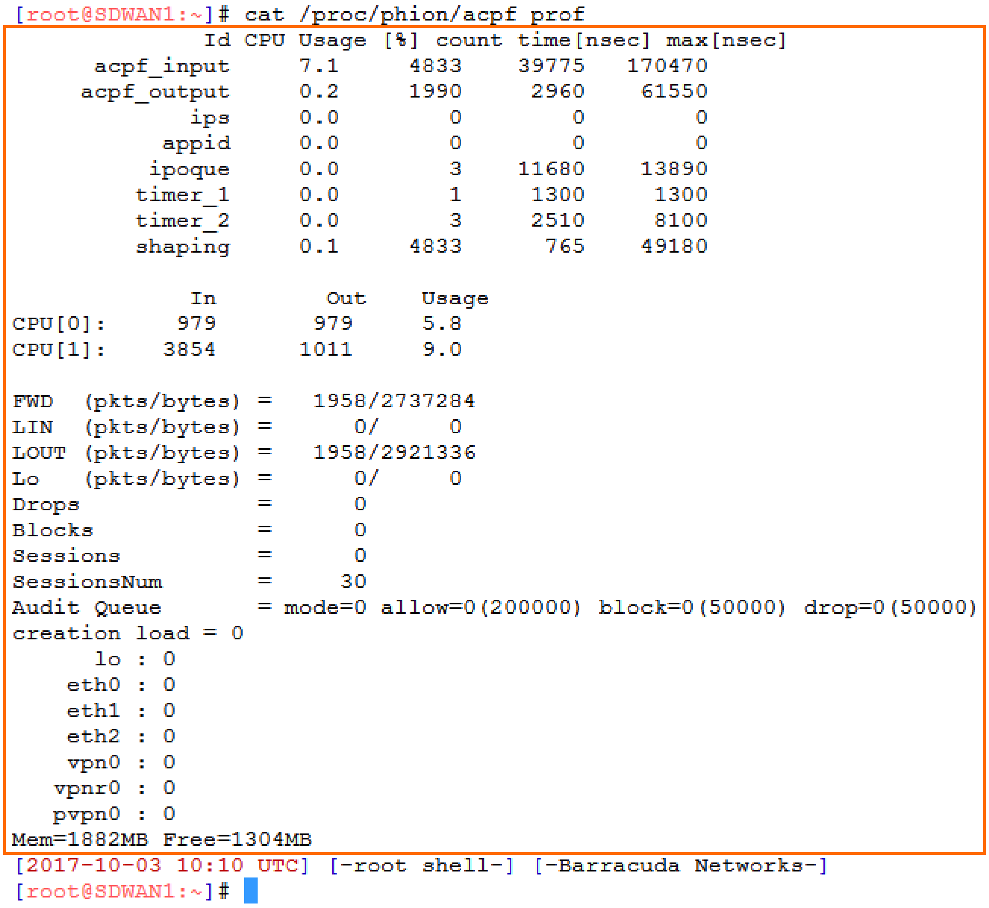
- ISP link speed
- VPN tunnel throughput
- Latency to the other firewall
- (virtual only) Cryptographic hardware performance test

Collect System Information

Collect the following information from both firewalls.

1. Log into the firewall via SSH.
2. Copy the performance output for the firewall kernel module:

   ```
   cat /proc/phion/acpf_prof
   ```



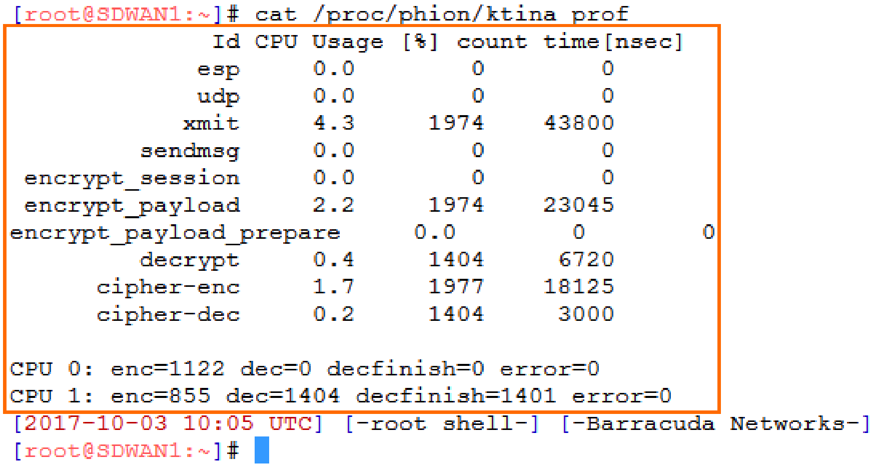
```
[root@SDWAN1:~]# cat /proc/phion/acpf_prof
                 Id CPU Usage [%] count time[nsec] max[nsec]
        acpf_input     7.1     4833    39775    170470
       acpf_output     0.2     1990     2960     61550
               ips     0.0        0        0         0
             appid     0.0        0        0         0
            ipoque     0.0        3    11680     13890
           timer_1     0.0        1     1300      1300
           timer_2     0.0        3     2510      8100
           shaping     0.1     4833      765     49180

              In         Out      Usage
CPU[0]:      979         979       5.8
CPU[1]:     3854        1011       9.0

FWD  (pkts/bytes) =   1958/2737284
LIN  (pkts/bytes) =      0/      0
LOUT (pkts/bytes) =   1958/2921336
Lo   (pkts/bytes) =      0/      0
Drops             =      0
Blocks            =      0
Sessions          =      0
SessionsNum       =     30
Audit Queue       = mode=0 allow=0(200000) block=0(50000) drop=0(50000)
creation load = 0
      lo : 0
    eth0 : 0
    eth1 : 0
    eth2 : 0
    vpn0 : 0
   vpnr0 : 0
   pvpn0 : 0
Mem=1882MB Free=1304MB
[2017-10-03 10:10 UTC] [-root shell-] [-Barracuda Networks-]
[root@SDWAN1:~]#
```

3. Copy the output from the VPN kernel module:

   ```
   cat /proc/phion/ktina_prof
   ```



```
[root@SDWAN1:~]# cat /proc/phion/ktina_prof
                   Id CPU Usage [%]  count  time[nsec]
                  esp      0.0        0        0
                  udp      0.0        0        0
                 xmit      4.3     1974    43800
              sendmsg      0.0        0        0
      encrypt_session      0.0        0        0
      encrypt_payload      2.2     1974    23045
encrypt_payload_prepare      0.0        0         0
              decrypt      0.4     1404     6720
           cipher-enc      1.7     1977    18125
           cipher-dec      0.2     1404     3000

CPU 0: enc=1122 dec=0 decfinish=0 error=0
CPU 1: enc=855 dec=1404 decfinish=1401 error=0
[2017-10-03 10:05 UTC] [-root shell-] [-Barracuda Networks-]
[root@SDWAN1:~]#
```

4. Collect the following values from the output:
   - **acpf_prof output**

- ◦ **ktina_prof output**

Check SMP

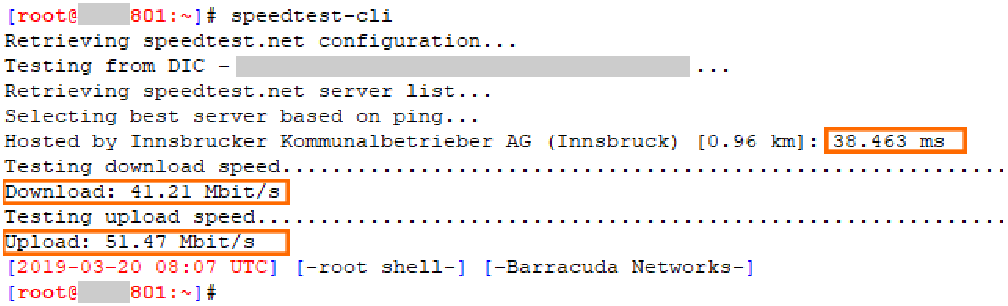
Execute setsmp and collect the output.

1. Log into the firewall via SSH.
2. Run command: `setsmp`
3. Collect the output.

Check ISP Link Speed

Execute the speed test on both firewalls.

1. Log into the firewall via SSH.
2. Run command: `speedtest-cli`

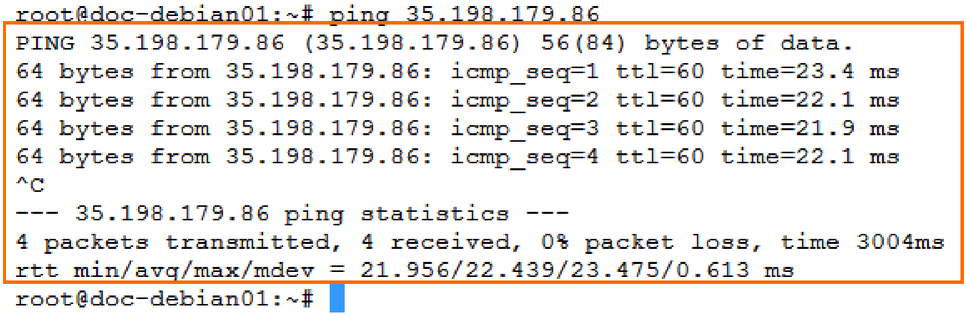


```
[root@[redacted]801:~]# speedtest-cli
Retrieving speedtest.net configuration...
Testing from DIC - [redacted] ...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by Innsbrucker Kommunalbetriebe AG (Innsbruck) [0.96 km]: 38.463 ms
Testing download speed................................................................
Download: 41.21 Mbit/s
Testing upload speed..................................................................
Upload: 51.47 Mbit/s
[2019-03-20 08:07 UTC] [-root shell-] [-Barracuda Networks-]
[root@[redacted]801:~]#
```

3. Collect the following the output from both firewalls:
   - ◦ **Speedtest output**

Ping Firewall to Firewall

1. Log into the firewall via SSH.
2. Enter `ping <public IP address of the remote firewall>`

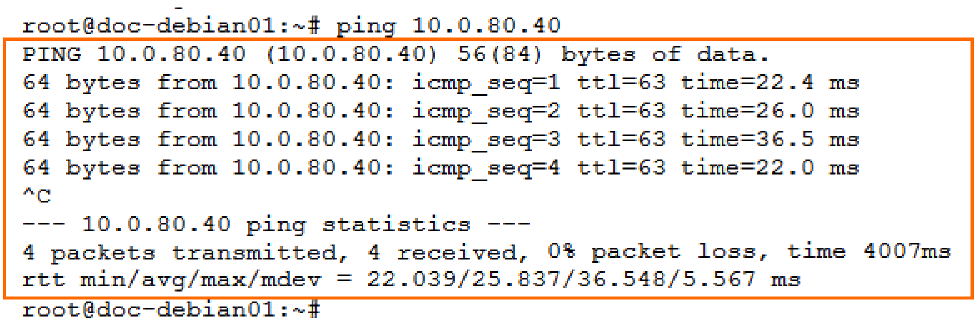


```
root@doc-debian01:~# ping 35.198.179.86
PING 35.198.179.86 (35.198.179.86) 56(84) bytes of data.
64 bytes from 35.198.179.86: icmp_seq=1 ttl=60 time=23.4 ms
64 bytes from 35.198.179.86: icmp_seq=2 ttl=60 time=22.1 ms
64 bytes from 35.198.179.86: icmp_seq=3 ttl=60 time=21.9 ms
64 bytes from 35.198.179.86: icmp_seq=4 ttl=60 time=22.1 ms
^C
--- 35.198.179.86 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 21.956/22.439/23.475/0.613 ms
root@doc-debian01:~#
```

3. Collect the following values from the output:
   - ◦ **Ping firewall to firewall**

Ping Test Client to Test Client

1. Log into the test client.
2. Enter `ping <IP address remote test client>`



```
root@doc-debian01:~# ping 10.0.80.40
PING 10.0.80.40 (10.0.80.40) 56(84) bytes of data.
64 bytes from 10.0.80.40: icmp_seq=1 ttl=63 time=22.4 ms
64 bytes from 10.0.80.40: icmp_seq=2 ttl=63 time=26.0 ms
64 bytes from 10.0.80.40: icmp_seq=3 ttl=63 time=36.5 ms
64 bytes from 10.0.80.40: icmp_seq=4 ttl=63 time=22.0 ms
^C
--- 10.0.80.40 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 22.039/25.837/36.548/5.567 ms
root@doc-debian01:~#
```

3. Collect the following values from the output:
    - **Ping test client to test client**

Test VPN Tunnel TCP Traffic Throughput

To install iperf on both test clients (Linux), use the following commands:

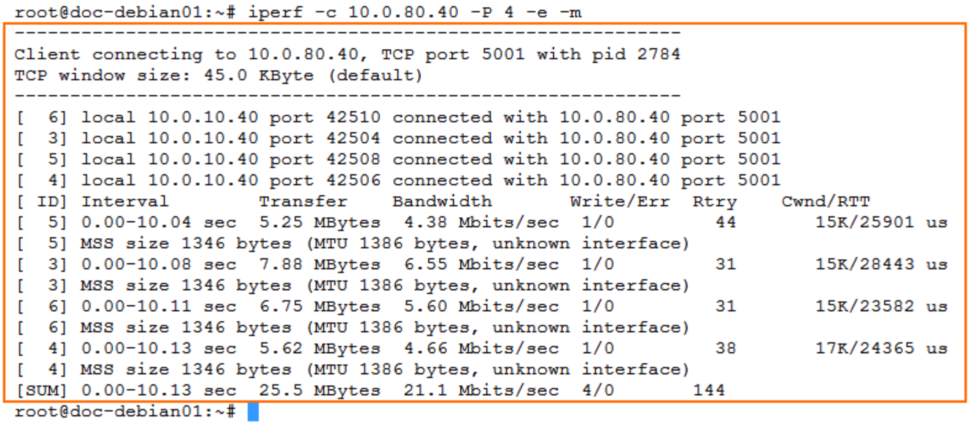
- For RedHat/CentOS: `yum install iperf3`
- For Debian/Ubuntu: `apt-get install iperf`

1. Log into the remote test client
2. Install **iperf**.
3. Start an iperf server:

```
iperf -s
```

4. Log into the local test client.
5. Install **iperf**.
6. Test the VPN throughput for TCP traffic:

```
iperf -c <IP address remote test client> -P <2 x number of CPU cores> -e
-m
```



```
root@doc-debian01:~# iperf -c 10.0.80.40 -P 4 -e -m
------------------------------------------------------------
Client connecting to 10.0.80.40, TCP port 5001 with pid 2784
TCP window size: 45.0 KByte (default)
------------------------------------------------------------
[  6] local 10.0.10.40 port 42510 connected with 10.0.80.40 port 5001
[  3] local 10.0.10.40 port 42504 connected with 10.0.80.40 port 5001
[  5] local 10.0.10.40 port 42508 connected with 10.0.80.40 port 5001
[  4] local 10.0.10.40 port 42506 connected with 10.0.80.40 port 5001
[ ID] Interval        Transfer     Bandwidth       Write/Err  Rtry     Cwnd/RTT
[  5] 0.00-10.04 sec  5.25 MBytes  4.38 Mbits/sec  1/0         44       15K/25901 us
[  5] MSS size 1346 bytes (MTU 1386 bytes, unknown interface)
[  3] 0.00-10.08 sec  7.88 MBytes  6.55 Mbits/sec  1/0         31       15K/28443 us
[  3] MSS size 1346 bytes (MTU 1386 bytes, unknown interface)
[  6] 0.00-10.11 sec  6.75 MBytes  5.60 Mbits/sec  1/0         31       15K/23582 us
[  6] MSS size 1346 bytes (MTU 1386 bytes, unknown interface)
[  4] 0.00-10.13 sec  5.62 MBytes  4.66 Mbits/sec  1/0         38       17K/24365 us
[  4] MSS size 1346 bytes (MTU 1386 bytes, unknown interface)
[SUM] 0.00-10.13 sec  25.5 MBytes  21.1 Mbits/sec  4/0       144
root@doc-debian01:~#
```

7. Repeat in the other direction.
8. Collect the following values from the output:
    - **VPN TCP iperf (fw1 to fw2)**

- **VPN TCP iperf (fw2 to fw1)**

Test VPN Tunnel UDP Traffic Throughput

Performing an iperf-reverse check in conjunction with the parameter **SD-WAN Learning Policy** set to **Auto** might lead to unexpected behavior. For more information, see SD-WAN.

1. Log into the remote test client
2. Install **iperf**.
3. Start an iperf server:

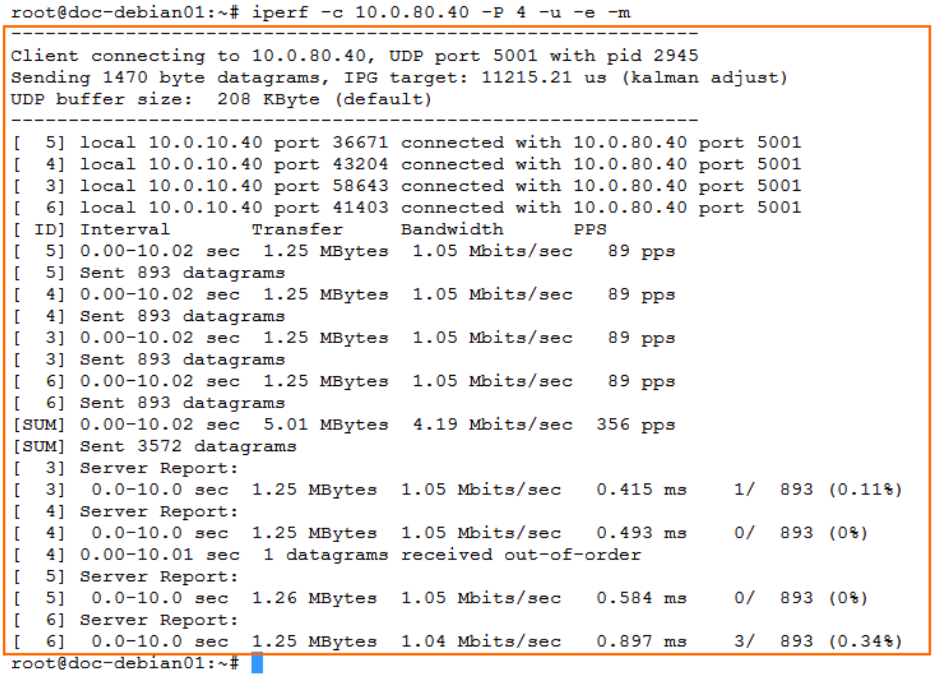
   ```
   iperf -s -u
   ```

4. Log into the local test client.
5. Install **iperf**.
6. Test the VPN throughput for UDP traffic.
   1. Specify a target bandwidth.

      ```
      -b <expected throughput / argument provided to -P
      ```

   2. Test the VPN throughput.

      ```
      iperf -c <IP address remote test client> -P <2 x number of CPU cores> -e -m -u
      ```



```
root@doc-debian01:~# iperf -c 10.0.80.40 -P 4 -u -e -m
------------------------------------------------------------
Client connecting to 10.0.80.40, UDP port 5001 with pid 2945
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  5] local 10.0.10.40 port 36671 connected with 10.0.80.40 port 5001
[  4] local 10.0.10.40 port 43204 connected with 10.0.80.40 port 5001
[  3] local 10.0.10.40 port 58643 connected with 10.0.80.40 port 5001
[  6] local 10.0.10.40 port 41403 connected with 10.0.80.40 port 5001
[ ID] Interval       Transfer     Bandwidth      PPS
[  5] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   89 pps
[  5] Sent 893 datagrams
[  4] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   89 pps
[  4] Sent 893 datagrams
[  3] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   89 pps
[  3] Sent 893 datagrams
[  6] 0.00-10.02 sec  1.25 MBytes  1.05 Mbits/sec   89 pps
[  6] Sent 893 datagrams
[SUM] 0.00-10.02 sec  5.01 MBytes  4.19 Mbits/sec  356 pps
[SUM] Sent 3572 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.415 ms    1/  893 (0.11%)
[  4] Server Report:
[  4]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.493 ms    0/  893 (0%)
[  4] 0.00-10.01 sec  1 datagrams received out-of-order
[  5] Server Report:
[  5]  0.0-10.0 sec  1.26 MBytes  1.05 Mbits/sec   0.584 ms    0/  893 (0%)
[  6] Server Report:
[  6]  0.0-10.0 sec  1.25 MBytes  1.04 Mbits/sec   0.897 ms    3/  893 (0.34%)
root@doc-debian01:~#
```

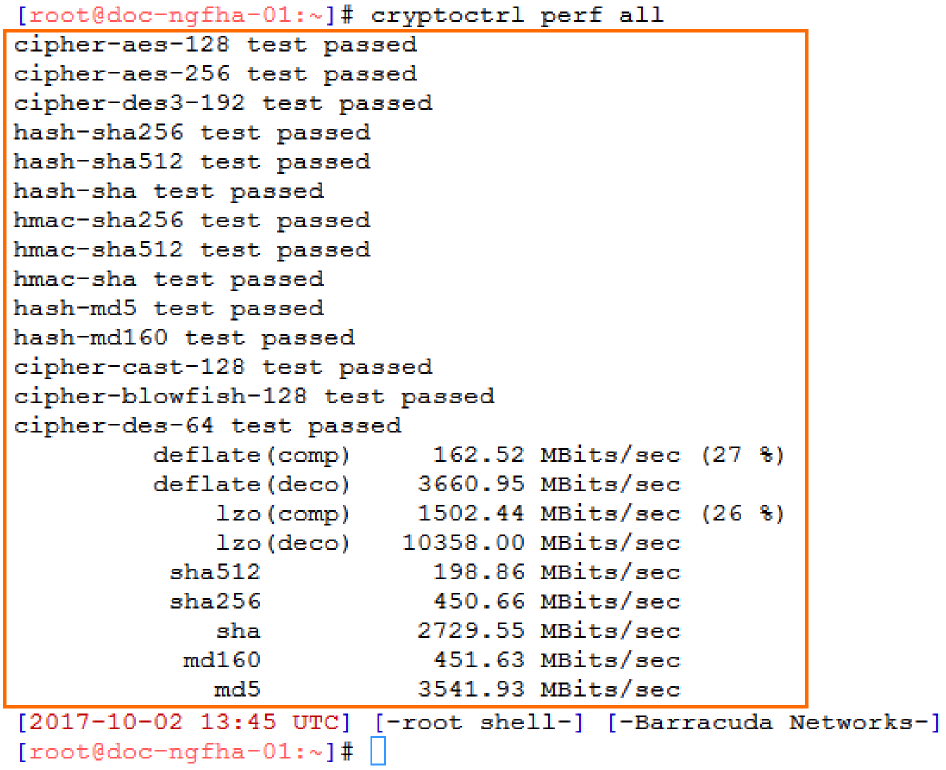
7. Repeat in the other direction.
8. Collect the following values from the output:
   - **VPN UDP iperf (fw1 to fw2)**
   - **VPN UDP iperf (fw2 to fw1)**

If you encounter a high number of "Replay Window Check failed" entries in the **VPN Status** drop cache after the test, increase the replay window setting of the transport.

(Virtual and Public Cloud Only) Test Hardware Capabilities

Execute the following test to measure the hardware capabilities on virtual and public cloud firewalls.

1. Log into the firewall via SSH.
2. Enter `cryptoctrl perf all`



```
[root@doc-ngfha-01:~]# cryptoctrl perf all
cipher-aes-128 test passed
cipher-aes-256 test passed
cipher-des3-192 test passed
hash-sha256 test passed
hash-sha512 test passed
hash-sha test passed
hmac-sha256 test passed
hmac-sha512 test passed
hmac-sha test passed
hash-md5 test passed
hash-md160 test passed
cipher-cast-128 test passed
cipher-blowfish-128 test passed
cipher-des-64 test passed
         deflate(comp)     162.52 MBits/sec (27 %)
         deflate(deco)    3660.95 MBits/sec
             lzo(comp)    1502.44 MBits/sec (26 %)
             lzo(deco)   10358.00 MBits/sec
          sha512           198.86 MBits/sec
          sha256           450.66 MBits/sec
             sha          2729.55 MBits/sec
           md160           451.63 MBits/sec
             md5          3541.93 MBits/sec
[2017-10-02 13:45 UTC] [-root shell-] [-Barracuda Networks-]
[root@doc-ngfha-01:~]# 
```

3. Collect the following values from the output:
    - **Cryptoctrl output**

Testing with SMB / CIFS Traffic

When testing performance with SMB/CIFS traffic an be difficult to receive reproducible results. When testing the same VPN tunnel with iperf and CIFS traffic, expect the transfer rate for the file transfer to be slower than the iperf value.

- Calculate the theoretical TCP throughput to know the theoretical bandwidth of the connection: https://www.switch.ch/network/tools/tcp_throughput/.
- If file transfer performance is very low, verify that you are not affected by issues with TCP receive windows scaling on Microsoft Windows. A quick search will offer troubleshooting steps and solutions for this problem.

Figures

1. fill_in_word_form.png
2. vpn_performance_testing01.png
3. vpn_perf_setup_01.png
4. vpn_perf_setup_02.png
5. vpn_perf_setup_03.png
6. vpn_perf_setup_04.png
7. vpn_perf_setup_05.png
8. vpn_perf_setup_08.png
9. vpn_perf_setup_09.png
10. box_rate_limit.png
11. acpf_proc_output.png
12. ktina_proc_output.png
13. speedtest.png
14. ping_fw_to_fw.png
15. ping_client_to_client.png
16. iperf_tcp.png
17. iperf_udp.png
18. cryptoctrl_test.png