# BlueField w/ DOCA Flow
## on inspiration of *FlexTOE*

*Guanshujie Fu & Prof. Jialin Li*

# FlexTOE[1]

Flexible TCP offload engine Intro
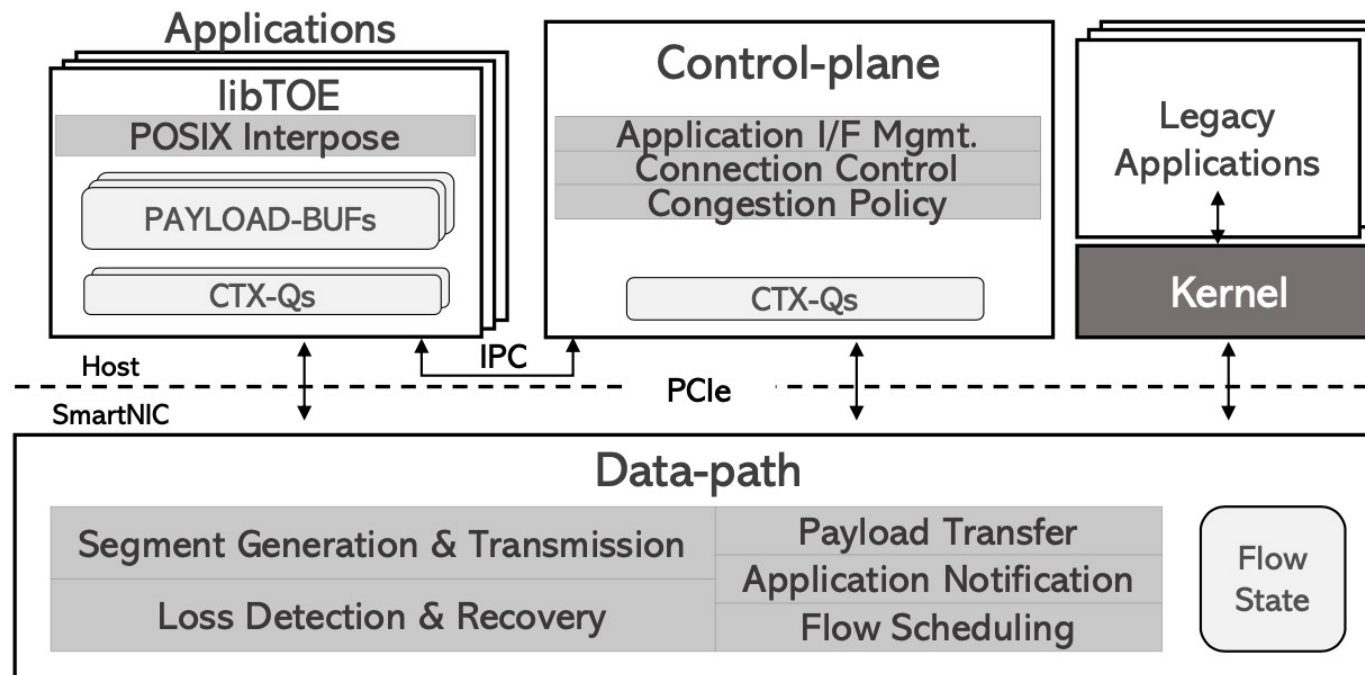
- *Intro*
  - *TCP-offload on SmartNICs[2]*
  - *Flexible, High-performance*

- *Basic Architecture*
  - *Control Plane* ⎤ *Host*
  - *Applications* ⎦
  - *Data Path*



- *Functionality*
  - *Data Path - scalable data transport of established connection*
    - *Flow schedule, TCP Seg generation & transmission, Loss detection & recovery, etc..*
  - *Control Plane*
    - *Congestion control, etc..*
  - *Applications (libTOE)*

# FlexTOE cont'd

Contributions of FlexTOE



- *Design Points*
  - *One-shot data-path*
  - *Modularity*
  - ***Fine-grained parallelism in data path***
    - *3 workflows: Host Control(**HC**), Transmit(**TX**), Receive(**RX**)*
    - *Decompose workflow into 5 stages: **pre-processing, protocol, post-processing, DMA, context-queue***
      - ***Pre-processing:** prepare/filter segment with MAC/IP*
      - ***Post-processing:** handle application interface parameter (e.g. congestion control parameter)*
      - ***Protocol (atomic):** (data-path code) modify state/seq num/…*
      - ***DMA:** fetch data…*
    - *5 stages are designed for **parallelism:** all stages except **protocol** can be executed in parallel.*

- *Contribution*
  - *FlexTOE considered the **hardware features** of its SmartNIC to design the 5 stages*
    - *Flow Process Cores (FPCs)*

1.   https://tcp-acceleration-service.github.io/FlexTOE/
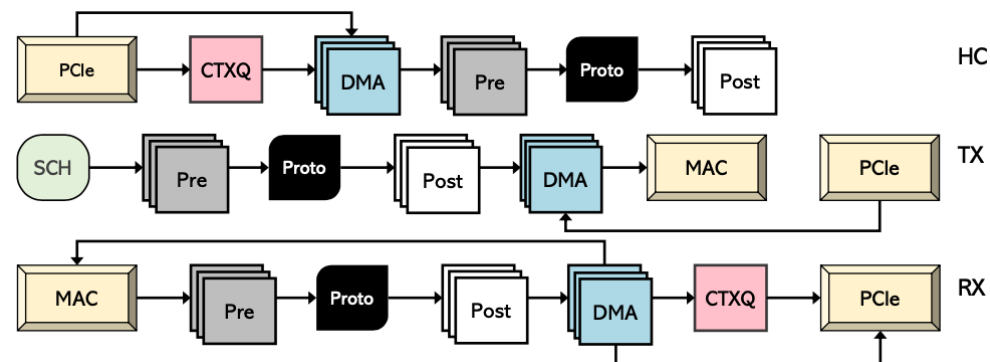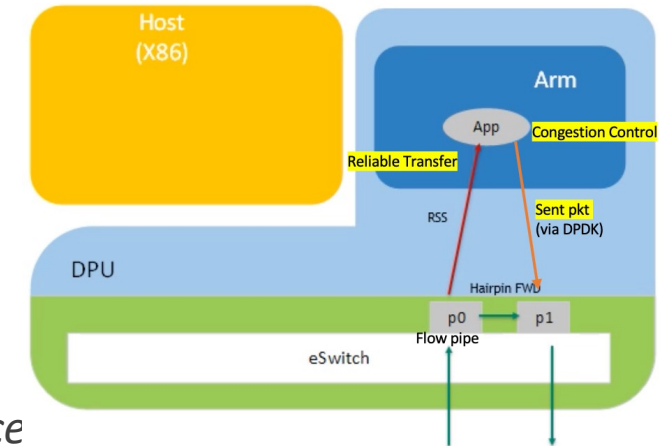
# Inspirations

FlexTOE, Bluefield and DOCA Flow

- *Original Design*
  - *In original sketch, the TCP-offload:*
    - *Has no clear definition for data-path/control plane*
    - *Mainly consider Receive(RX), while Transmit(TX) is ambiguous*
    - *Application in x86 Host is not included*

- *What can be considered*
  - *Compared to FlexTOE, Bluefield has* <u>wimpy **Arm core**</u> *and* <u>Hardware Acce</u>
    - ***Arm core*** *will be responsible for* ***partial control plane and data-path****:*
      - ***Re-transfer, Congestion Control, Application Notify, etc.***
    - ***Flow pipe*** *will be responsible for* ***partial data-path****:*
      - ***Loss detection, TCP ACK, TCP Sequence #, etc.***
  - ***x86 Host*** *plays a role*
    - ***Applications*** *run on Host*
      - ***Transfer payload data to Bluefield, Notify Bluefield, etc.***

# Inspirations cont'd

Bluefield and DOCA Flow

- *Scheme*
  - *Clear Stages*
    - **pre-processing, protocol, post-processing, DMA...**
  - **Arm core** *takes the main place*
    - *TCP segment generation & (re)transmission*
    - *Congestion Control*
    - *Application Notification*
  - **Flow pipe parallelism**
    - *Potential parallelism in hardware acceleration*
- *Problems*
  - *Due to limited time, many details are to be considered*
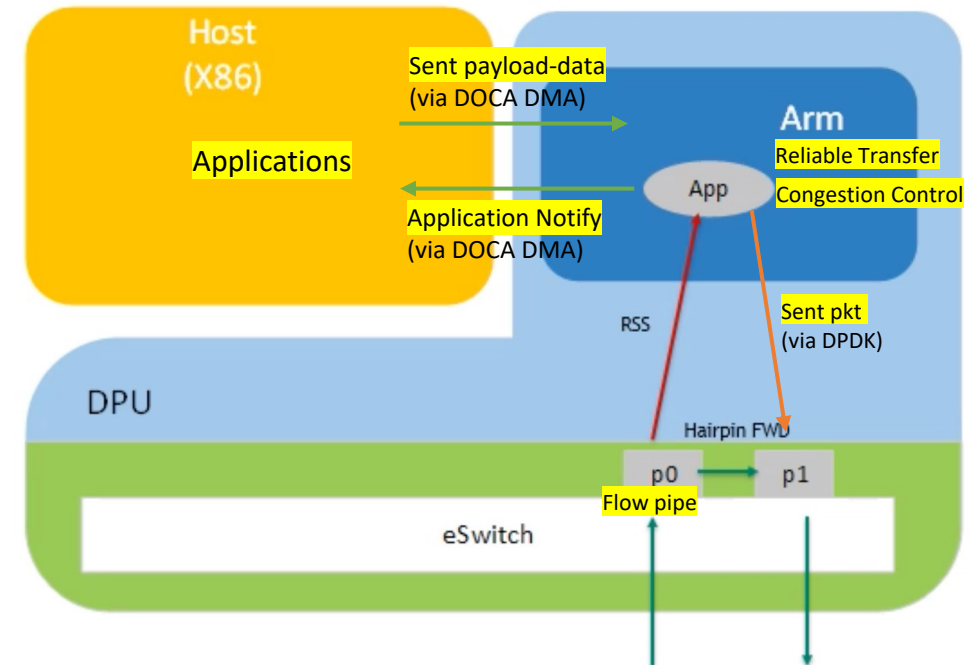
# Thank you

*Guanshujie Fu & Prof. Jialin Li*