

400G kTLS/SSL Offload & Encryption with FreeBSD

Secure TCP/IP processing with T7/S7 Inline kTLS/SSL Crypto Function

Executive Summary

The Inline kTLS/SSL offload capability offered by Chelsio T7/S7 adapters is uniquely capable of producing 395Gbps to 370Gbps rate kTLS/SSL performance. They are designed specifically to perform computationally intensive cryptographic operations more efficiently than general purpose CPUs. Servers with system load comprising of cryptographic operations see great performance improvement by offloading crypto operations to the Chelsio T7/S7 adapter. Thanks to an inbox driver in the FreeBSD kernel, the seventh generation adapters from Chelsio are plug-and-play solutions for secure networking.

Chelsio T7/S7 adapters are supported natively in the FreeBSD kernel, requiring no external drivers or packages. This allows immediate deployment on secure appliances and ensures a predictable update cycle—making T7/S7 an easy, reliable choice for secure, high-performance environments.

This paper presents Chelsio 400G Inline kTLS/SSL offload performance using T7/S7 for networking and inline cryptographic offload only (handshake still on host) with a real-world application, nginx web server running in FreeBSD. It establishes the fact that using T7/S7 to offload the encryption, the web server consumes significantly less CPU with consistent performance.

Chelsio T7/S7 Inline kTLS/SSL

Solution

The Terminator 7 (T7/S7) ASIC from Chelsio Communications, Inc. is a seventh generation, high performance 1/10/25/40/50/100/200/400G unified wire engine which enables concurrent secure communication and secure storage, all for the price and power of a typical NIC. T7 supports most popular AES/SHA cipher suites in inline kTLS/SSL mode, yielding the following **provisional** bandwidth metrics:

- **T7 Adapters:** 367 Gbps (wire BW) / 326 Gbps (application BW)
- **S7 (on Turin-based systems):** 395 Gbps in the TX direction

T7/S7 adapters offload the kTLS/SSL PDU crypto, while handshake is still performed by the host.

- OpenSSL modifies and provides hooks for data transmission, receive, and key programming.
- Third party/customized kTLS/SSL implementations are also supported.
- Supports crypto for all kTLS/SSL ports.

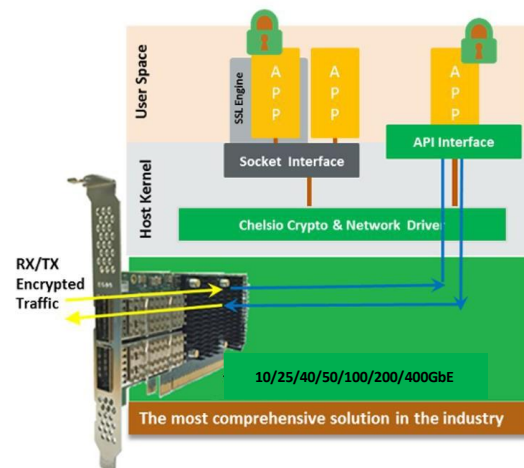


Figure 1 - Chelsio T7/S7 Inline kTLS/SSL

Test Results

The following graph compares the throughput and CPU usage using Chelsio T7 crypto accelerator in Inline kTLS/SSL offload mode. The baseline numbers are collected using **wrk** tool with connections ranging from 10 to 100.

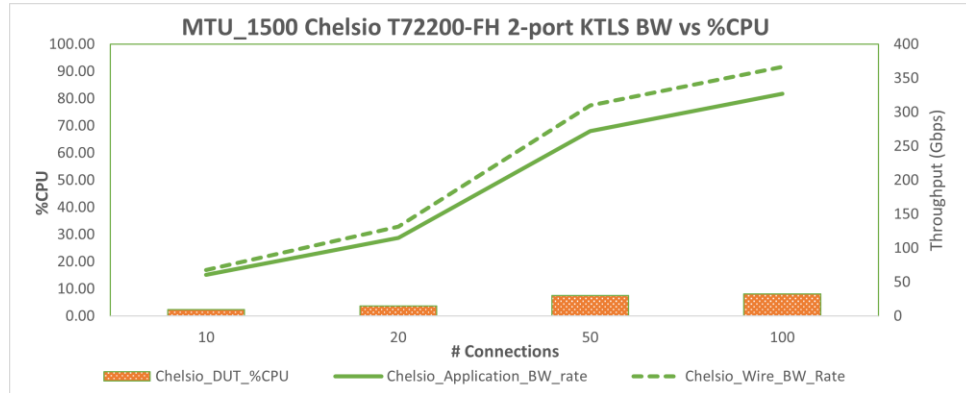


Figure 2 - Throughput and CPU Usage vs. # Connections

The above graph tracks the connection scaling, CPU utilization, and throughput performance for the Chelsio T7 adapter using Kernel TLS (kTLS). Driven by the T7-specific test bed, bandwidth peaks at a provisional **367 Gbps (wire)** and **326 Gbps (application)**. By comparison, the S7 achieves **395 Gbps (TX) on Turin-based systems** (environment details and scaling data not shown).

Test Configuration

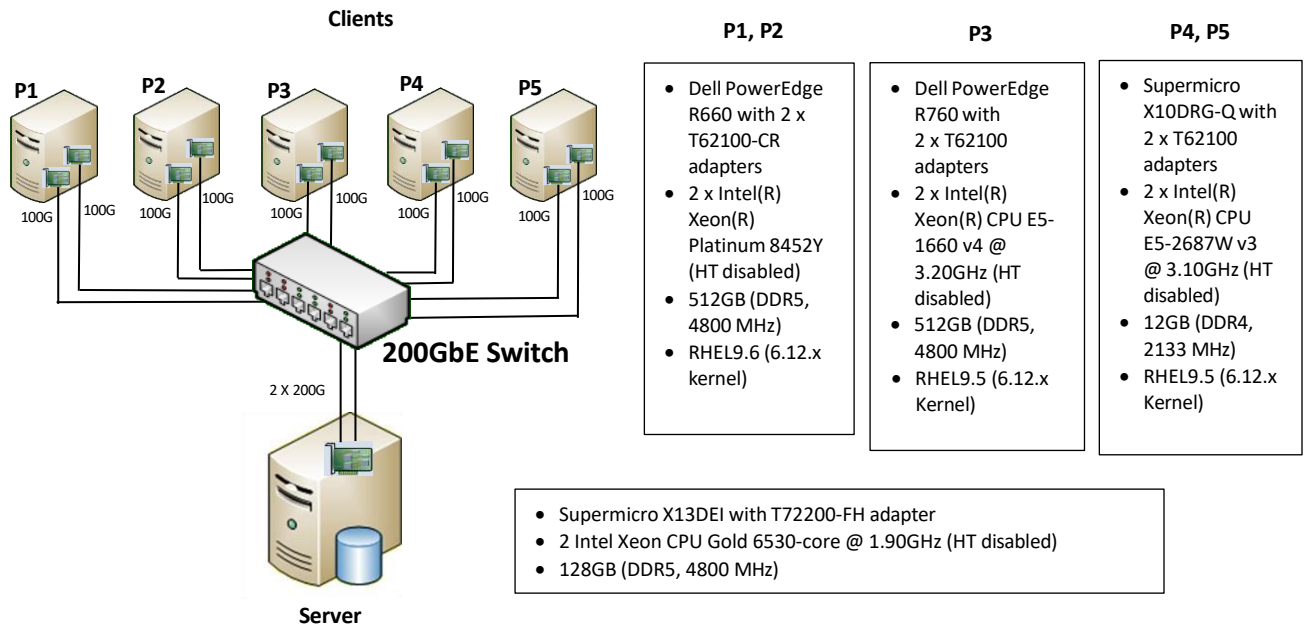


Figure 3 – Test setup

The following test bed configuration applies exclusively to the T7 platform; the corresponding S7 configuration on Turin-based systems is omitted. The T7 setup consists of a server connected to five clients using dual ports on each system. The server machine was configured with FreeBSD inbox drivers, and the client was configured with the latest Chelsio Unified Wire drivers for Linux. A Chelsio T72200-FH adapter was installed on the server with a standard MTU of 1500B, while two Chelsio T62100-CR adapters were installed in each client system, with Port 0 of each adapter configured with the standard MTU of 1500B.

Setup Configuration

Server

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. Download and install **FreeBSD-16.0** from <https://www.freebsd.org/> and compile latest kernel with below fix:

<https://reviews.freebsd.org/D53385>

- ii. Add below knobs to `/boot/loader.conf` :

```
root@host:~ # cat /boot/loader.conf
hw.cxgbe.ntxq="32"
hw.cxgbe.nrxq="32"
hw.cxgbe.qsize_txq="2048"
hw.cxgbe.qsize_rxq="2048"
hw.cxgbe.holdoff_timer_idx="5"
hw.cxgbe.lro_mbufs="1024"
```

- iii. Configure T7 interfaces with KTLS enabled:

```
root@host:~ # sysctl kern.ipc.tls.enable=1
root@host:~ # sysctl kern.ipc.tls.ifnet.permitted=1
root@host:~ # kldload if_cxgbe
root@host:~ # ifconfig che0 103.11.11.235/24 mtu 1500 up
root@host:~ # ifconfig che1 103.22.22.235/24 mtu 1500 up
root@host:~ # sysctl -w dev.chnex.0.tls.inline_keys=1
```

- iv. Install nginx server and use below config file:

```
root@host:~ # cat /usr/local/etc/nginx/nginx.conf
worker_processes auto;
events {
    worker_connections 4096;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    client_body_buffer_size 16k;
    proxy_buffer_size 4k;
    proxy_buffers 32 4k;
    proxy_busy_buffers_size 64k;
    server {
```

```
listen      443 ssl;  
listen     [::]:443 ssl;
```

```
server_name nolin;
ssl_certificate      cert.pem;
ssl_certificate_key  cert.key;
ssl_session_cache   shared:SSL:1m;
ssl_session_timeout 5m;
ssl_ciphers          HIGH:!aNULL:!MD5;
ssl_protocols        TLSv1.3 TLSv1.2 TLSv1.1;
ssl_prefer_server_ciphers on;
ssl_conf_command Options KTLS;
location / {
    root    /usr/local/www/nginx;
    index  index.html index.htm;
}
}
```

v. Create two files each of 2GB size for client access:

```
root@host:~ # ls /usr/local/www/nginx/P*2G
/usr/local/www/nginx/P0_2G    /usr/local/www/nginx/P1_2G
```

vi. Start nginx server:

```
root@host:~ # cpuset -l <0-31> -n ft:0 service nginx onestart
```

This configuration binds the nginx process to CPU cores 0-31, which corresponds to the NUMA node local to the Chelsio T72200-FH network adapter.

Client

- Same as server BIOS options.
 - Compile and install kernel version 6.12.28 on RHEL9.5 OS on each Peer, install the latest drivers available from [Chelsio Download Center](#).
- i. Disable virtualization, C-states technology, VT-d, Intel I/O AT, SR-IOV in system BIOS.
 - ii. Add the below parameters to grub kernel command line and boot into the newly installed 6.12.28 kernel.

```
[root@host~]# grubby --args="intel_idle.max_cstate=0 processor.max_cstate=0
intel_pstate=disable" --update-kernel="boot/vmlinuz-6.12.28"
```

iii. Install Chelsio Unified Wire drivers.

```
[root@host~]# cd ChelsioUwire-x.x.x.x
[root@host~]# make install
```

iv. Set the below tuned-adm profile.

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

v. Load the Chelsio Network driver and bring up interfaces with IPv4 address.

```
[root@host~]# modprobe cxgb4
[root@host~]# ifconfig ethX <IPv4 address> up; ifconfig ethY <IPv4 address> up
```

ethX is a T6 interface from adapter1, ethY is a T6 interface from adapter2.

vi. CPU affinity was set.

```
[root@host~]# t4_perftune.sh -n -Q nic -c 1-16
```

vii. On each peer run below commands in parallel:

```
[root@host~]# numactl -N0 -m0 wrk -c<#conns> -t20 -d30 -v https://103.11.11.235/P0_2G &  
[root@host~]# numactl -N1 -m1 wrk -c<#conns> -t20 -d30 -v https://103.22.22.235/P1_2G &
```

#conns is number of connections per client (ranging from 1 - 10).

Conclusion

This paper describes the performance of Chelsio's T7 SSL offload solution on FreeBSD. The Chelsio T7 Inline kTLS/SSL offload delivers provisional bandwidths of **367 Gbps wire** and **326 Gbps application** with **minimal CPU utilization**, while S7 achieves **395 Gbps (TX) on Turin-based systems**. By significantly outperforming software-based encryption and offloading cryptographic operations to hardware, these solutions ensure higher efficiency, lower latency, and consistent scalability. With native FreeBSD inbox driver support, T7 provides a seamless, high-performance solution for secure and efficient data center networking.

Related Links

[Chelsio T7 NIC Performance Leadership Across Tx, Rx, and BiDi Workloads](#)

[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](#)