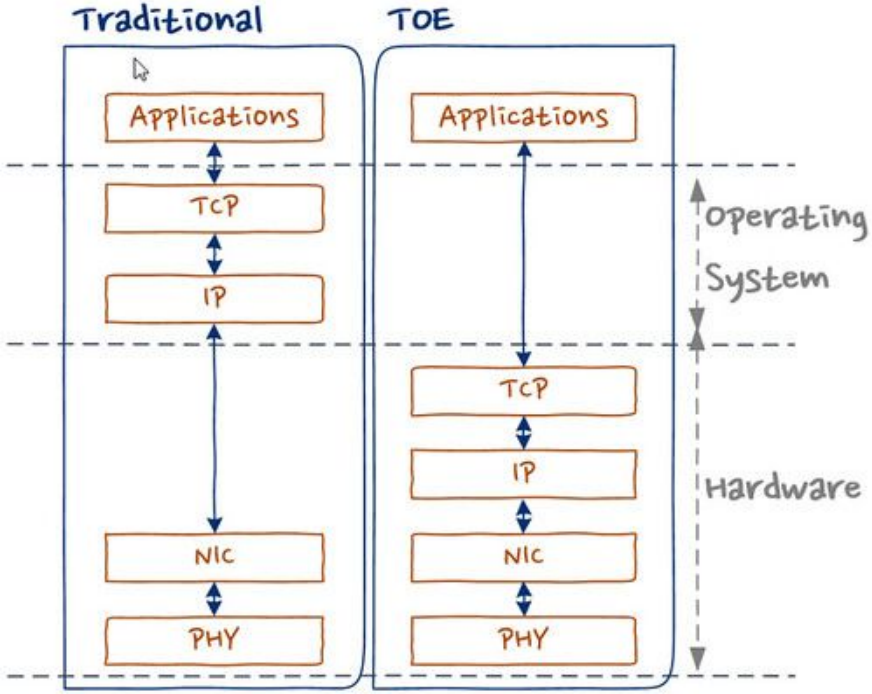
TCP and ULP Offload via AF_XDP Sockets

Tom Herbert, Felipe Magno de Almeida
Netdev0x17

THIS IS NOT TOE!!!

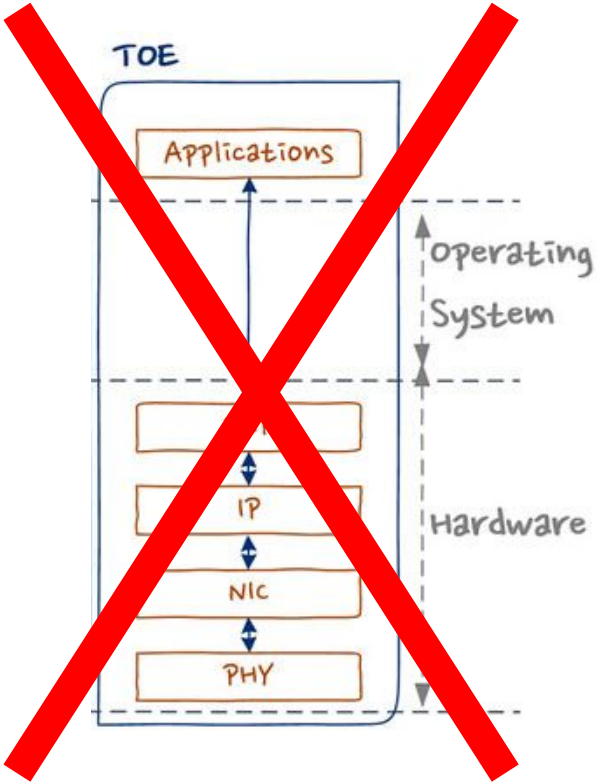
The brief history of TCP Offload

- TOE: TCP Offload Engines, offload TCP to specialized hardware
- Promise was great performance
- TOE was popular 20 years ago
- TOE startups, Windows Chimney
- Open Onload, run TCP stack in userspace, socket intercept using LD_PRELOAD



But TOE was a really, really bad idea!

- Security updates
- Point-in-time solution
- Different network behavior
- Performance
- Hardware-specific limits
- Resource-based denial-of-service attacks
- RFC compliance
- Linux features
- Requires vendor-specific tools
- Poor user support
- Short term kernel maintenance
- Long term user support
- Long term kernel maintenance
- Eliminates global system view



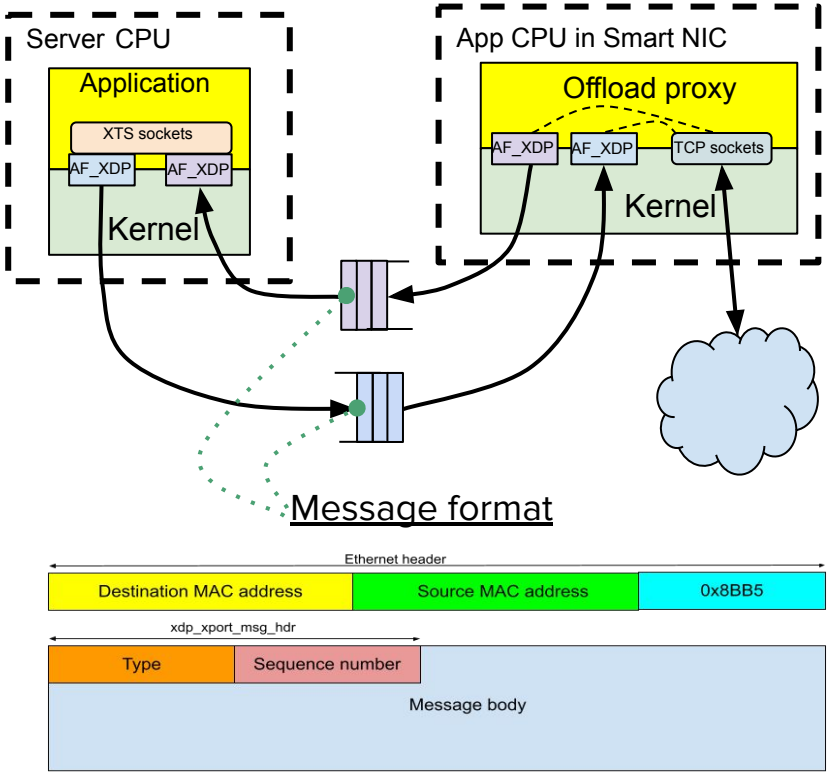
Source: https://wiki.linuxfoundation.org/networking/toe

Modern motivations for TCP and ULP Offload

- Offload infrastructure processing from server CPUs
- Not just TCP offload, but also ULP offload like RDMA, TLS
- (and yes, higher throughput and lower latency would still be nice)

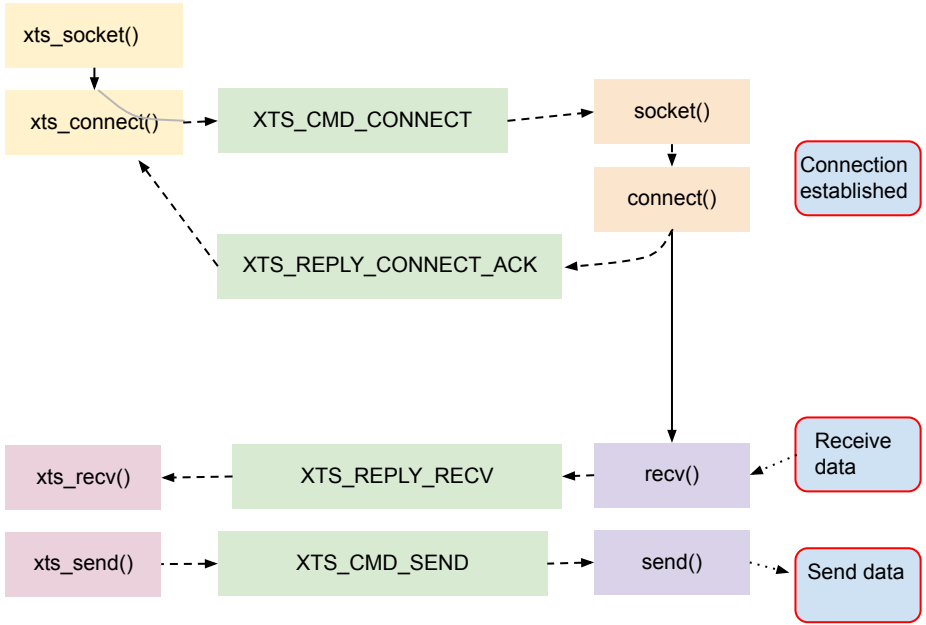
TCP and ULP offload via AF_XDP sockets

- Server CPU and App CPU communicate over shared NIC queues via AF_XDP
- Messages sent on queues contain directives and data
- Application uses “XTS sockets” like normal socket calls. Socket calls are mapped to messages
- App CPU runs an offload proxy that proxies between queues & TCP sockets

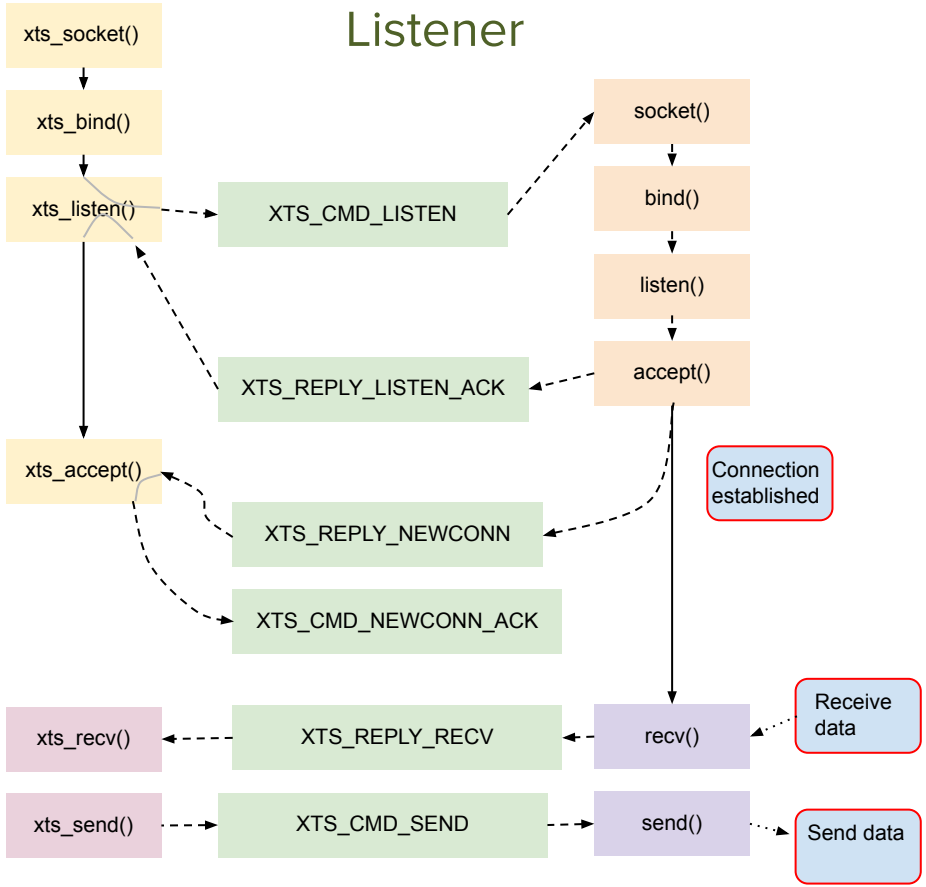


Operations

Connect



Listener

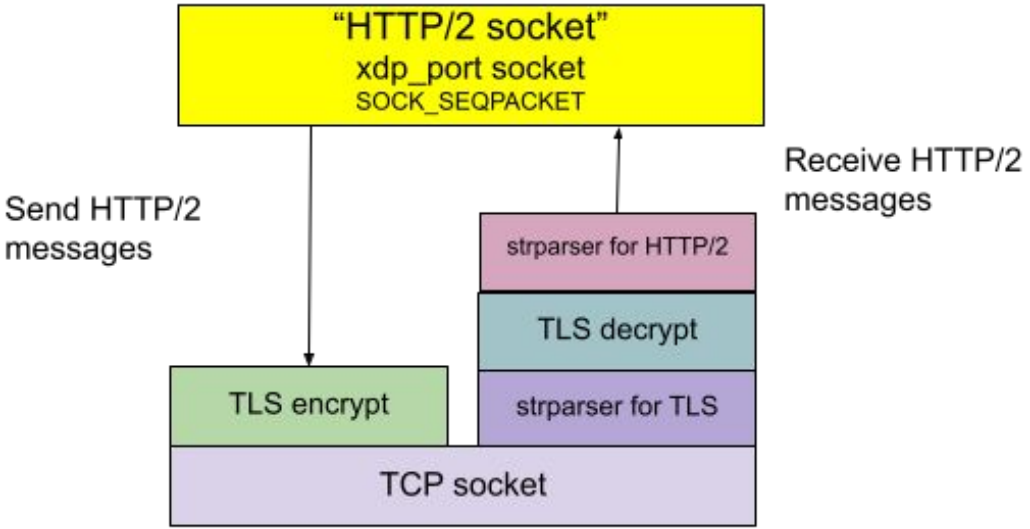


Design points

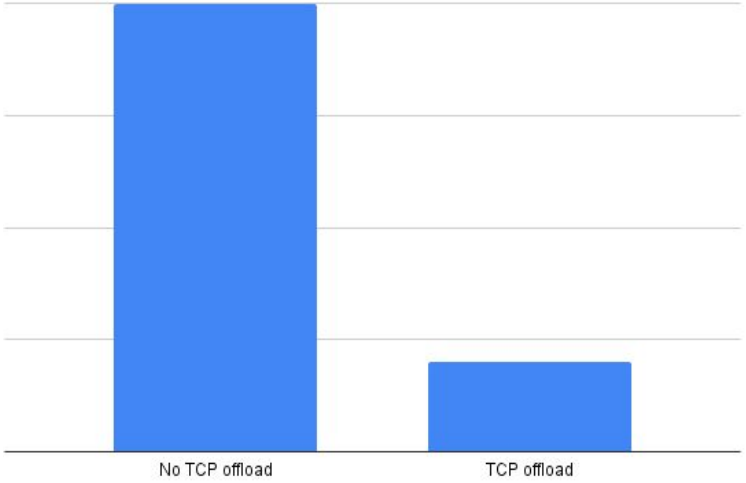
- HW requirement: Shared queues **must** be lossless and in order delivery
- Socket flow control is needed, done by a credits mechanism
- Message are encapsulated in experimental EtherType frame
- Security is derived from AF_XDP sockets framework
- Polling of XTS sockets is described in paper
- No kernel changes, no app changes if with LD_PRELOAD
- Supports zero copy, header data split, packet message steering
- AppCPU can take advantage of HW accelerations

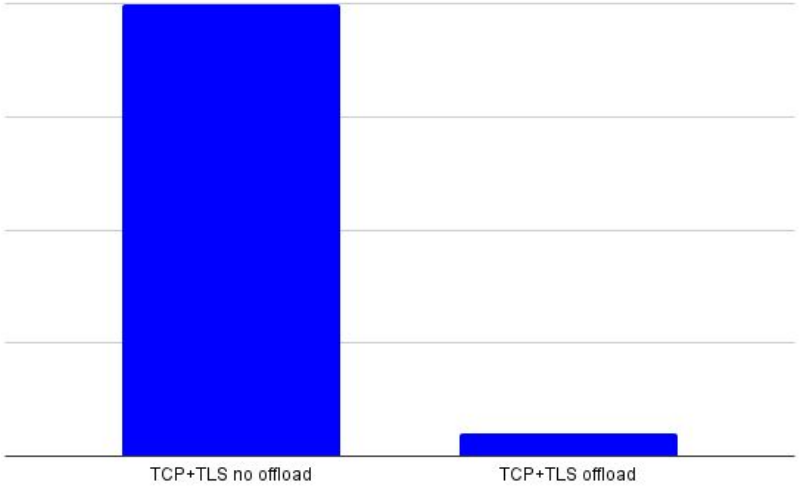
ULP offload

- Create “ULP sockets” to send and receive ULP messages
- Push ULPs onto a stack
- strparser to delineate ULP messages in a stream



Performance*





*Relative performance based on preliminary data

Implementation status and future work

- Working on Bluefield
- xdp_xport library for applications
- Userspace proxy running in SmartNICs
- First version into open source in Q1’24

Future work:

- In depth performance analysis
- Advanced features
- LD_PRELOAD for zero application change
- Support for various ULPs

Thank You!