FTF | FREESCALE TECHNOLOGY FORUM
POWERING INNOVATION

Application Specific Fast-Path for QorIQ and QorIQ Qonverge

FTF-ENT-F0012

Hemant Agrawal
Software Architect

August 2012

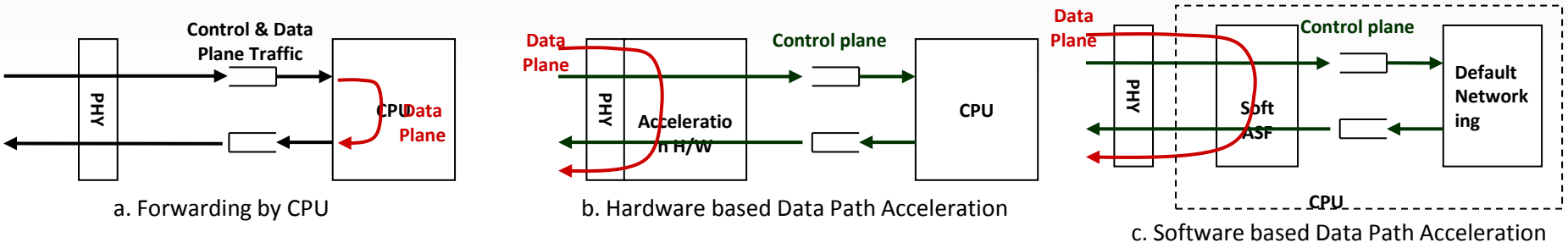
Agenda

- **Application Specific Fast-Path (ASF) concepts and architecture**
  - Features
  - Use-cases
- **ASF internals**
  - IPv4 Forwarding
  - NAT Processing
  - IPSEC Processing
  - IP Termination
- **Performance**
  - Benchmarks on DPAA and non-DPAA
  - Leveraging hardware acceleration
- **Conclusion**

Faster Packet Processing

- All flows are created equal …
- … But some flows can be put on a fast-track
- Store flows requiring simple, deterministic processing in a cache
- Recognize cached flows and process such packets in a separate highly optimized context

What is Fast-Path?



a. Forwarding by CPU

b. Hardware based Data Path Acceleration

c. Software based Data Path Acceleration

- Most network packet processing can be broken down into two paths
  - Fast path, also known as the data path, requires quick and efficient switching/routing of packets
  - Slow path, also known as the control path, requires more processing and has more inherent latency than the data path
- Application Specific Fast-Path
  - Optimized software implementation for Data Path
  - Customized for QorIQ platforms
  - Achieve higher throughput for specific applications
  - Leverages functionality provided by H/W

Why Application Specific Fast-Path?

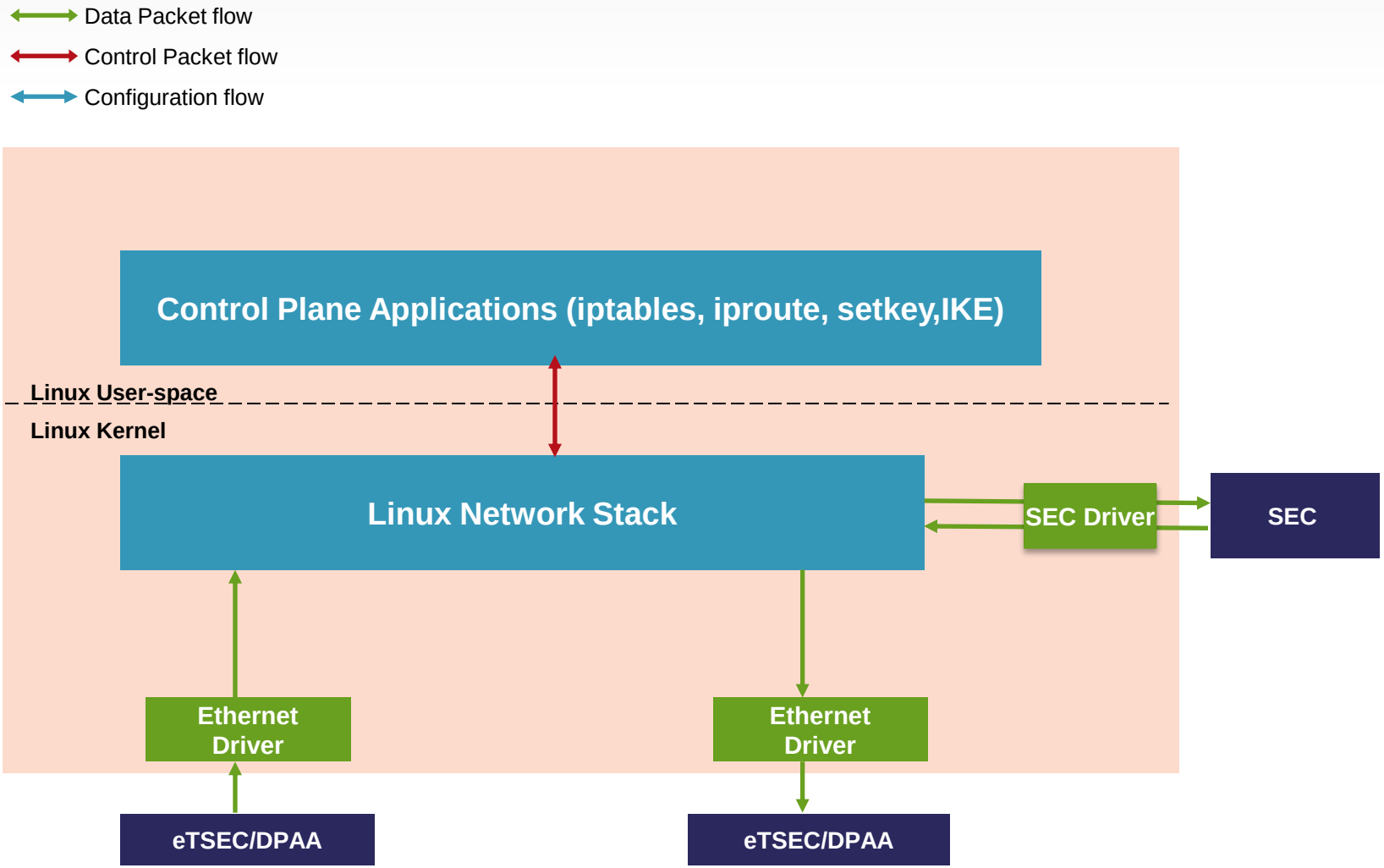
- **What is Fast-Path?**
  - Offloads data-path processing from a conventional network stack to a highly optimized context
  - Tightly coupled with hardware accelerators/low-level drivers
- **Why do we need it?**
  - 2x-8x speedup over conventional network stack
  - Better leverage of our HW acceleration capabilities
  - Make extra headroom in CPU for other tasks
- **Why do it in software?**
  - No need for specialized HW engines – save die-size/power
  - Open source C code in Linux – easy to compile, integrate, debug and customize
  - Bundled with SDK – out-of-the-box experience
- **What is our differentiation?**
  - Seamless integration with Linux® & VortiQa*
  - More features – IPSec, Zero-copy termination

freescale™

Fast-Path for Different Applications

Application	Fast-Path	Slow-Path
Layer 2 bridging	FDB lookup, VLAN add/delete, Learning	Aging, STP
IPv4 forwarding	Dest-cache lookup, L2 modify	LPM route-table lookup, ARP
NAPT	5-tuple lookup, IP/Port/L2 modify	Connection setup/destroy, policy, ALG
Firewall	Access control list, pin-holes	Stateful packet inspection, ALG
IPSec	5-tuple lookup, encap/decap + crypto	SA setup, security policy
QoS	Enforcement – sched, police, congestion, shaper	Policy, provisioning

Linux Network Processing

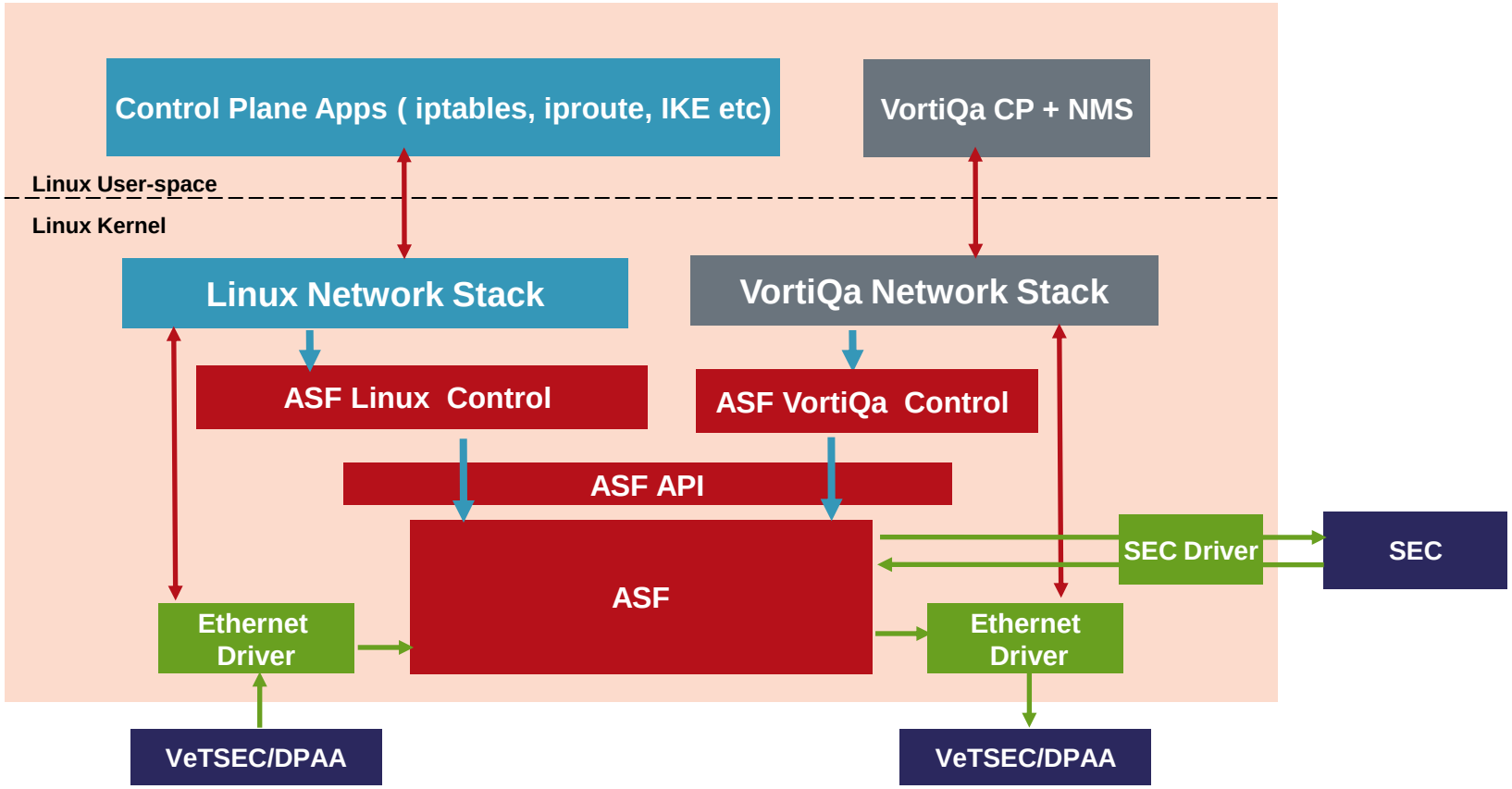


Linux Network Processing with ASF

Data Packet flow

Control Packet flow

Configuration flow



Target Application Segments

- **Low-End gateways**
  - E.g. Multi-Service gateway, SOHO/SMB, WLAN AP
  - Applications – Routing, L2 switching, NAPT/Firewall, QoS
- **Security appliances**
  - Enterprise UTM, VPN gateway
  - Applications – IPSec, NAPT/Firewall
- **Wireless backhaul**
  - Femto, Pico, Macro cells
  - Applications – IPSec, Zero-copy termination, QoS
- **Edge/Access**
  - DSLAM, SGN
  - Applications – IPSec, NAPT/Firewall, QoS, IPv6, VSG

ASF Feature List (SDK 1.2 & WUSDK)

- **Core Features**
  - IPv4 Forwarding
  - NAPT/Firewall
  - IPSec (ESP/AH-Tunnel, NAT-T)
  - Fragmentation & re-assembly
  - Zero Copy local termination to user-space
  - Per-flow statistics and aging
  - IPv6 Forward, NAPT/Firewall (Off-train)
  - QoS – policing, shaping, scheduling (HW).

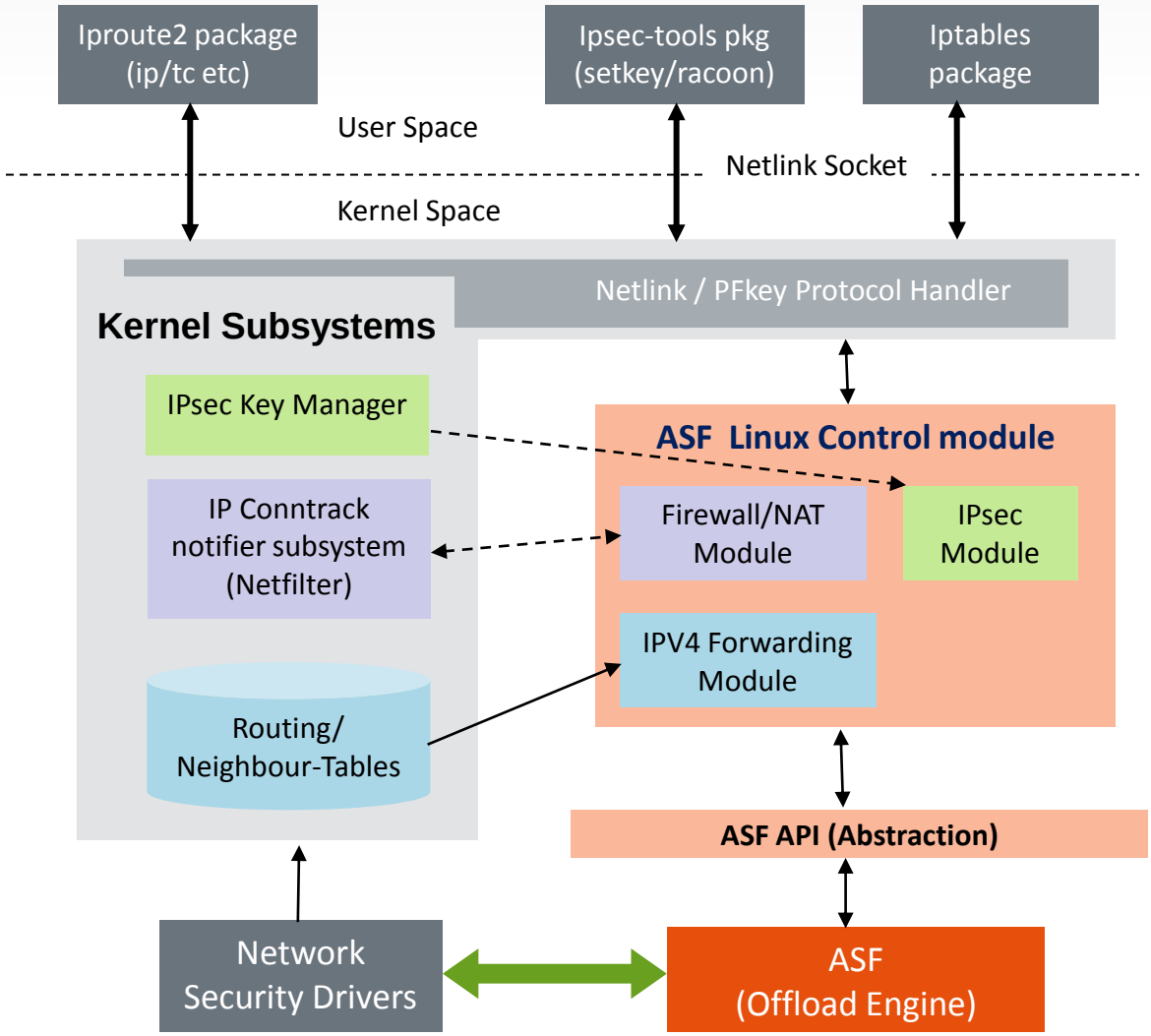
- **Platform Integration**
  - Platforms
    - P102x, P101x, P202x
    - P4080, P1023, P3041, P2040, P5020
    - PSC9131/32
  - IP
    - VeTSEC, eTSEC
    - SEC 3.x, SEC 4.x
    - DPAA – FM/BM/QM

- **Network Stack Integration**
  - Linux (2.6.33, .34, .35 & 3.0)
    - Conn-track + IP-Tables (NAT/Firewall)
    - IPv4 Forwarding + IP Toolkit
    - Native IPSec Stack + Racoon/Strong-Swan
    - PMAL Socket (IP-Termination)
    - Real Time Kernel Patch
    - IPv6 (Off-train)
    - TC integration for QoS
  - VortiQa
    - VortiQa NAPT/Firewall
    - VortiQa IPSec
    - VortiQa IPv6 (Off-train)

- **Interface Support**
  - Ethernet
  - VLAN
  - PPPoE

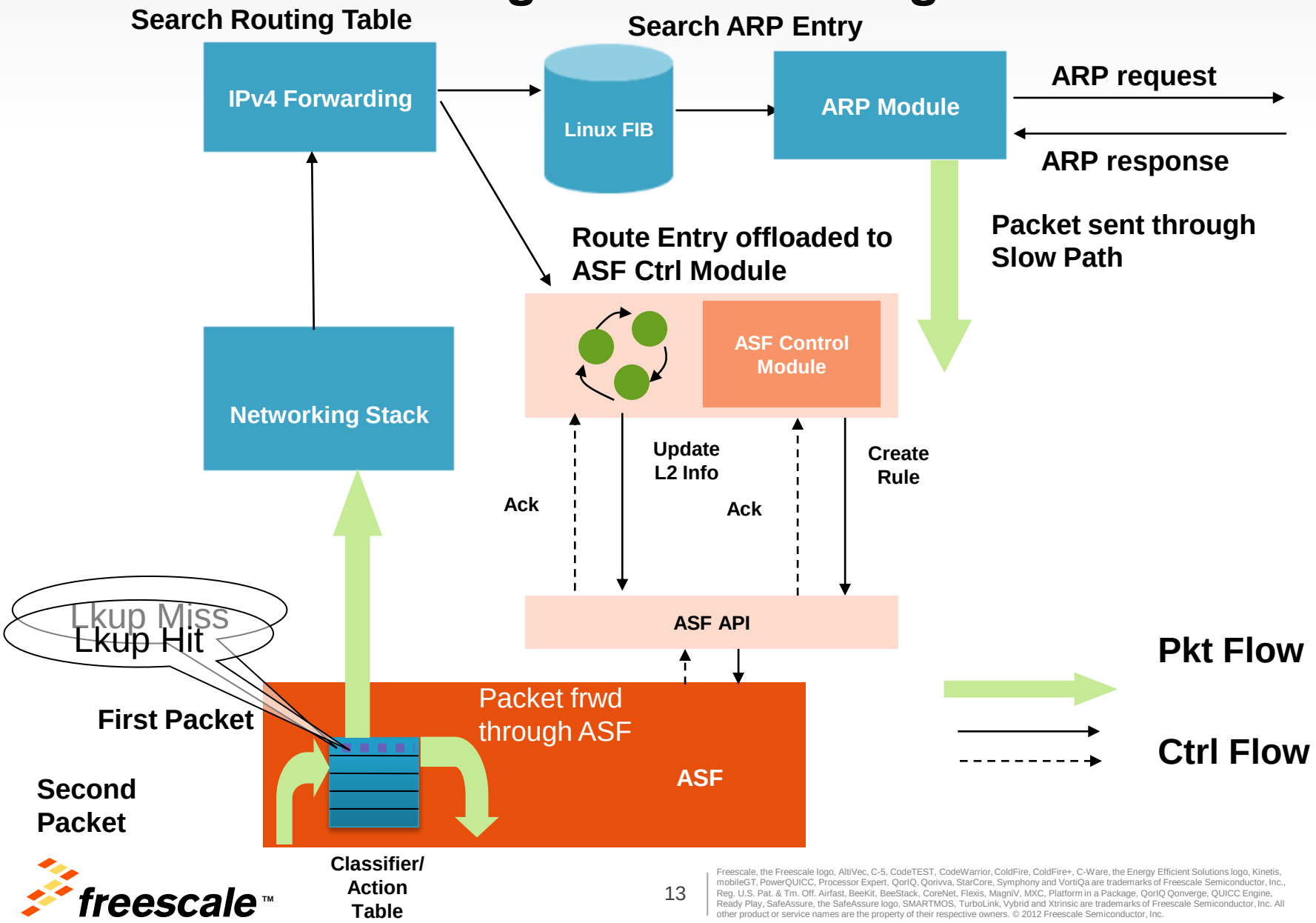
freescale™

ASF Control Linux Integration

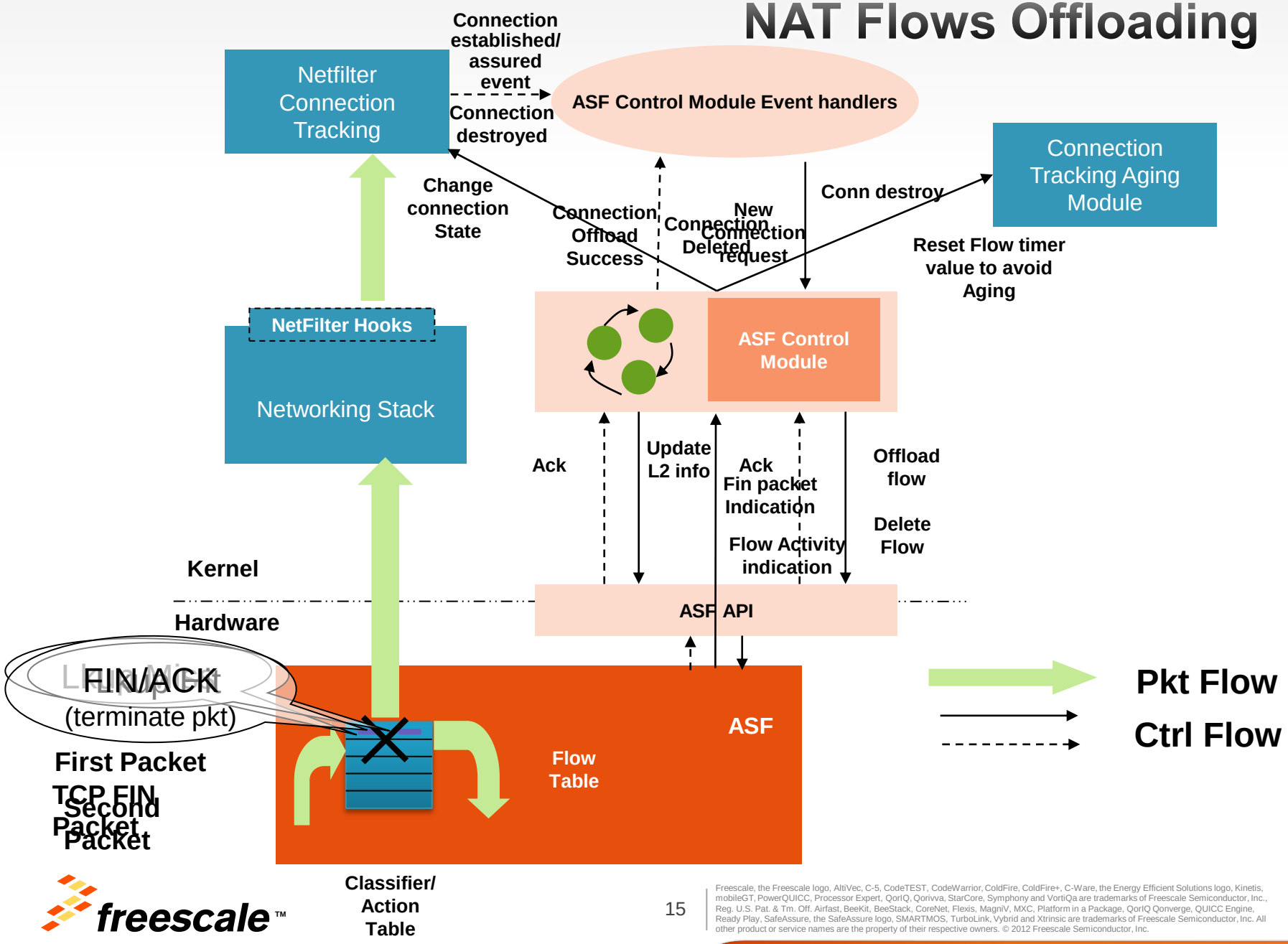


- Seamless Integration with Linux Network stack
- Standard Linux tools for configuration
- Uses existing notification mechanisms
- Control module adds logic for offload
- ASF-API abstracts actual fast-path implementation

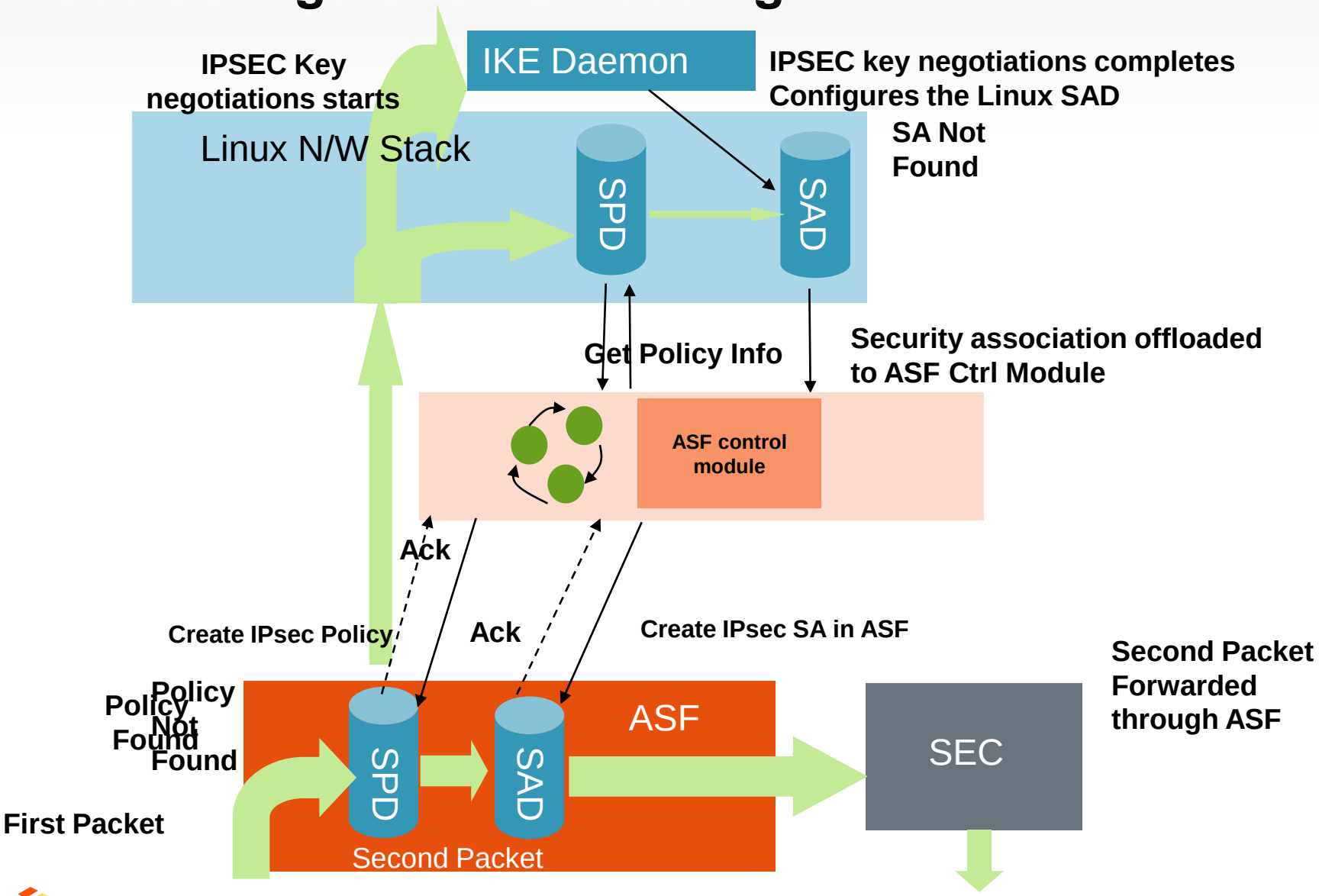
ASF IPv4 Forwarding Flow Offloading



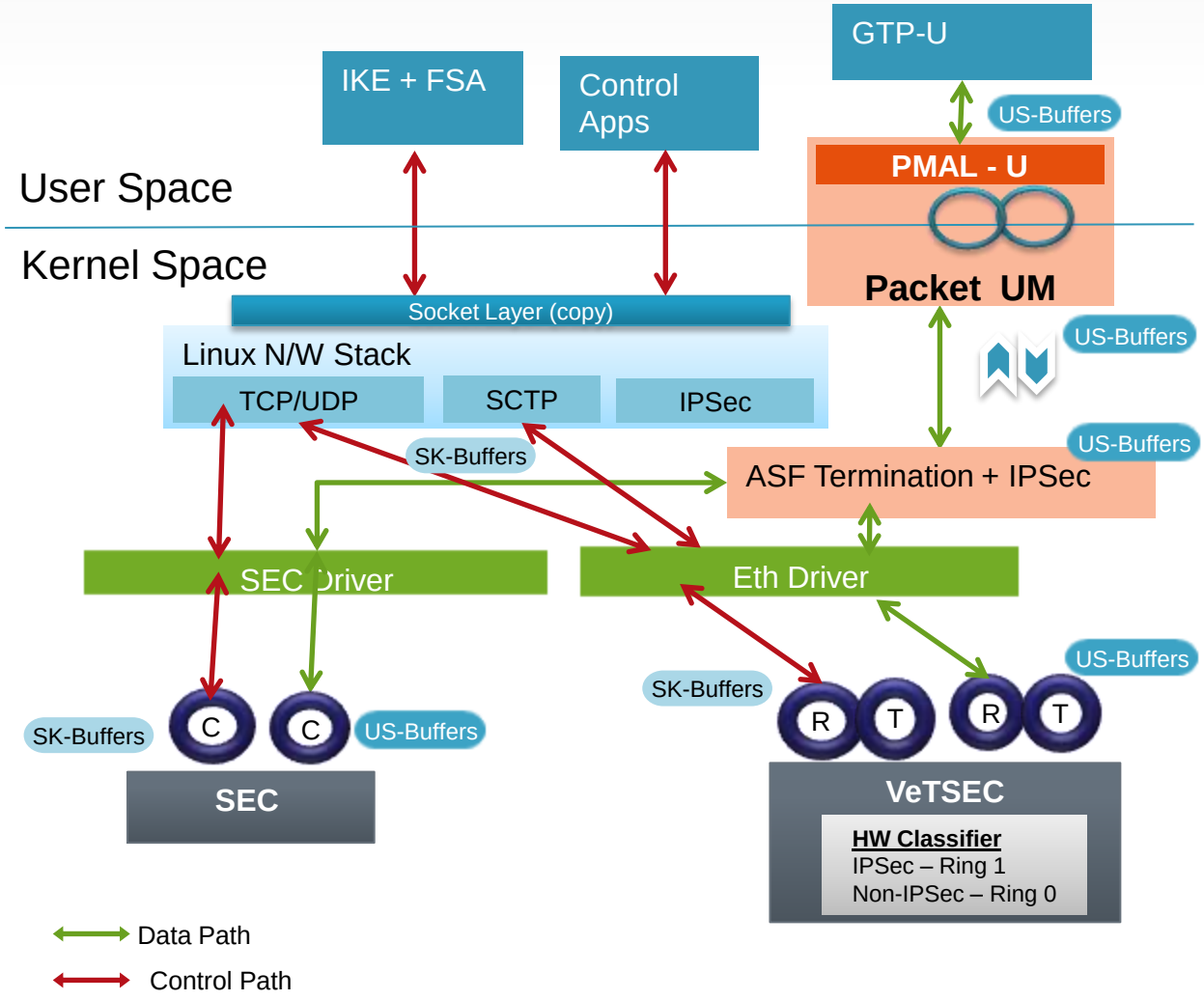
NAT Flows Offloading



IPSec Configuration Offloading

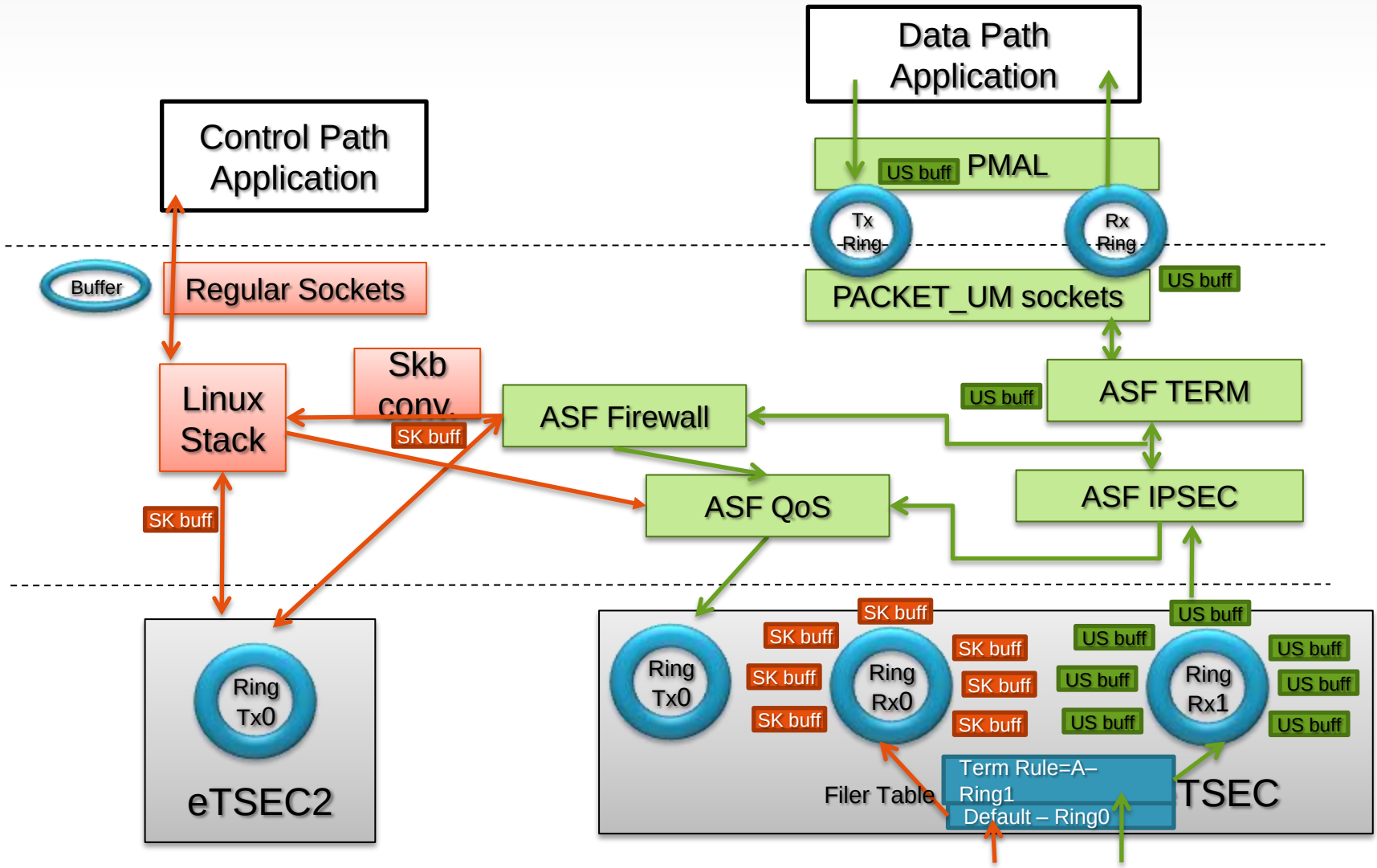


ASF Local UDP Termination – with IPSEC



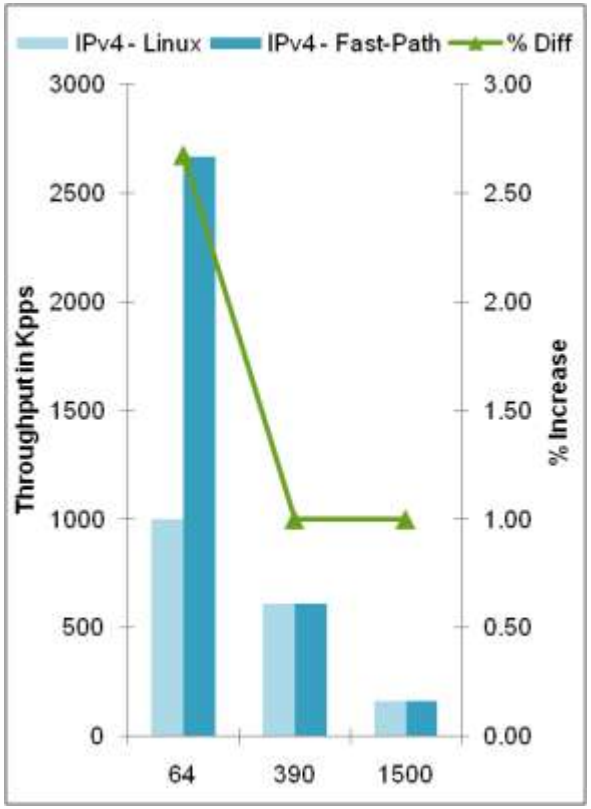
- Zero-copy packet termination to user-space.
- Standard sockets with Packet-UM options.
- Multiple applications & flows using 5-tuple.
- All features of ASF Firewall, IPSec, QoS.
- Uses multiple BD-rings & classification in hardware.

ASF QoS support

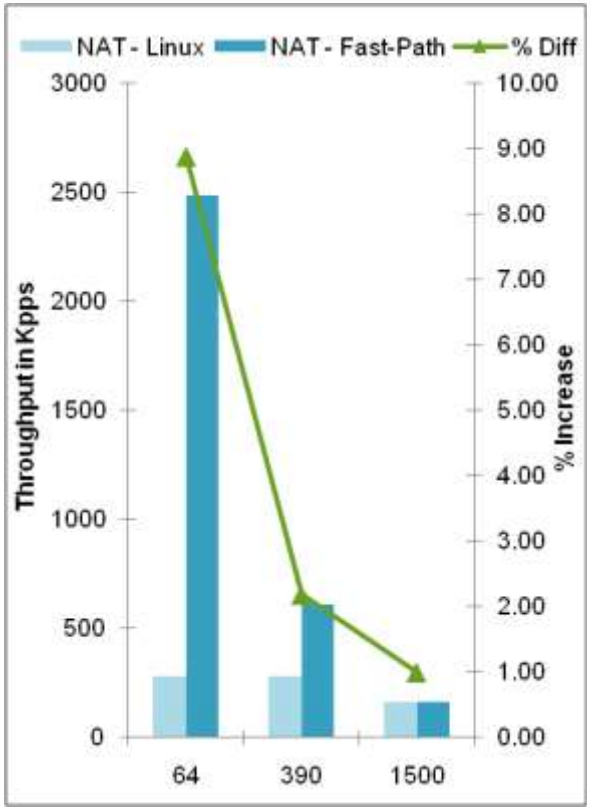


ASF Performance Advantage

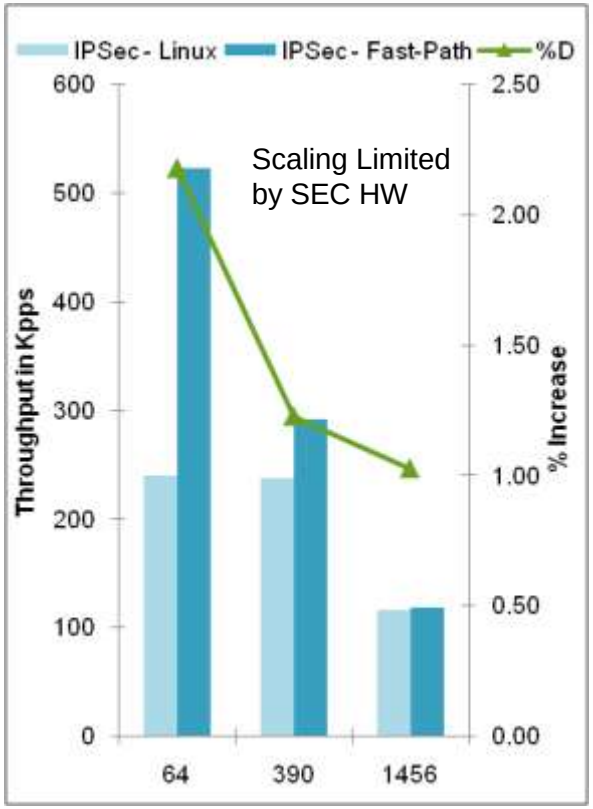
- Results on P2020 RDB - 1200/600/400 : 2-core SMP Linux



IPv4



NAPT

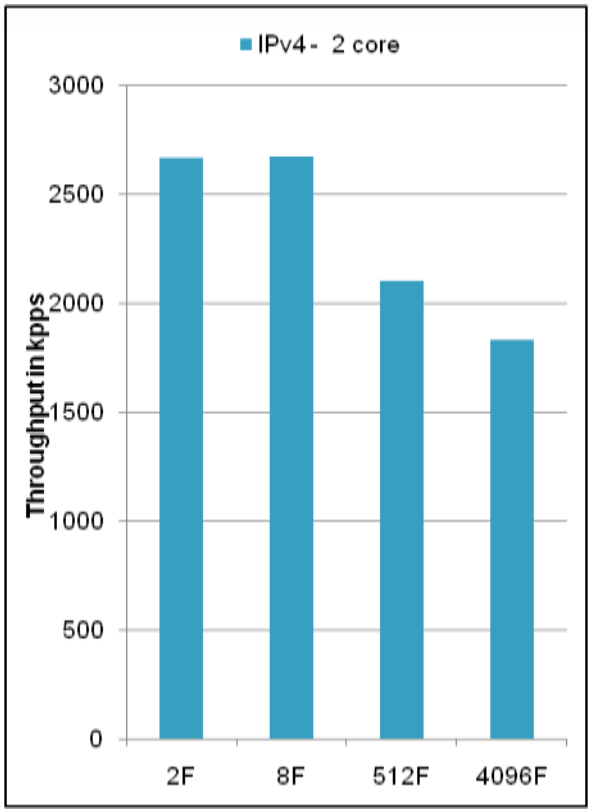


IPSec

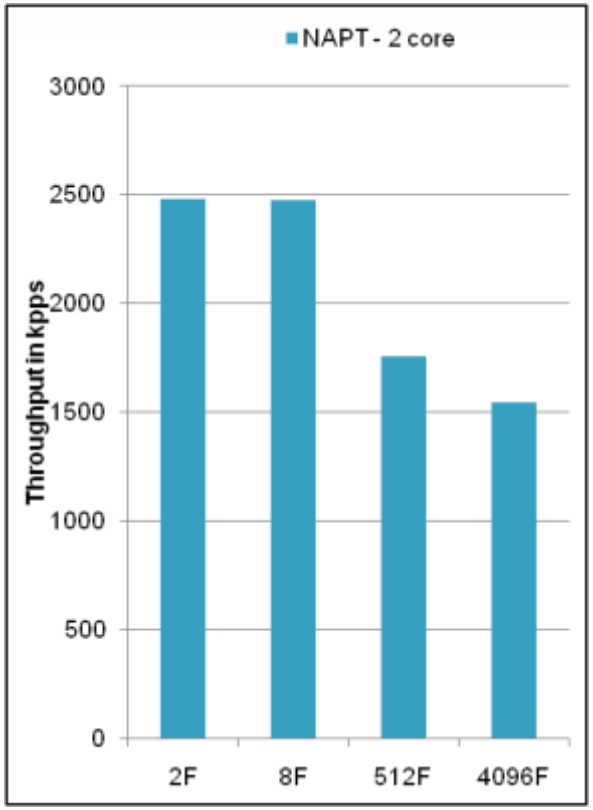
Significant (2x to 8x) performance improvement over native Linux

ASF Flow Scaling

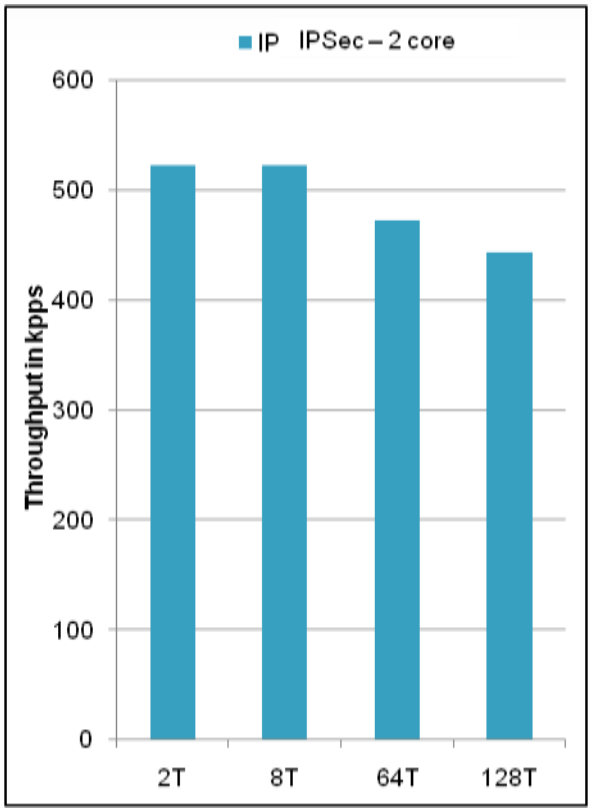
- Results on P2020 RDB - 1200/600/400 : 2-core SMP Linux – 64 byte traffic



IPv4



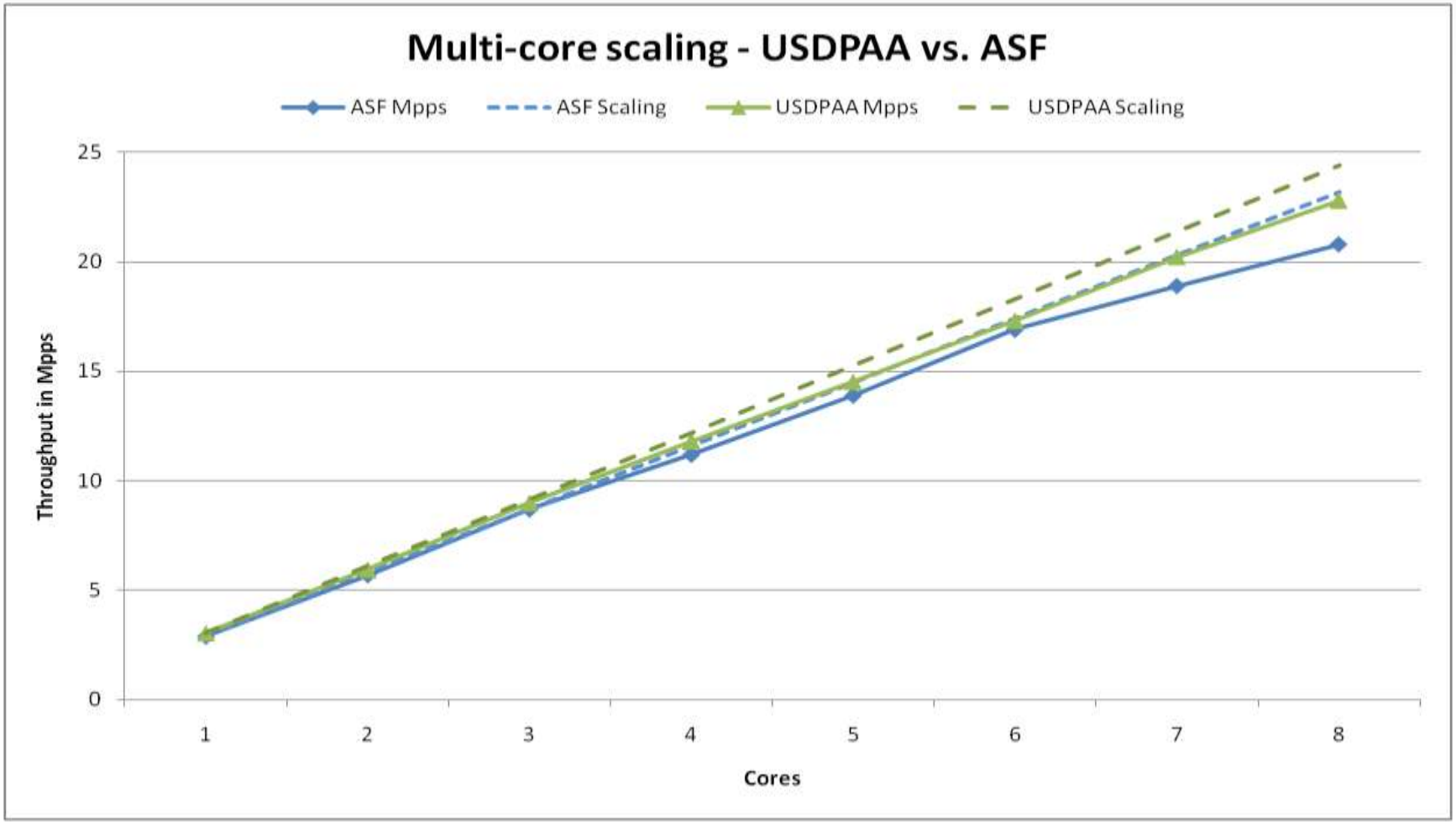
NAPT



IPSec

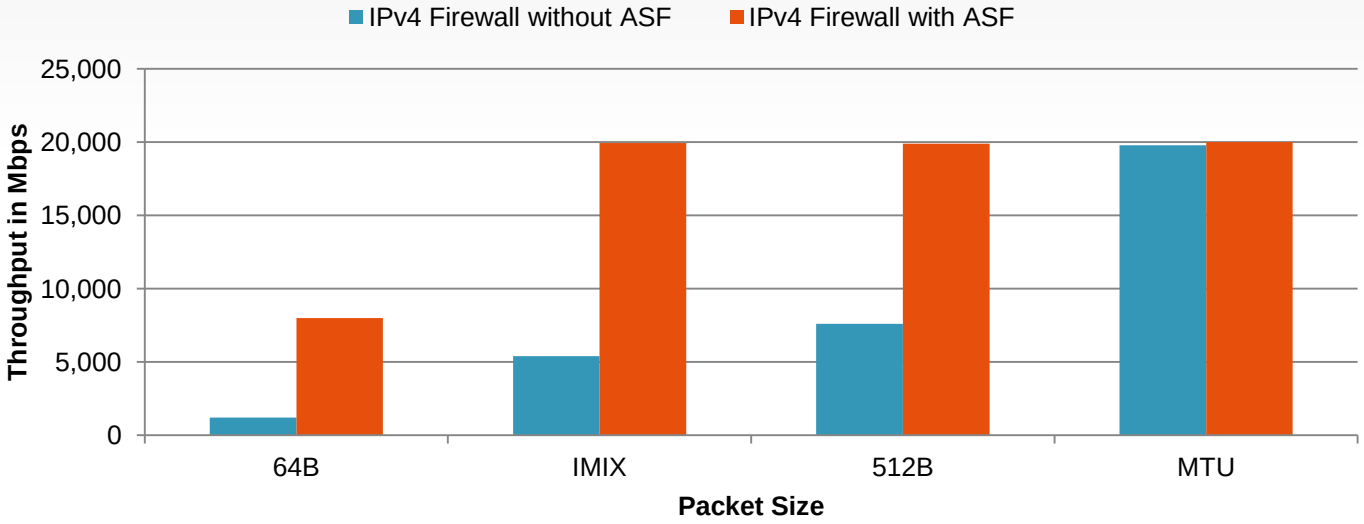
Low performance degradation for handling multiple flows

ASF Multicore Scaling

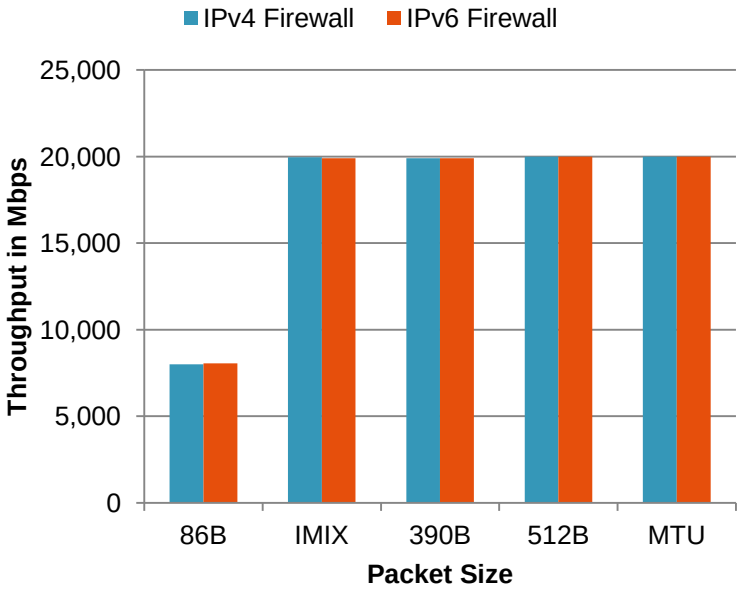


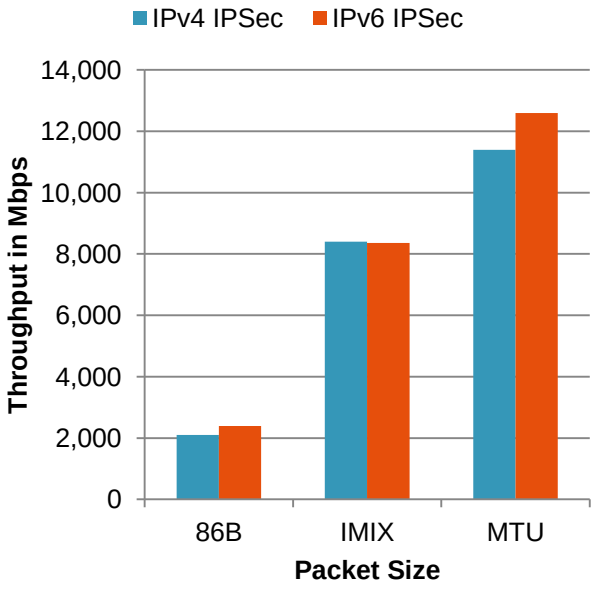
Almost linear scaling across multiple cores

ASF with VortiQa



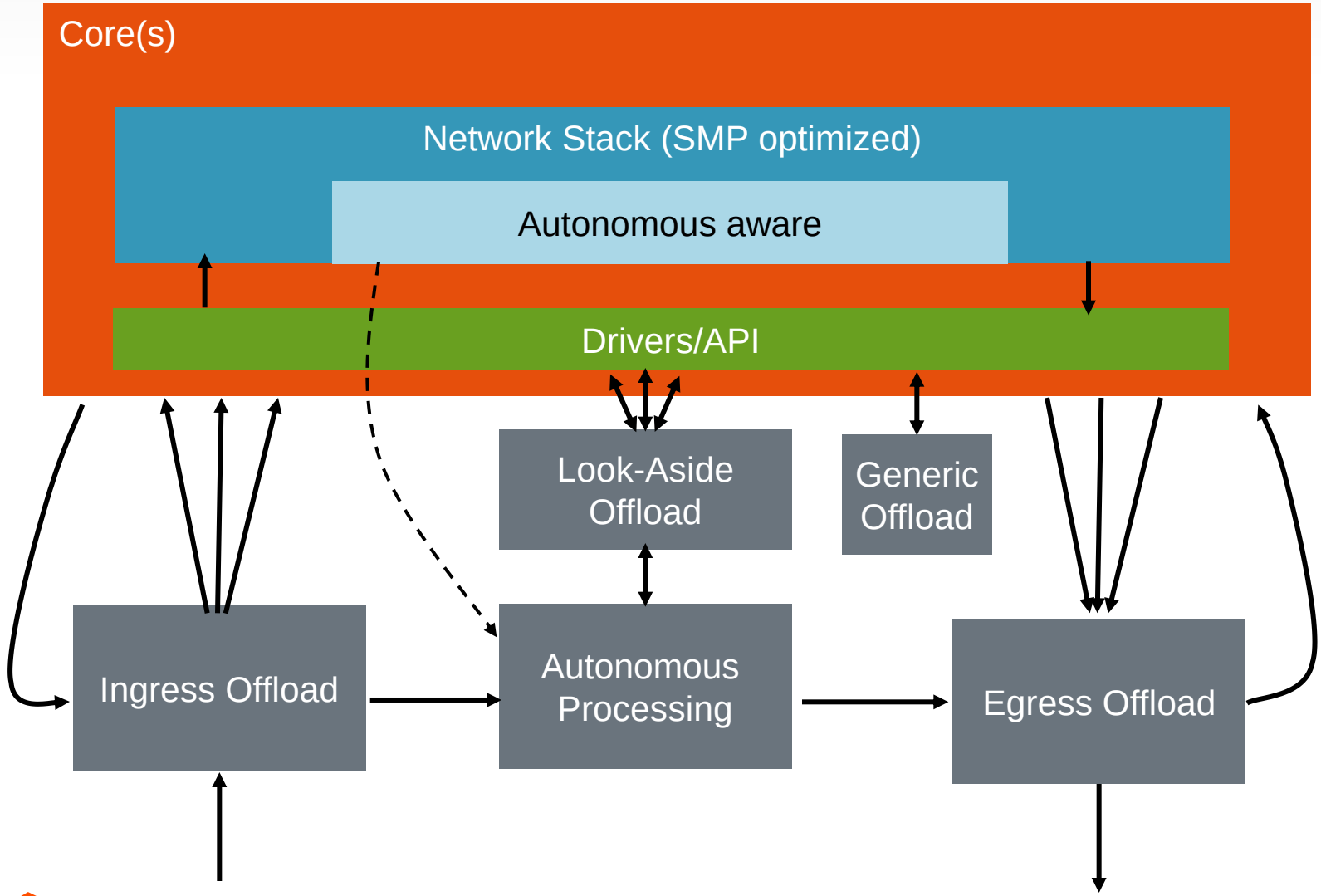
ASF boosts VortiQa performance by up to 7x





High Performance IPv4 & IPv6

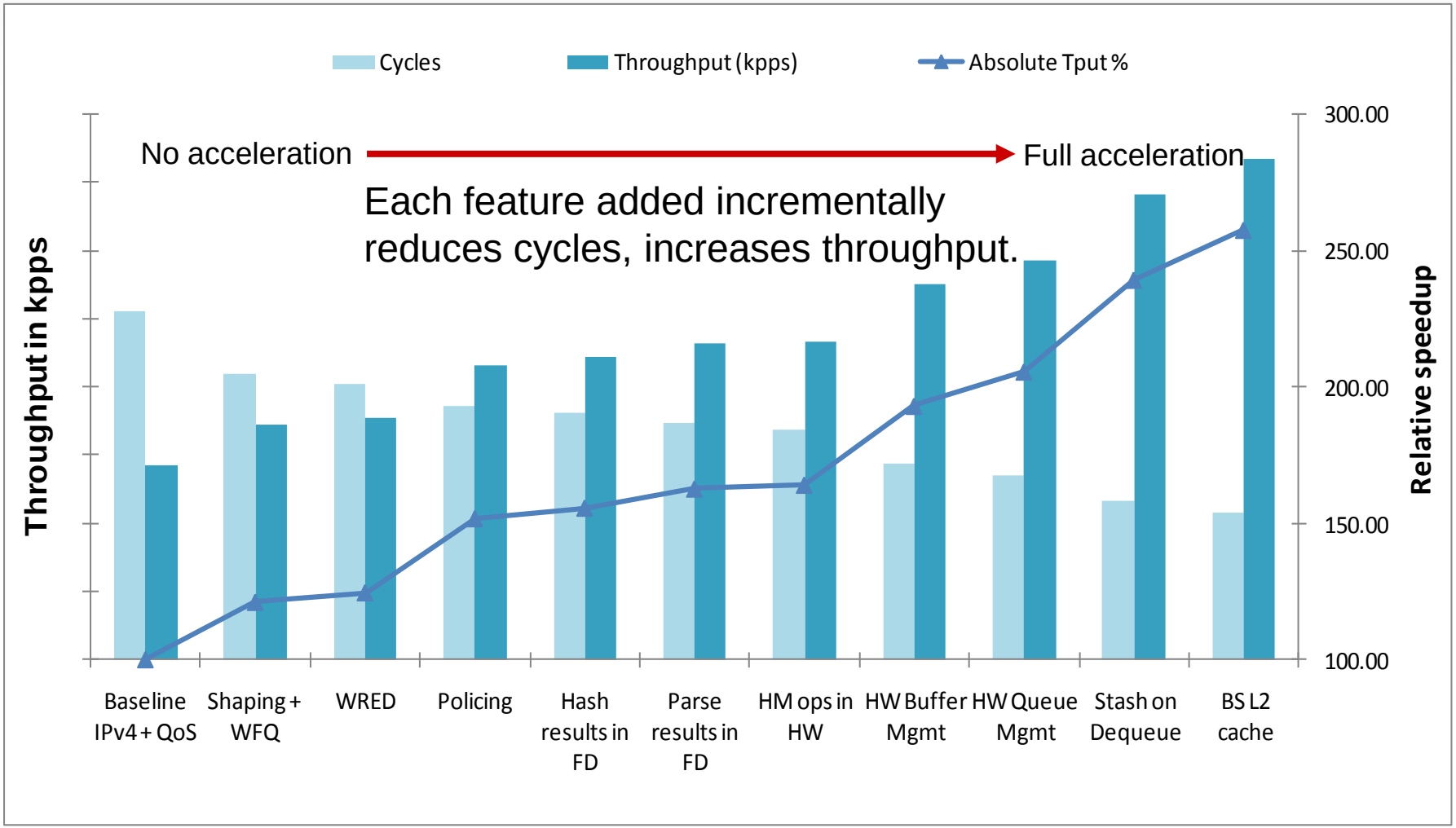
Data Path Hardware Acceleration



Leveraging Hardware Acceleration in ASF

Offload	Feature	Advantage
Ingress	Hash calculation	Packet distribution to multiple cores, flow-pinning, table lookup
	Coarse classification	Offload stateless ACL processing
	Packet parsing	Avoid software overhead
Generic	Hardware buffer management	No buffer alloc/free operations in software
	Hardware queue management	Simpler packet Rx/Tx, efficient stashing (to L1/L2), leaves room in cache for other data
Egress	Hardware QoS	Avoid software overhead, mitigate DoS attacks, prioritize CPU cycles
Core	Backside L2 cache	Faster access for multiple flow tables
Look-Aside	Protocol-aware cryptography	Offload protocol encapsulation/decapsulation, sequence tracking etc.

Hardware Acceleration Advantage



Hardware Acceleration provides up to 2x- 3x improvement

Summary and Q&A

- **ASF – Application Specific Fast-Path**
  - Optimized packet processing path
  - Consistent interface across platforms
  - Easy integration with network stacks
  - Single solution across QorIQ & Qonverge platforms

- **Performance Advantage**
  - Flexibility to leverage hardware acceleration
  - Optimized for Multicore scaling

Facebook.com/Freescale
Tag yourself in photos
and upload your own!

Tweeting?
Please use hashtag
#FTF2012

Session materials will be posted @ www.freescale.com/FTF
Look for announcements in the FTF Group on LinkedIn or follow Freescale on Twitter


freescale™