SONiC DASH on Intel® IPU:
Implementation and Performance

Speakers:
Shweta Shrivastava (Shweta.Shrivastava@intel.com)
Namrata Limaye (Namrata.Limaye@intel.com)

Key Contributor(s):
Cristian Dumitrescu (Cristian.Dumitrescu@intel.com)

Outline

- SONiC-DASH for Smart Switches
- Introduction to Intel IPU
- Intel Implemented Features and Scale
- DASH Pipeline Overview
- Intel DASH Solution
  - Overall Architecture
  - P4 Implementation
- DASH Performance on Intel® IPU E2100
- Looking Ahead

SONiC-DASH for Smart Switches

What is SONiC DASH?

- Disaggregated API for SONiC Hosts
- Builds upon SONiC (a NOS)
- Leverages programmable network hardware
- Suitable for multiple use cases, including smart switches

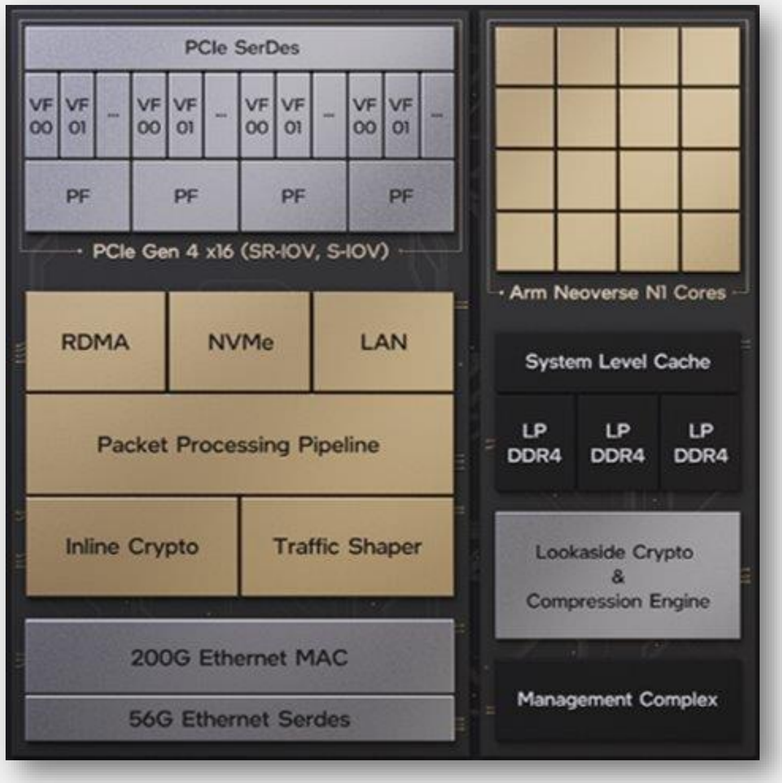
Usecase: Smart Switches

- Smart Switch = Switch + Smart NICs
- Switch: “simple” processing (fast); NICs: complex processing (slower)
- Add intelligence to the DC switch rather than the server

DASH Implementation

- DASH is open source
- Functionality is defined in P4 (BMv2 ref model)
- Enables bidirectional communication between geographically apart VMs
- Implements stateful connections, 5-stage advanced ACLs
- Adv features: HA, Private Link and Service Tunneling

Intel® Infrastructure Processing Unit SoC E2100



Hyperscale Proven

- Co-designed with Google
- Deployed in a wide range of cloud instances (general compute, AI/ ML, storage) today
- Security and isolation from the ground up

Technology Innovation

- Programmable Packet Processing Engine
- NVMe storage interface
- Reliable Transport (Falcon) for lossy fabrics
- Advanced crypto and compression acceleration

Software

- P4 Studio support
- Open-source software including IPDK, DPDK and SPDK
- Enable broad adoption of IPUs

Intel-Implemented DASH Baby Hero Features and Scale

Features

- Vnet2Vnet
  - Outbound routing
  - Inbound routing
- Connection Tracking with Ageing
  - Flow tables with 32M bidirectional flows
  - TCP State Machine
- ACL rules
  - 5 stages per direction
  - Exact match and prefix match support for Baby Hero rules
  - Actions: Permit, Deny, Permit and continue, Deny and continue
- Underlay routing

Baby Hero Scale

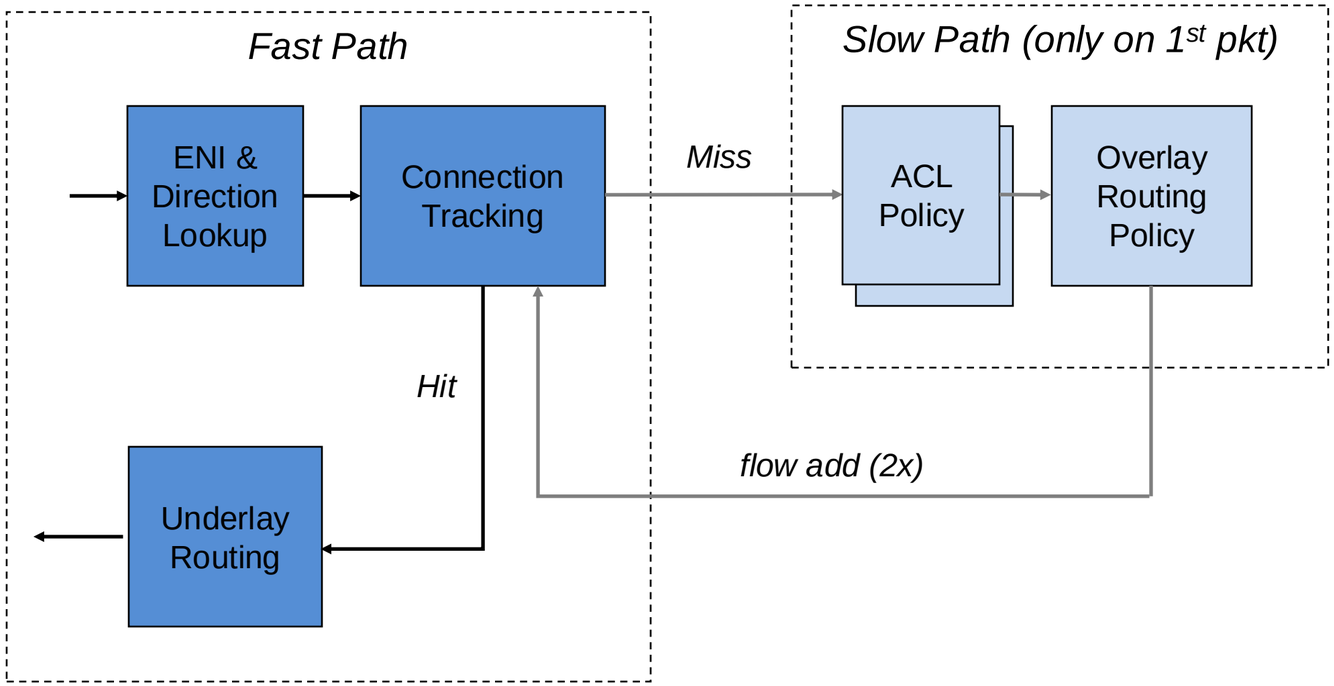
	Scale
ENIs	32
Total Outbound Routes	160K
NSGs	32
ACL prefixes	320K
CA-to-PA Mappings	80K
Total Inbound Routes	32
Active Connections/Flows	32M bidirectional
CPS	3M
Background Flows TCP	15M
Background Flows UDP	15M

DASH Hero Scale Requirements

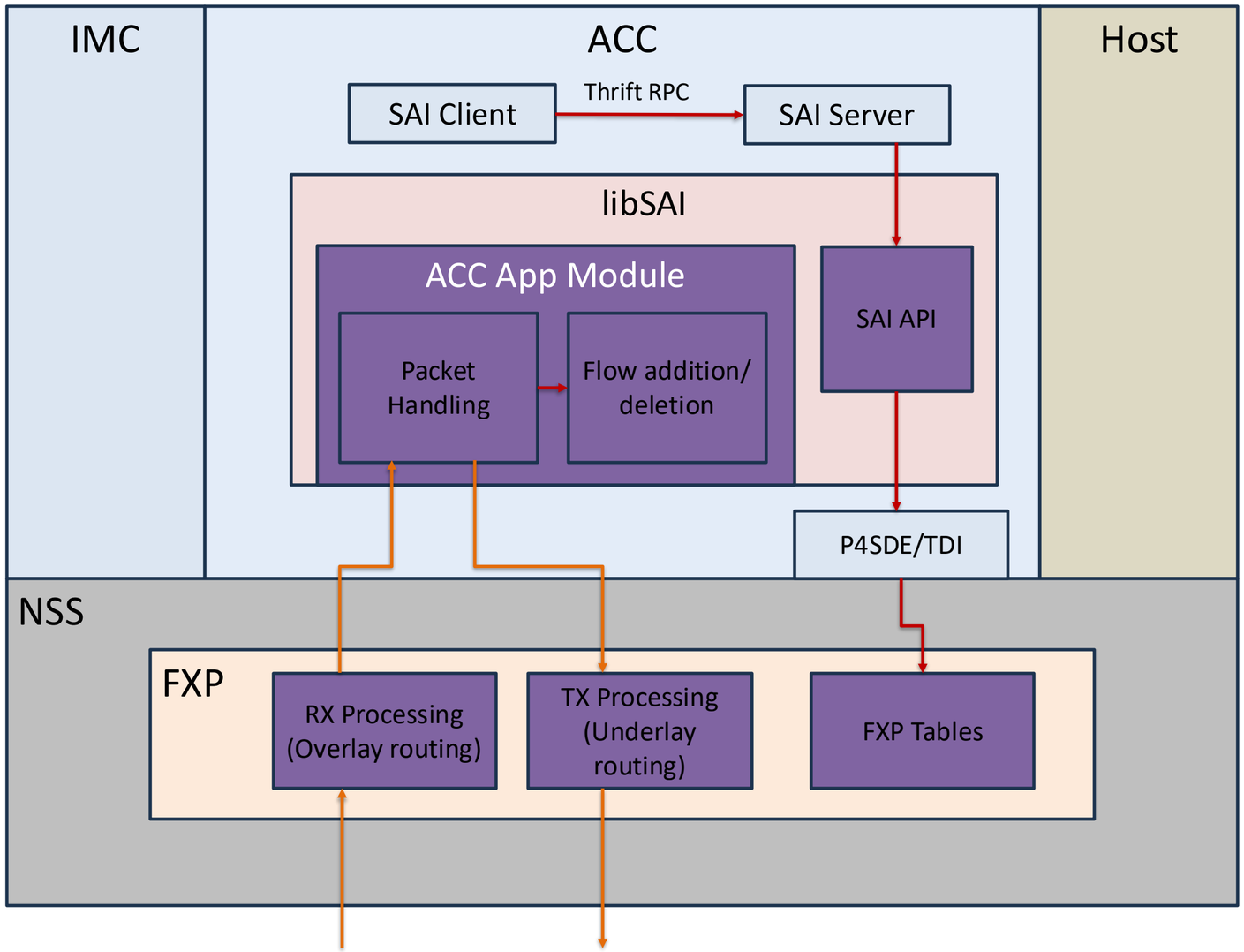
	Scale
ENIs	32
Total Outbound Routes	3.2M
NSGs	320
ACL prefixes	32M
ACL ports	3.2M
CA-to-PA Mappings	8M
Total Inbound Routes	320K
Active Connections/Flows	32M bidirectional
CPS	3M
Background Flows TCP	15M
Background Flows UDP	15M

SONiC DASH Pipeline Logical View

- First use-case (called Vnet2vnet) is essentially a virtual switch allowing geographically apart VMs to communicate.
- Pipeline is connection tracking centric: once a new flow is detected (1st pkt misses flow table), IF policy allows (ACL), THEN 2x flows are added: 1x for forward traffic and 1x for reverse traffic.
- Policy: 5-level ACL tables.
- ConnTrack: TCP state machines (SYN -> FIN) & UDP for up to 32M connections.

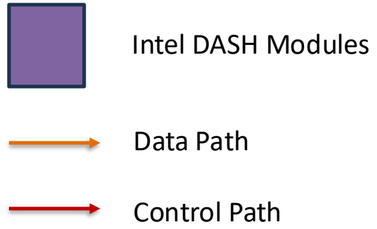


Architecture of Intel DASH Solution

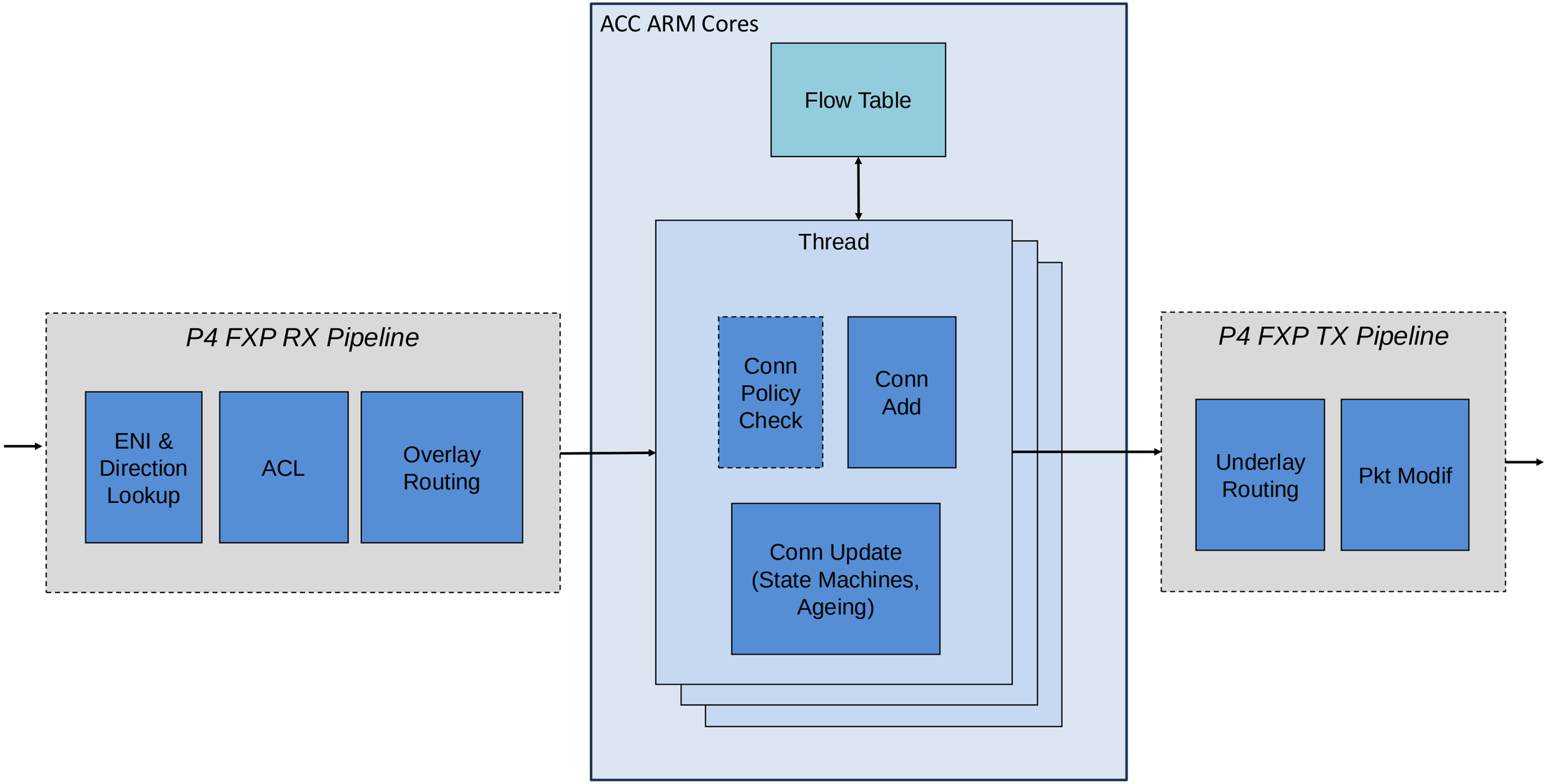


Dash Pipeline Components

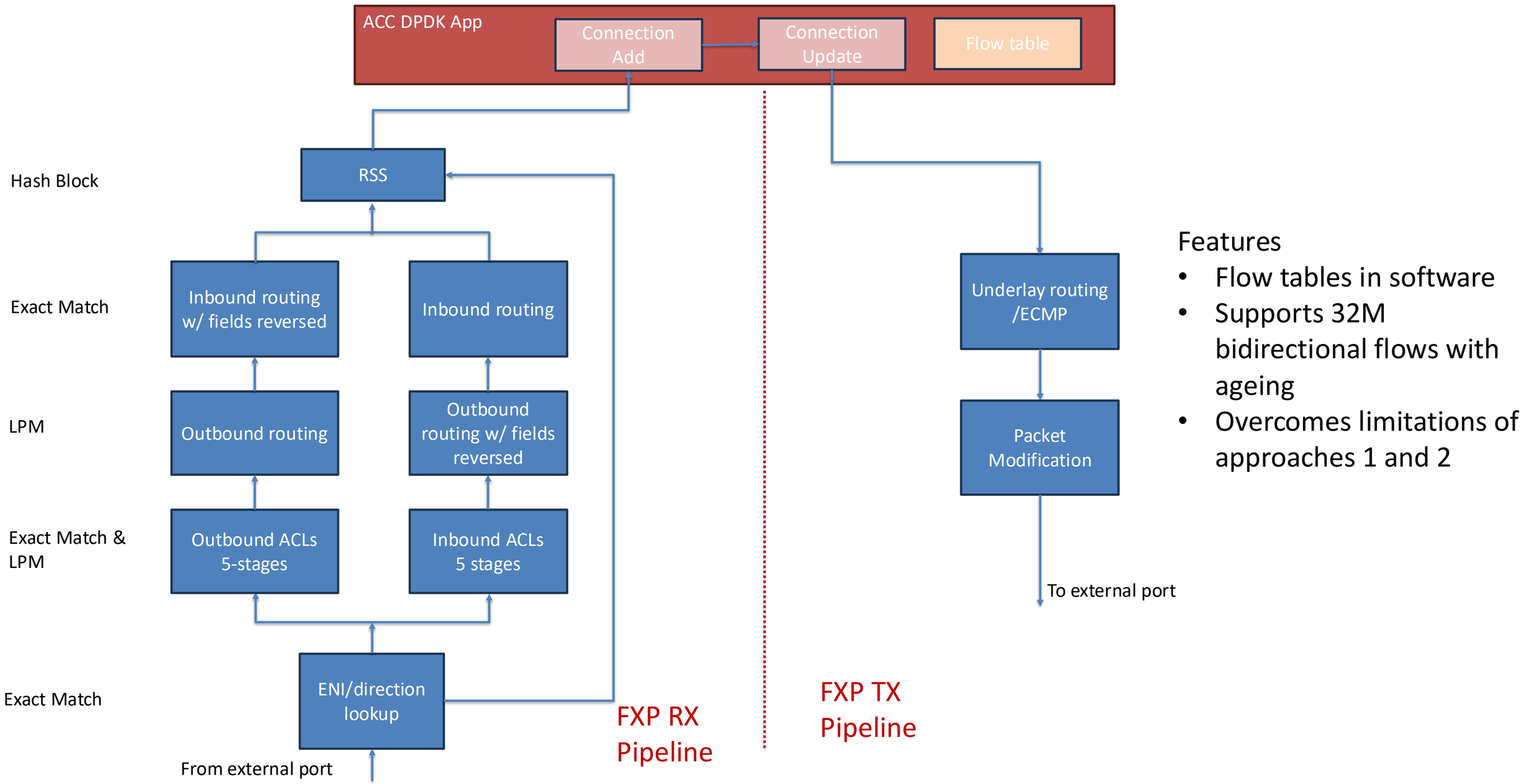
- **Dash P4 pipeline** that implements inbound/outbound and underlay routing, CA-to-PA mapping and flow tables (support for 32M flows bidirectional flows)
- **ACC App** with handling of slow path and fast path packets, TCP state machine, adding of flows to hardware and ageing (of fast path packets).
- **Custom SAI API** integrated with P4 pipeline to program entries into hardware tables via TDI calls.



SONiC DASH Pipeline Implementation on Intel IPU



P4 Implementation



Features

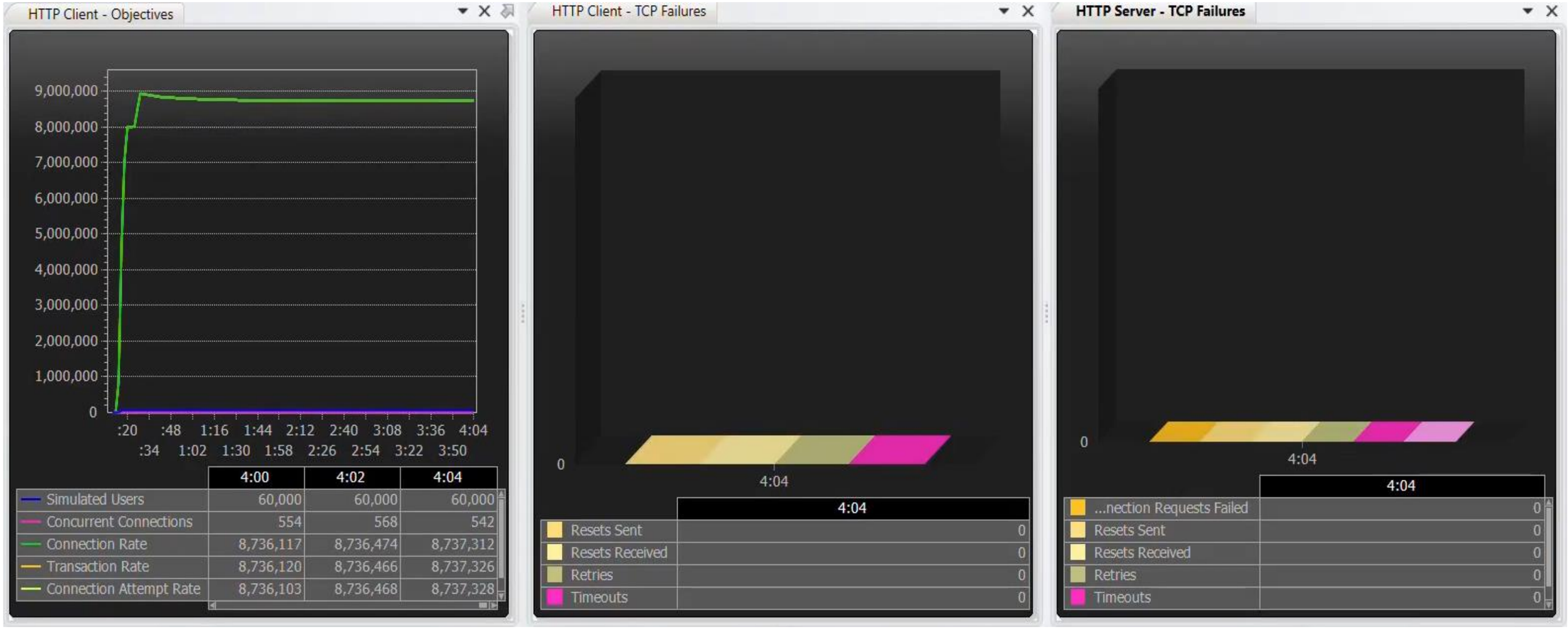
- Flow tables in software
- Supports 32M bidirectional flows with ageing
- Overcomes limitations of approaches 1 and 2

Performance Results

Max TCP CPS without background flows

- Target CPS rate: 10M
- Test duration: 240 seconds
- Number of ARM cores use on ACC: 12
- Packets Lost : 0

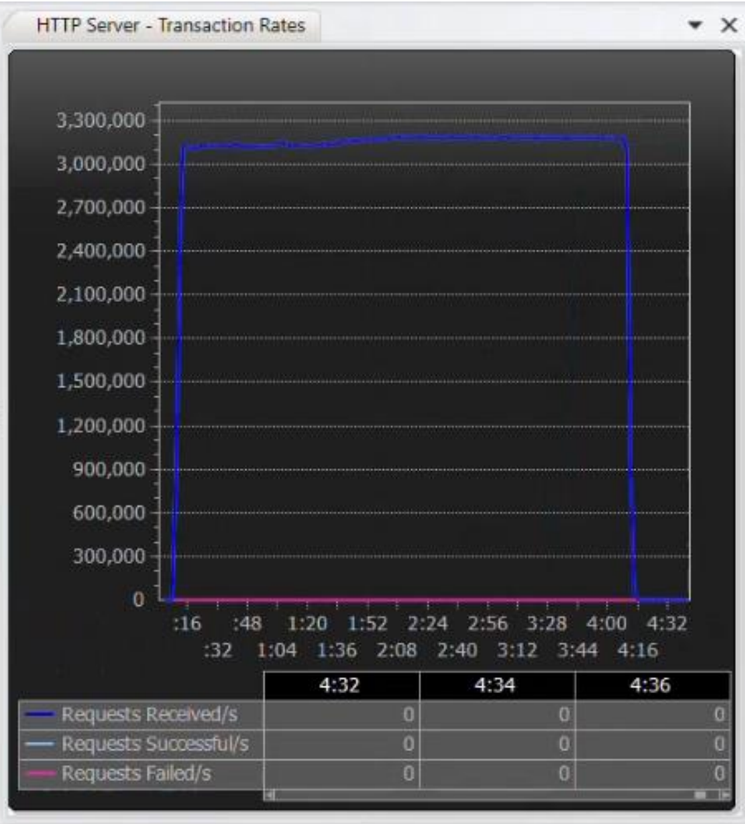
Max CPS: 8.7M



TCP CPS with background flows

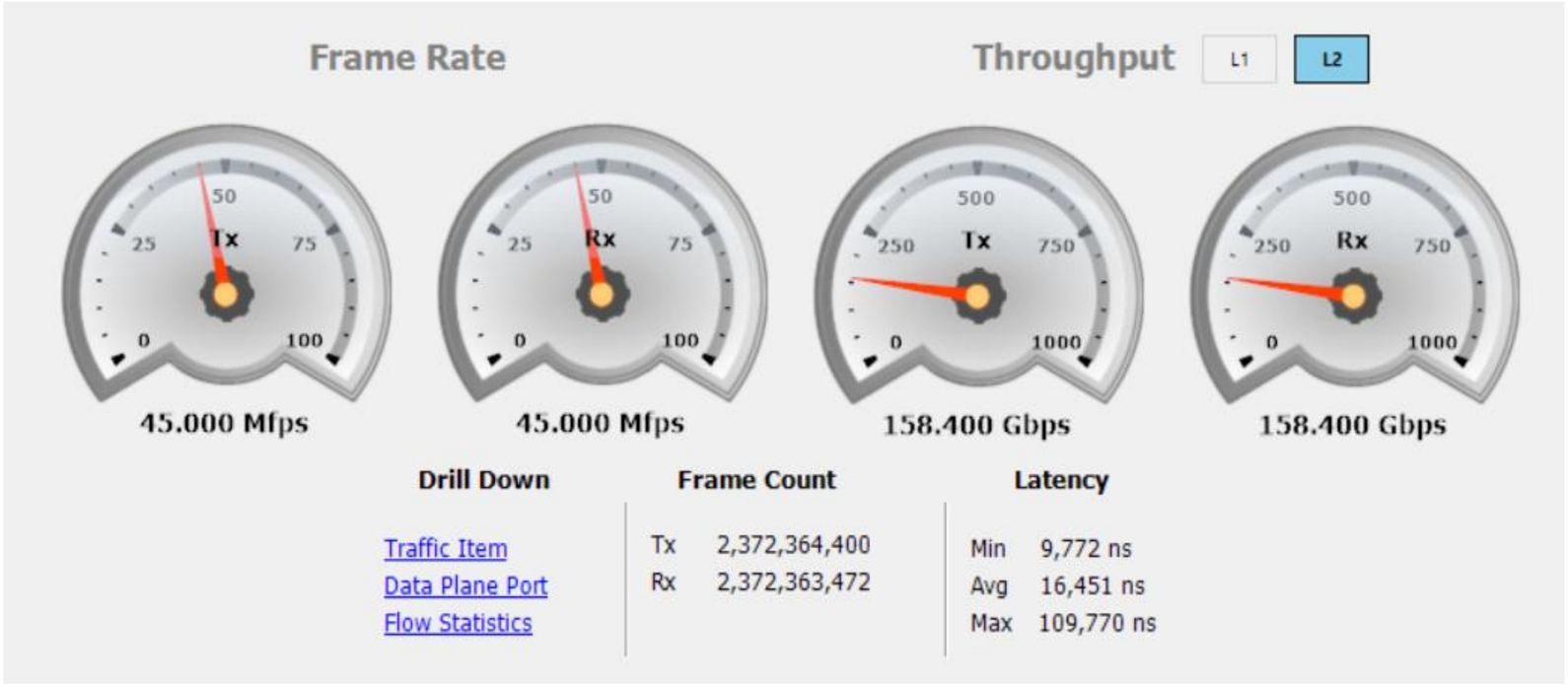
- #Background flows: 15M TCP + 15M UDP
- Ageing time for bg flows: 10 sec
- Test duration: 240 seconds
- Packet Loss: 0.001%

CPS: 3.2M



Throughput and Latency

- Total number of flows: 320K
- Number of ARM cores use on ACC: 12
- **PPS Rate: 45M**
- **Throughput: 158 Gbps**



Result Summary

Test Scenario	Results
Max TCP CPS w/o bg flows	8.7 M
TCP CPS w/ bg flows (15M TCP + 15M UDP bg flows)	3.2M
PPS	45M PPS
Throughput (with 440-byte packets)	158 Gbps

Looking Ahead

- Flow table implementation in hardware using P4
- P4 DPDK software implementation

- Hero Scale
- High Availability
- SONiC Integration
- Service Tunnel
- Private Link

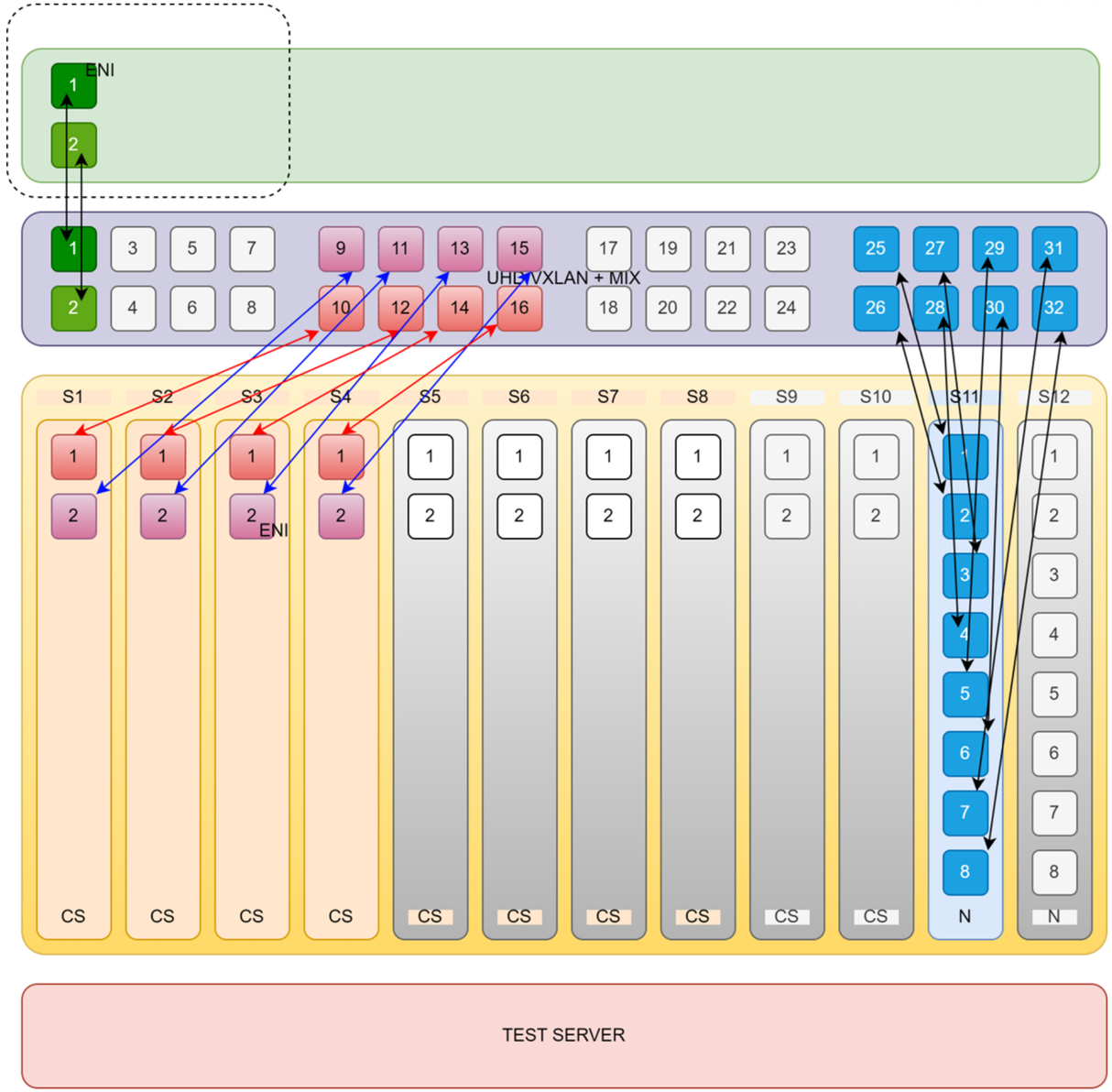
Notices and Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details.
- Intel technologies may require enabled hardware, software or service activation.
- Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.
-

Thank You

Backup

Test Setup



Intel® IPU under test details: Intel® IPU Adapter E2100-CCQDA2 - 2x100G; power management state 0; MEV-TS release 1.5.0.8751 ***note: this is a bump-in-the-wire use case, so the host server is not involved in traffic flow*