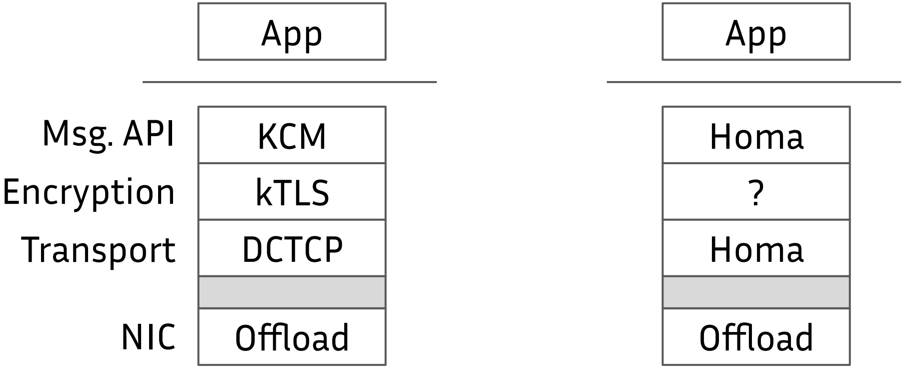
HomaLS: Tunneling Messages through Secure Segments

Tianyi Gao and Michio Honda
University of Edinburgh

Netdev 0x16
October 26, 2022, Lisbon

Motivation

- Datacenter transport needs encryption
  - Third-party network/hardware/software on the paths
- Homa enables low latency and message-native transport



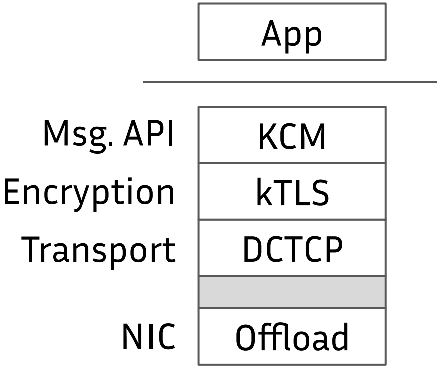
How to get encrypted?

Datacenter networking stack [NetDev 1.2 2016]

It's time to replace TCP in the datacenter [NetDev 0x16 2022]

Motivation

- Datacenter transport needs encryption
  - Third-party network/hardware/software on the paths



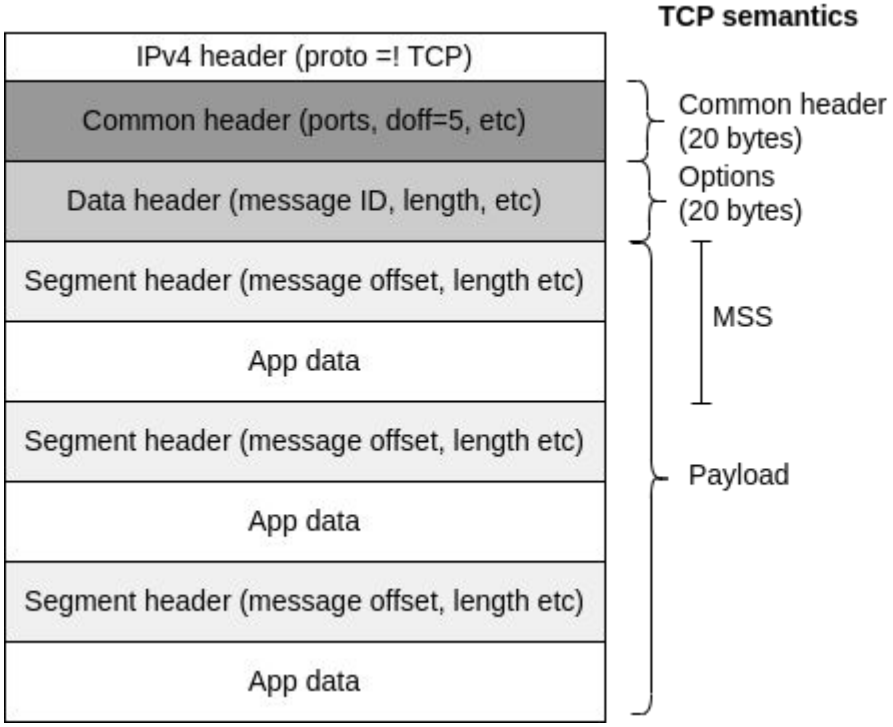
Datacenter networking stack
[NetDev 1.2 2016]

Review: Encryption in transport

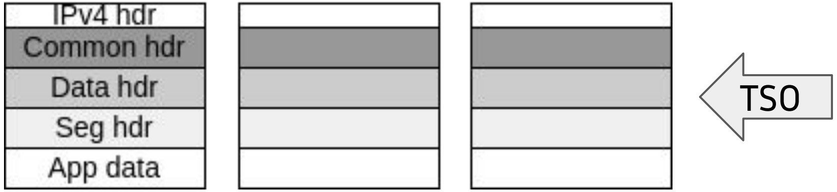
- Ubiquitous encryption
- Encryption at segmentation
- Opportunistic hw offload
- In-kernel users

Homa GSO segment

- Overlay the TCP header
  - Protocol number set to 140
  - “TSO”-able for Mellanox NICs



TCP semantics
IPv4 header (proto =! TCP)
Common header (ports, doff=5, etc)
Data header (message ID, length, etc)
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
Common header (20 bytes)
Options (20 bytes)
MSS
Payload



Encrypting Homa segments

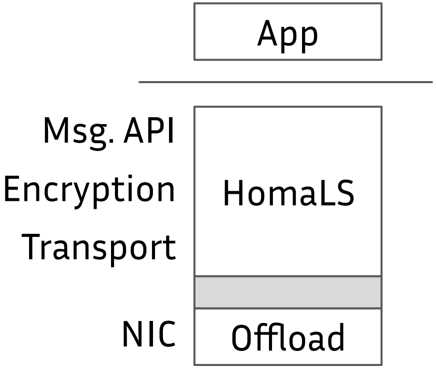
- kTLS offload (TLS_DEVICE) is a candidate
  - Together with TSO
- We need NICs that work for:
  - Non-TCP in the IPv4 header
  - Preserve TCP options
- ~~Chelsio T6~~
- **Mellanox CX-6**

Encrypting Homa segments

- **TLS offloading seems to work for Mellanox CX6**
- Preliminary experiment:
  - Rewrite the proto no. in the IPv4 header in a TCP's GSO segment in the driver (`mlx5e_xmit_core()`), and see what are generated
    - Non-TSO TX command needs a driver patch to avoid `eth_get_headlen()` in `mlx5e_calc_min_inline()`

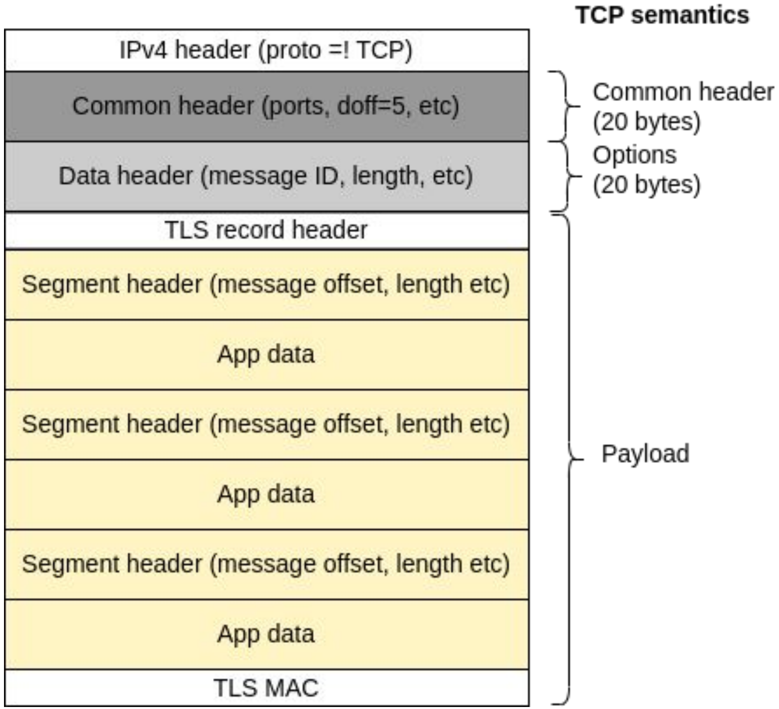
HomaLS (Homa-Layer Security)

- Integrate Homa with TLS



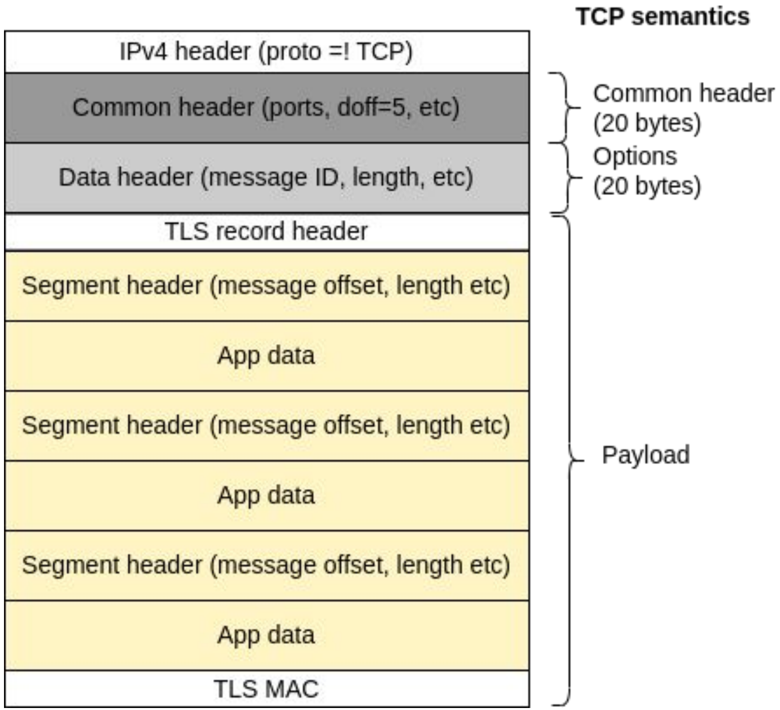
HomaLS (Homa-Layer Security)

- Integrate Homa with TLS
  - Resemble TLS/GSO segment to the hardware (except for proto no.)


TCP semantics
IPv4 header (proto =! TCP)
Common header (ports, doff=5, etc)
Common header (20 bytes)
Data header (message ID, length, etc)
Options (20 bytes)
TLS record header
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
TLS MAC
Payload

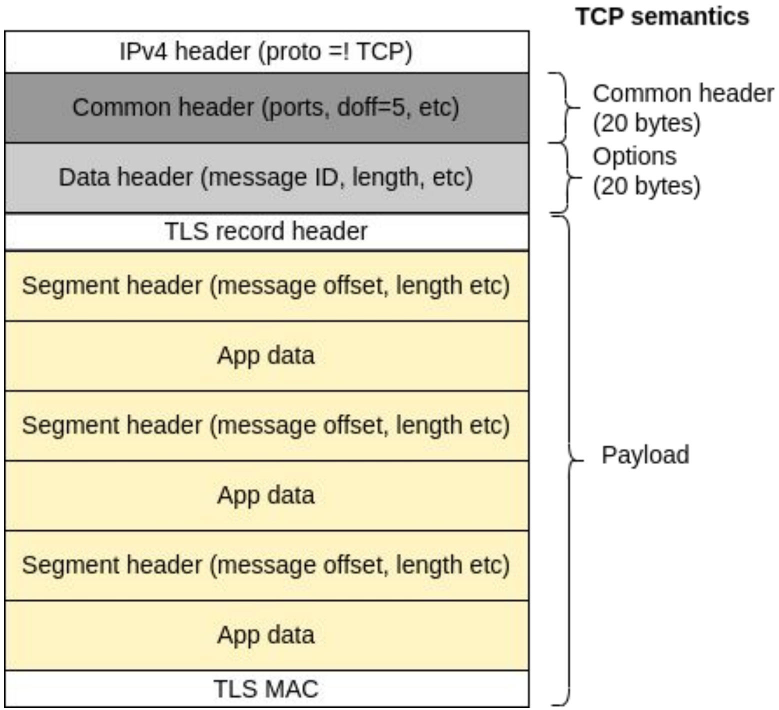
HomaLS (Homa-Layer Security)

- Integrate Homa with TLS
    - Resemble TLS/GSO segment to the hardware (except for proto no.)
    - (for now) IPID for reassembling
        - Will switch to TCP seqno


TCP semantics
IPv4 header (proto =! TCP)
Common header (ports, doff=5, etc)
Data header (message ID, length, etc)
TLS record header
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
TLS MAC
Common header (20 bytes)
Options (20 bytes)
Payload

HomaLS (Homa-Layer Security)

- Integrate Homa with TLS
  - Resemble TLS/GSO segment to the hardware (except for proto no.)
  - (for now) IPID for reassembling
    - Will switch to TCP seqno
  - Per-client (4-tuple) key
    - `setsockopt()` like kTLS

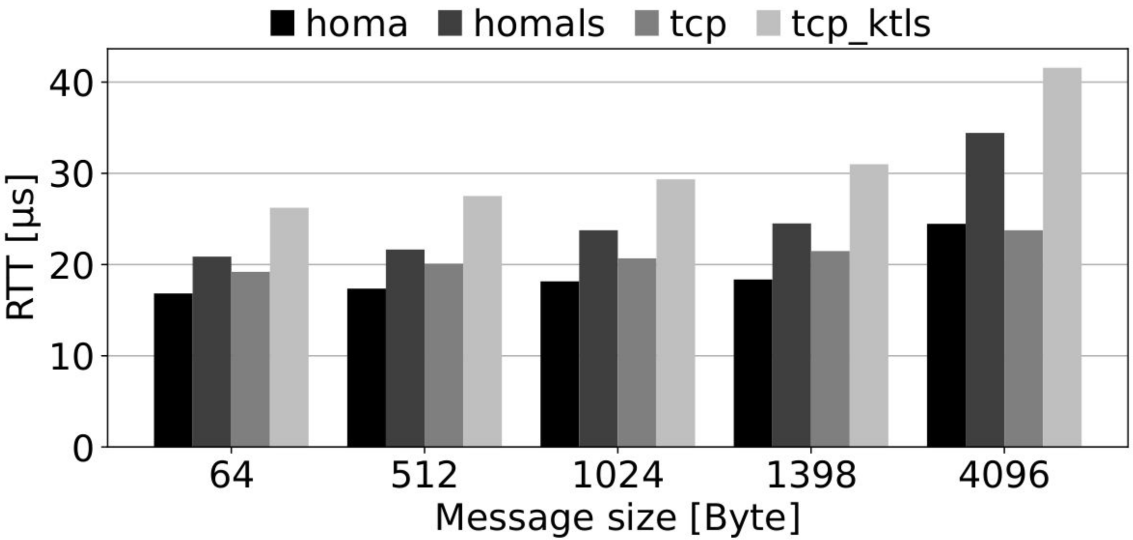

TCP semantics
IPv4 header (proto =! TCP)
Common header (ports, doff=5, etc)
Data header (message ID, length, etc)
TLS record header
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
Segment header (message offset, length etc)
App data
TLS MAC
Common header (20 bytes)
Options (20 bytes)
Payload

Software architecture

- Like kTLS
  - `setsockopt()` to register (exchanged) key to a socket
- Unlike kTLS
  - No ULP - integrated into Homa closely
  - TLS context holds remote address and port
    - Homa socket is one-to-many
- ~800 LoC
  - SW implementation
  - HW offloading is work-in-progress

Performance

- Message RTTs
- SW crypto
  - TSO but no TLS offload
- 17-21% shorter RTT than kTLS/TCP



Two machines equipped with Xeon E5-2640v4 and ConnectX-4 NIC

Summary and discussion

- RX-side offload
- Test more NICs for kTLS offload over non TCP packets
  - Netronome NFP3800, Intel IPUs, any others?
- Use of PSP instead of kTLS
  - When will the hardware offloading be ubiquitous?

https://github.com/uoenoplab/HomaLS-netdev0x16