TLDK OVERVIEW

TRANSPORT LAYER DEVELOPMENT KIT

Mohammad Abdul Awal
Mohammad.Abdul.Awal@intel.com
April 2017

Contributions from Ray Kinsella, Konstantin Ananyev, Keith Wiles

Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No computer system can be absolutely secure.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

What is TLDK (1)?

- TLDK stands for “Transport Layer Development Kit”
- TLDK is a high performance L4 protocol library
  - Provides APIs to develop L4 applications in client/server modes.
- Built on top of DPDK.
  - Uses DPDK API/features across the libraries.
  - Follow DPDK design concepts.
    - Bulk packet processing, Non-blocking API, No-mode switching, Cache optimization , Memory locality etc.
- The provided API is not compatible with BSD socket API.
  - keep similar semantics whenever possible.

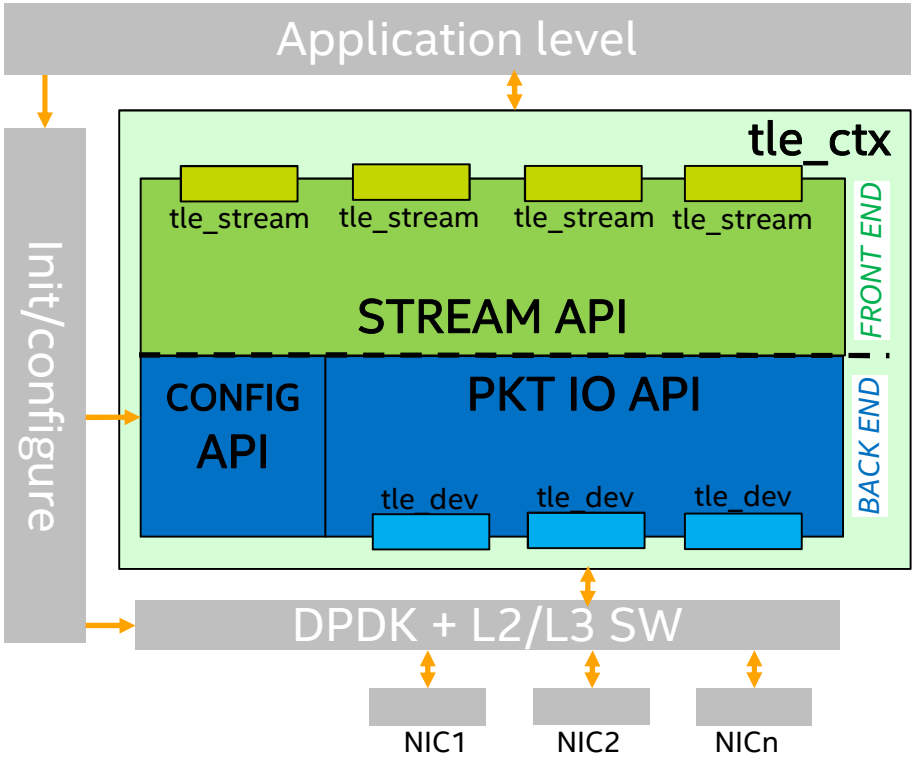
What is TLDK (2)?

- TLDK is application driven
  - protocols are driven by the application needing the data, instead of data rx/tx events driving the application.
- TLDK supports
  - UDP and TCP processing libraries.
  - TCP options.
    - MSS, timestamp, window scaling, (SACK in roadmap).
  - TCP features.
    - ddos protection, congestion control, (delayed ack in roadmap).
  - TCP HW offloads; (TSO in roadmap).
  - Ipv4/IPv6

What is TLDK (3)?

- Provides code examples demonstrating a number of use cases.
  - simple send/recv or both on UDP/TCP streams.
  - L4 packet forwarding between different streams.
  - reassemble/fragment IP packets (based on DPDK librte_ip_frag).
  - Support RSS using L4 ports.
  - All code examples support IPv4/IPv6.

TLDK API Overview



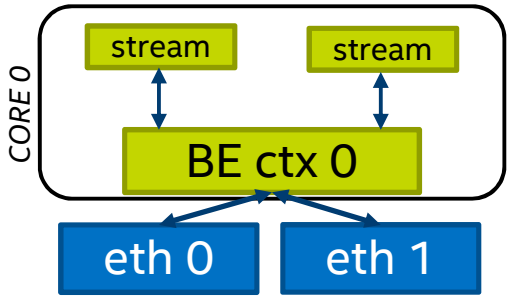
- TLDK context represents an 'independent copy of stack'.
- API can be logically divided into:
  - Back-End (BE):
    - Config API (dev add/remove).
    - PKT IO (RX/TX bulk).
  - Front-End (FE):
    - stream control and IO (open, close, listen, recv, send, etc.).
- BE API is not thread-safe.
- FE API is thread-safe.

TLDK API Overview

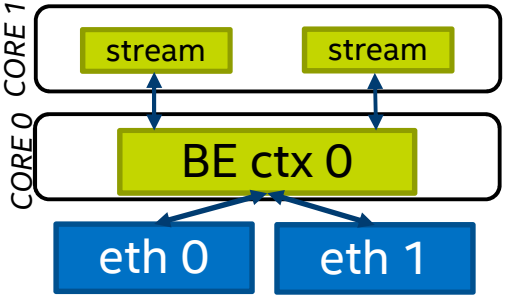
- TLDK Backend
  - tle_ctx_create(const struct tle_ctx_param *ctx_prm)
  - tle_add_dev(struct tle_ctx *ctx, const struct tle_dev_param *dev_prm)
  - tle_tcp_tx_bulk(struct tle_dev *dev, struct rte_mbuf *pkt[], uint16_t num)
  - tle_tcp_rx_bulk(struct tle_dev *dev, struct rte_mbuf *pkt[], struct rte_mbuf *rp[], int32_t rc[], uint16_t num)
- TLDK Frontend
  - tle_tcp_stream_listen(struct tle_stream *s)
  - tle_tcp_stream_accept(tle_stream *s, struct tle_stream *rs[], uint32_t num)
  - tle_tcp_stream_open(struct tle_ctx *ctx, const struct tle_tcp_stream_param *prm)
  - tle_tcp_stream_close_bulk(struct tle_stream *ts[], uint32_t num)
  - tle_tcp_stream_send(tle_stream *s, struct rte_mbuf *pkt[], uint16_t num)
  - tle_tcp_stream_recv(tle_stream *s, struct rte_mbuf *pkt[], uint16_t num)
  - tle_tcp_process(struct tle_ctx *ctx, uint32_t num)

Possible deployment scenarios

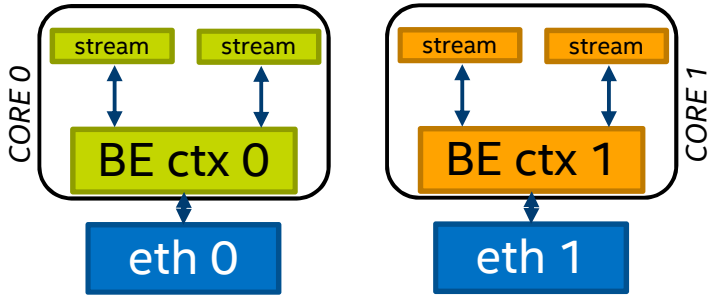
1. one TLDK ctx, BE and FE on the same core



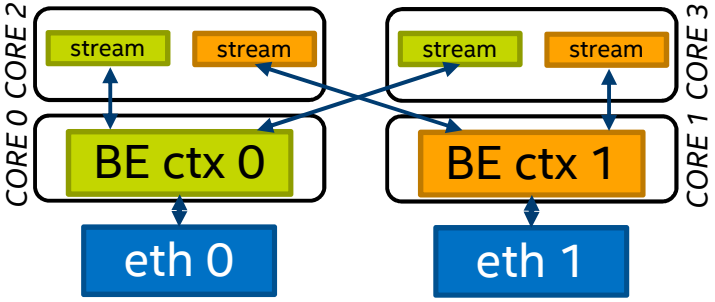
2. one TLDK ctx, BE and FE on different cores



3. two TLDK ctxs, for each BE and FE on the same core



3. two TLDK ctxs, BE and FE on different cores

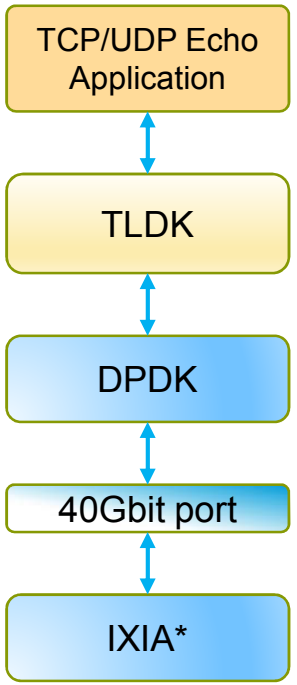


TLDK UDP Performance

CPU: Intel® Xeon® Processor E5-2699 v3 @ 2.30GHz
64G Ram, Dual socket system, 2x400GB SSD, 2x1TB drives
NIC: Intel® Ethernet Controller XL710 for 40GbE QSFP+
Firmware: 5.04 0x80002505 0.0.0
DPDK: 16.07
Linux: Ubuntu* 15.10 (GNU/Linux 4.2.0-16-generic x86_64)
TLDK: Current release (2016-09-15)

UDP Packet size used is 64 bytes, 5 cores we max out the PCI

#Physical Cores	#Queues	Frame Rate Mpps
1	1	7.4
2	2	14.8
3	3	22.2
4	4	29.5
5	5	36.4 (max for PCI)



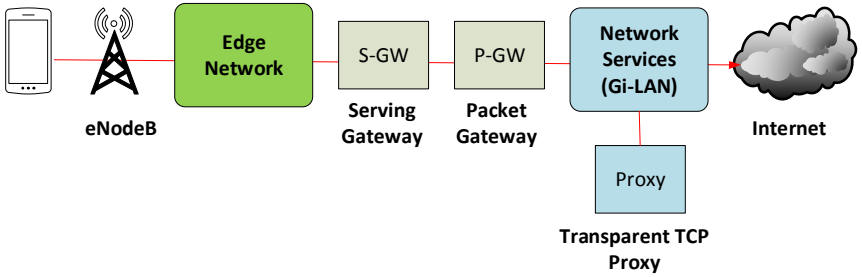
TLDK TCP Performance

TLDK TCP	CPS	CRPS	TG, SUT, DUT
HW	-	410k	Ixia, TLDK TCP FWD, BDW @ 2.3Ghz
SW	550k	300k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710

TLDK TCP – SW	RPS	TG, SUT, DUT
RPS (100B RR, over 100 connections)	910k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710
RPS (100B RR, over 1k connections)	860k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710
RPS (100B RR, over 10K connections)	820k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710
RPS (1k RR, over 100 connections)	630k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710
RPS (1k RR, over 1k connections)	620k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710
RPS (1k RR, over 10k connections)	600k	Warp17 v1.4, TLDK TCP FWD, BDW 2.2Ghz @ 2.20 + 1 x XL710

Some possible use-cases for TLDK

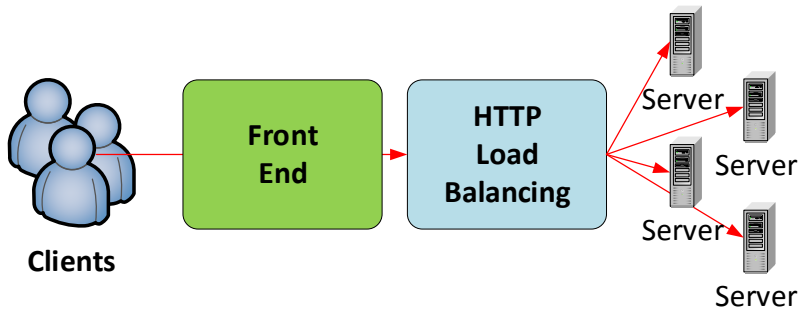
TCP Transparent Proxy



- Common middle box in the cellular network, deployed in the Gi-LAN
- Observes & passes-through the 3-way hand-shake and setups a shadow flow.
- Sends ACK's on behalf of Client, caches TCP packets close to the network.
- Benefits are latency splitting, reduction in retransmission and buffer-bloat.

Reverse Proxy & Load-balancer

- Common middle box in the data-centre, deployed between the Front-end (IDS/FW) and webservers.
- Typically supports features; server fault detection, session persistence.
- Typically supports distribution algorithms; round-robin, ip-hash and least-connected.
- Benefits are optimizing resource utilization, maximizing throughput, reducing latency, and ensuring fault-tolerant configurations.

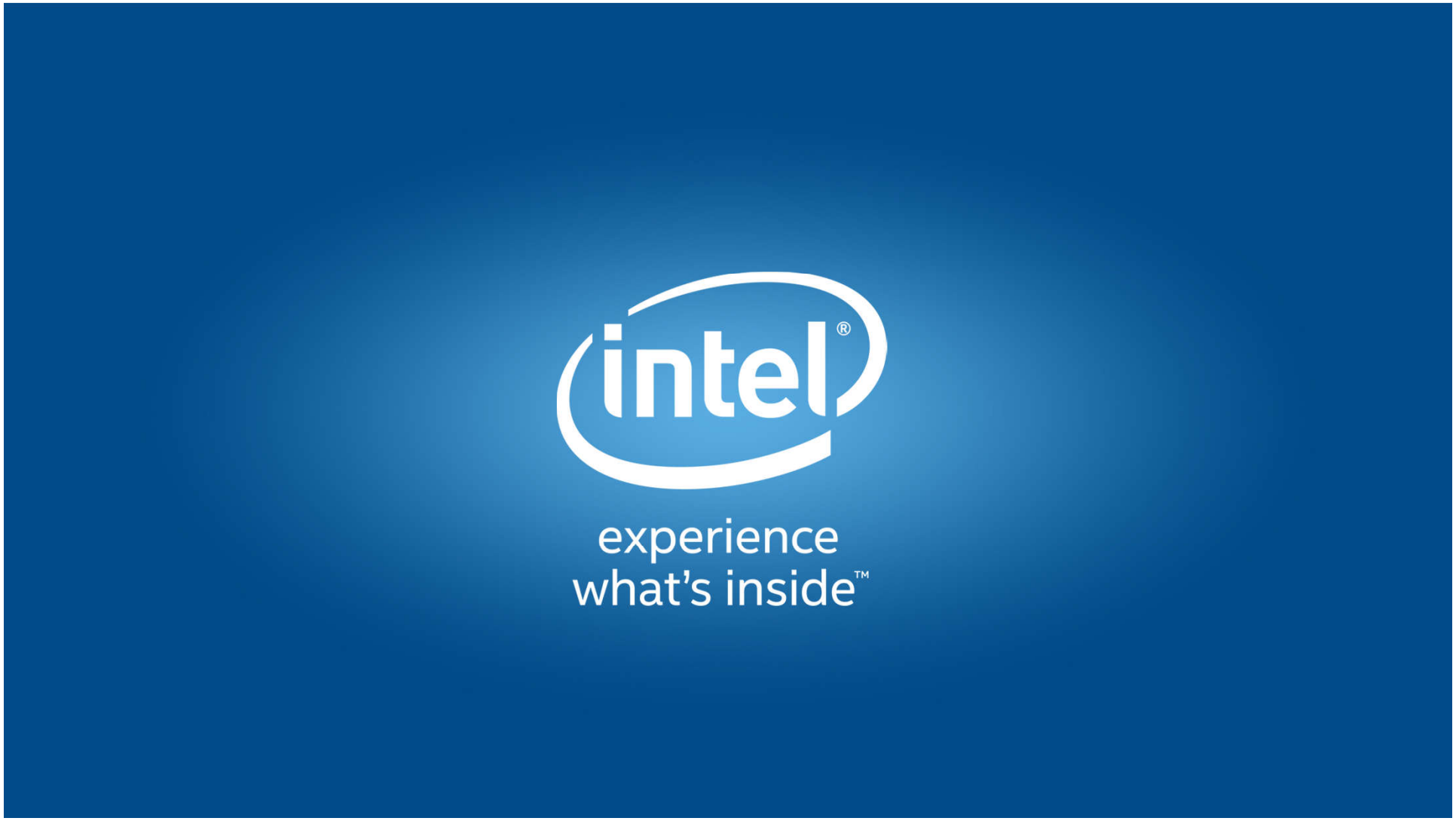


Summary

- Network operators, data centres are optimizing TCP workloads to reduce impedance mismatch, improve overall utilization, UX …
- Userspace TCP/UCP stack's are growing in popularity for reasons of performance and ease of upgrade.
- TLDK is a grounds up based on core DPDK design concepts designed to achieve ***Network Node*** performance requirements.
- TLDK is now available @ **wiki.fd.io/view/TLDK**

ML: **tldk-dev@lists.fd.io** E-mail: Mohammad.Abdul.Awal@intel.com

Please come and join us!


intel
experience
what's inside™