Chelsio T7 NIC Performance Leadership Across Tx, Rx, and BiDi Workloads

Delivering Scalable 400 GbE Connectivity and CPU Efficiency on Linux

Executive Summary

The **Chelsio T7 adapters** deliver exceptional networking performance on Linux, achieving near line-rate throughput with minimal CPU utilization. Designed for high-speed data center and enterprise workloads, the T7 leverages advanced offload capabilities and efficient PCIe Gen5 architecture to accelerate network I/O and reduce system bottlenecks. With robust Linux driver support and optimized TCP/IP performance, it ensures consistent low latency, scalability, and superior efficiency for modern applications demanding **400 GbE** connectivity.

Test Results

Note: These results are preliminary, and further improvements are currently in progress.

The below graphs compare the Tx and Rx performance of Chelsio, NVIDIA CX7, and BRCM NICs with MTU 1500. In the Tx results (**Figure 1**) across 48 connections, all NICs reach ~380 Gbps line rate, but Chelsio achieves it with the **lowest CPU usage**, highlighting its efficiency advantage. In the Rx results (**Figure 2**), Chelsio delivers higher throughput at smaller I/O sizes and reaches line rate at larger sizes. Overall, Chelsio achieves Tx line-rate performance with the lowest CPU usage, and in Rx it delivers better performance at smaller I/O sizes compared to Broadcom and NVIDIA, while also providing the highest bandwidth and superior CPU efficiency.

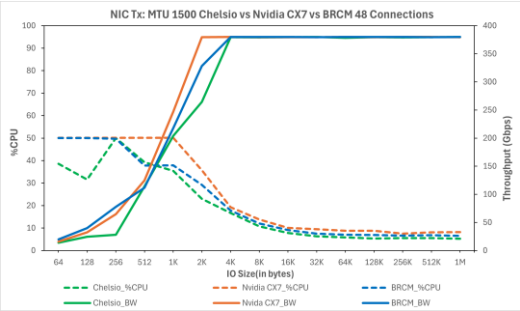

Figure 1 – NIC Tx Performance Comparison: Chelsio vs Nvidia CX7 vs BRCM (MTU 1500)

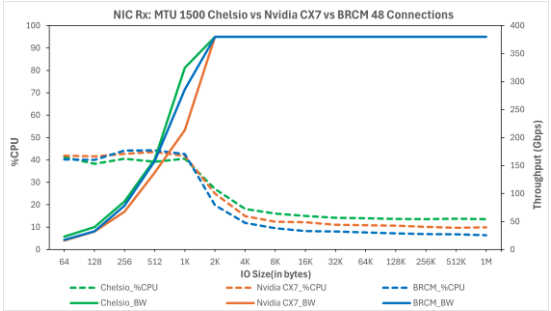

Figure 2 – NIC Rx Performance Comparison: Chelsio vs Nvidia CX7 vs BRCM (MTU 1500)

The below graph compares the BiDi performance of Chelsio, NVIDIA CX7, and BRCM NICs with MTU 1500 across 96 connections. Chelsio scales rapidly, reaching about 740 Gbps—significantly exceeding BRCM. It also maintains lower CPU usage at larger I/O sizes, providing superior efficiency. Overall, Chelsio delivers the highest BiDi bandwidth and superior CPU efficiency, making it the top performer.

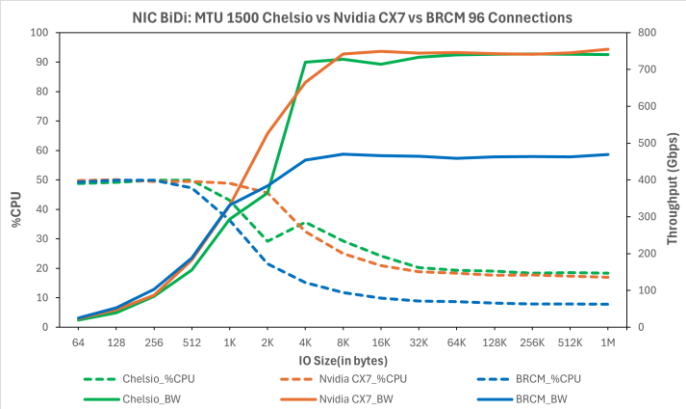

Figure 3 – NIC BiDi Performance Comparison: Chelsio vs Nvidia CX7 vs BRCM (MTU 1500)

Test Configuration

The setup consists of a server connected to three clients through a switch. The server and client machines are configured with the latest Chelsio Unified Wire drivers for Linux. One Chelsio T72200-FH adapter is installed on the server with both the ports connected to the switch and configured with a standard MTU of 1500 bytes, while two Chelsio T62100-CR adapters are installed in each client system, with Port 0 of each adapter connected to the same switch and configured with a standard MTU of 1500 bytes. Note that the tests are repeated with BRCM and Nvidia CX7 adapters installed in the same server.

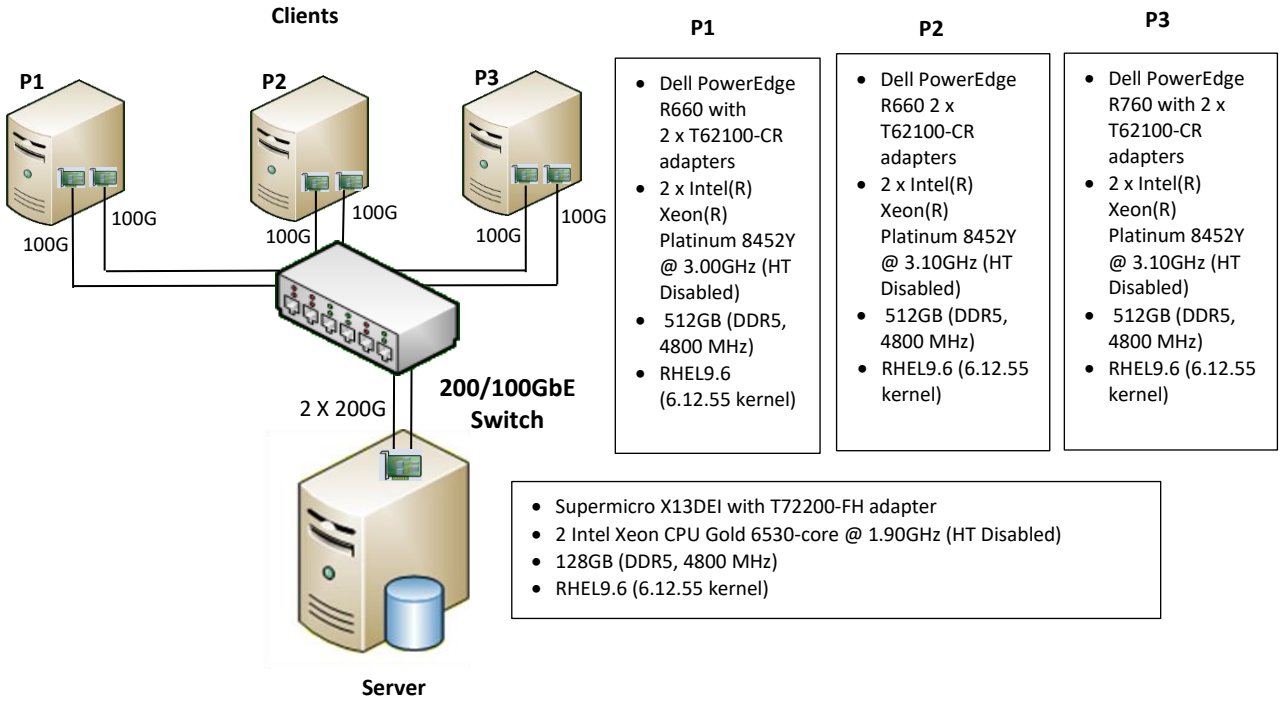

Figure 4 – Test Setup

Setup Configuration

BIOS settings:

- Disable Hyper-Threading Technology.
- Disable Sub-NUMA Clustering (SNC).
- Disable CPU Power Management Features including C-states, P-states, and Turbo Boost
- Configure System power management profile in Maximum Performance Mode.

i. Install RHEL 9.6 and compile 6.12.55 kernel.

ii. Tune profile in the TuneD system tuning service:

```
[root@host ~]# tuned-adm profile network-throughput
```

Server

Make the following tunings:

a. For each port, run the below commands:

```
[root@host~]# ethtool -C ethX adaptive-rx on
[root@host~]# ethtool -A ethX autoneg off tx on rx on
```

b. Set IRQ affinity as follows: Port 0 mapped to CPU cores 0–15, Port 1 mapped to CPU cores 16–31.

```
[root@host~]# t4_perftune.sh -i ethX -c 0-15 -s
[root@host~]# t4_perftune.sh -i ethY -c 16-31 -s
```

where, *ethX* is port0 and *ethY* is port1 of the T72200-FH adapter.

Clients

Make the following tunings:

a. For each port, run the below commands:

```
[root@host~]# ethtool -C ethX adaptive-rx on
[root@host~]# ethtool -A ethX autoneg off tx on rx on
```

b. Set IRQ Affinity as:

```
[root@host~]# t4_perftune.sh -n -Q nic -s
```

BRCM

Make the following tunings:

a. For each port, run the below commands:

```
[root@host~]# ethtool -C ethX adaptive-rx on
[root@host~]# ethtool -A ethX autoneg off tx on rx on
```

b. Set IRQ Affinity as: Port 0 and 1 mapped to CPU 0-31.

c. Set sysctl tuning using below command:

```
[root@host~]# t4_perftune.sh -n -s -C
```

Nvidia CX7

Make the following tunings:

a. For each port, run the below commands:

```
[root@host~]# ethtool -C ethX adaptive-rx on
[root@host~]# ethtool -A ethX autoneg off tx on rx on
[root@host~]# ethtool -K ethX1 large-receive-offload on
[root@host~]# ethtool -L ethX combined 32
```

b. Set IRQ Affinity as: Port 0 and 1 mapped to CPU 0-31.

c. Set sysctl tuning using below command:

```
[root@host~]# t4_perftune.sh -n -s -C
```

Commands Used

1. Start iperf3 Server on Client Machines on both interfaces:

```
[root@client ~]# for i in 1 2; do
   numa_node=$(cat /sys/class/net/<intf_${i}>/device/numa_node);
   numactl -N ${numa_node} -m ${numa_node} iperf3 -s -D -p <PORT_NO_${i}>
done
```

Notes:

Replace "<intf_${i}>" with the actual network interface names (Example: ens1f0, ens1f1).

"<PORT_NO_${i}>" should be unique for each interface (Example: 5201, 5202).

The numactl binding ensures the iperf3 process runs on the same NUMA node as its network interface, optimizing performance.

2. Run tests from the Server.

Assume that the DUT interface is attached to NUMA node 0 (Example: T72200-FH adapter in CPU1 PCIe slot).

Client interfaces have IPs assigned as "<Client${client}_intf${i}_IP>".

2.1. Transmit (TX) Test: To measure transmit performance from the DUT to clients,

```
[root@server ~]# for client in 1 2 3; do
    for i in 1 2; do
       numactl -N 0 -m 0 iperf3 -c <Client${client}_intf${i}_IP> -Z --
skip-rx-copy -N -t 30 -i 5 -p <PORT_NO_${i}> -l <IO_Size> -P
<Num_of_Connections> -f g -O 10 &
    done
done
wait
```

2.2. Receive (RX) Test: To measure receive performance from clients to the DUT,

```
[root@server ~]# for client in 1 2 3; do
    for i in 1 2; do
       numactl -N 0 -m 0 iperf3 -c <Client${client}_intf${i}_IP> -Z --
skip-rx-copy -N -t 30 -i 5 -p <PORT_NO_${i}> -l <IO_Size> -P
<Num_of_Connections> -f g -O 10 -R &
    done
done
wait
```

2.3. Bi-directional (BiDi) Test: To measure simultaneous transmit and receive performance,

```
[root@server ~]# for client in 1 2 3; do
    for i in 1 2; do
       numactl -N 0 -m 0 iperf3 -c <Client${client}_intf${i}_IP> -Z --
skip-rx-copy -N -t 30 -i 5 -p <PORT_NO_${i}> -l <IO_Size> -P
<Num_of_Connections> -f g -O 10 --bidir &
    done
done
wait
```

Conclusion

The Chelsio T7 adapters consistently prove to be the top performer in both throughput and CPU efficiency. Chelsio achieves TX line-rate performance with the lowest CPU usage among the three NICs. In Rx, it delivers better performance at smaller I/O sizes compared to Broadcom and NVIDIA, while also providing the highest bandwidth and superior CPU efficiency at larger sizes. During the BiDi test, Chelsio demonstrates a clear advantage, reaching 740 Gbps with lower CPU usage throughout. Overall, the results show that Chelsio offers the highest throughput, best scalability, and most efficient CPU utilization, making it the strongest and most reliable choice across Tx, Rx, and BiDi workloads.

Related Links

High-Performance NIC Optimization on Linux With Chelsio T7
AI Networking Solution: Chelsio T7 DPU and S7/T6 SmartNICs
T7 Product Brief
AI Networking: The Role of DPUs
Offload Protocols with Inline IPsec demonstration on T7 Emulation Platform
iSCSI JBOF with T7
NVMe/TCP and iSCSI JBOF with T7