A close-up photograph of a green printed circuit board (PCB) with various electronic components, including several large square integrated circuits. A black microscope is positioned over the board, with its lens focused on the components. The background is dark and out of focus.

# Hardware Defined Networking

Picking the right data plane hardware  
for SDN and nfv

Gergely Pongrácz  
Senior Specialist, Ericsson Research

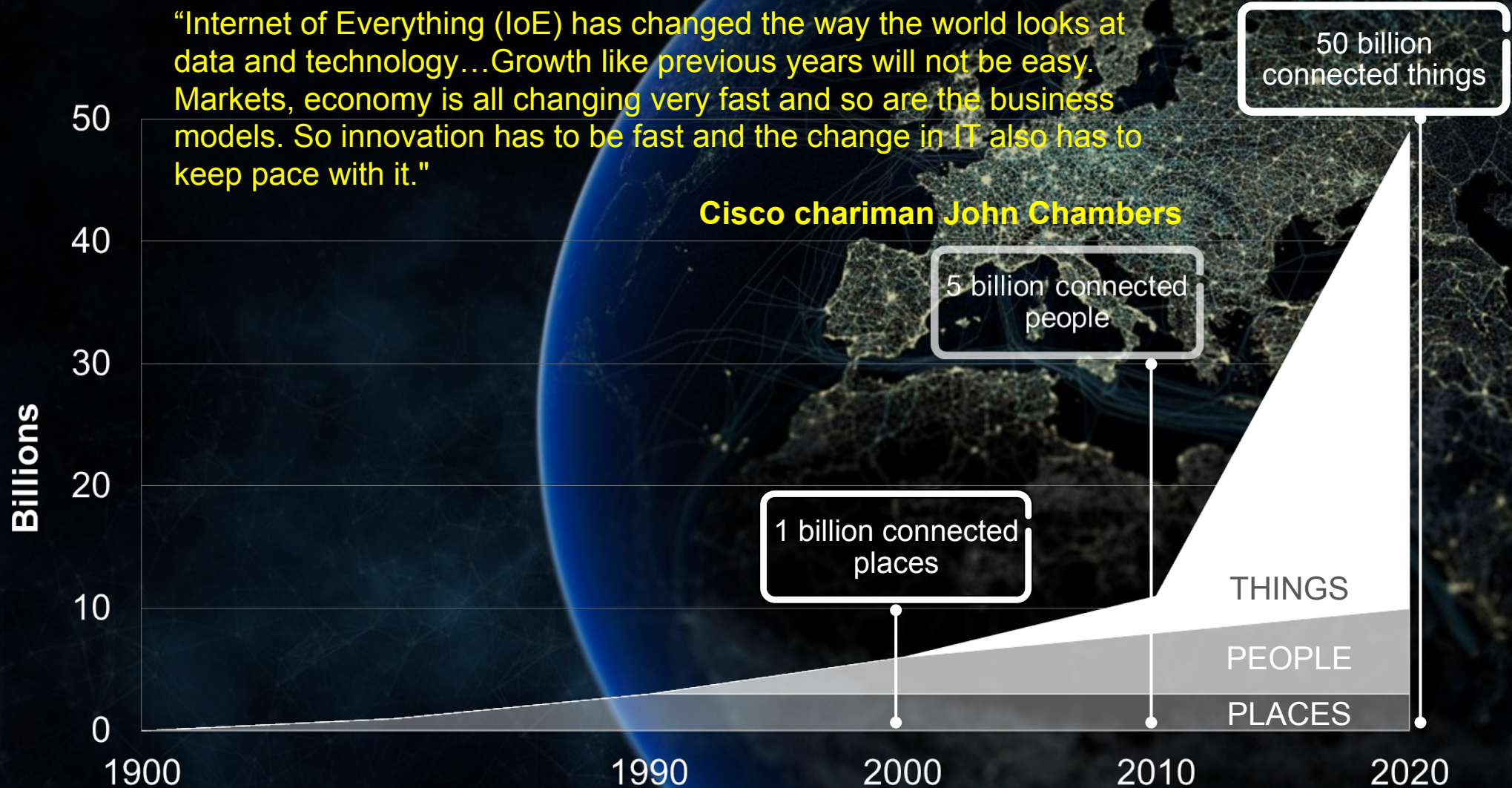


# PACE OF CHANGE



"Internet of Everything (IoE) has changed the way the world looks at data and technology...Growth like previous years will not be easy. Markets, economy is all changing very fast and so are the business models. So innovation has to be fast and the change in IT also has to keep pace with it."

**Cisco chairman John Chambers**



# Industry Trends



Spectrum is the most  
valuable resource

Simplicity+Automation  
for complex networks



Throughput to  
customer experience

Nodes & Domains  
to E2E systems



Fully enabled  
mobile enterprise

Telecom, Datacom  
& Mediacom



# Performance vs. Flexibility



Tech  
Evo

**These are the main drivers for SDN and NFV**

meaning

- more flexibility
- easier and centralized control
- generic, programmable hardware

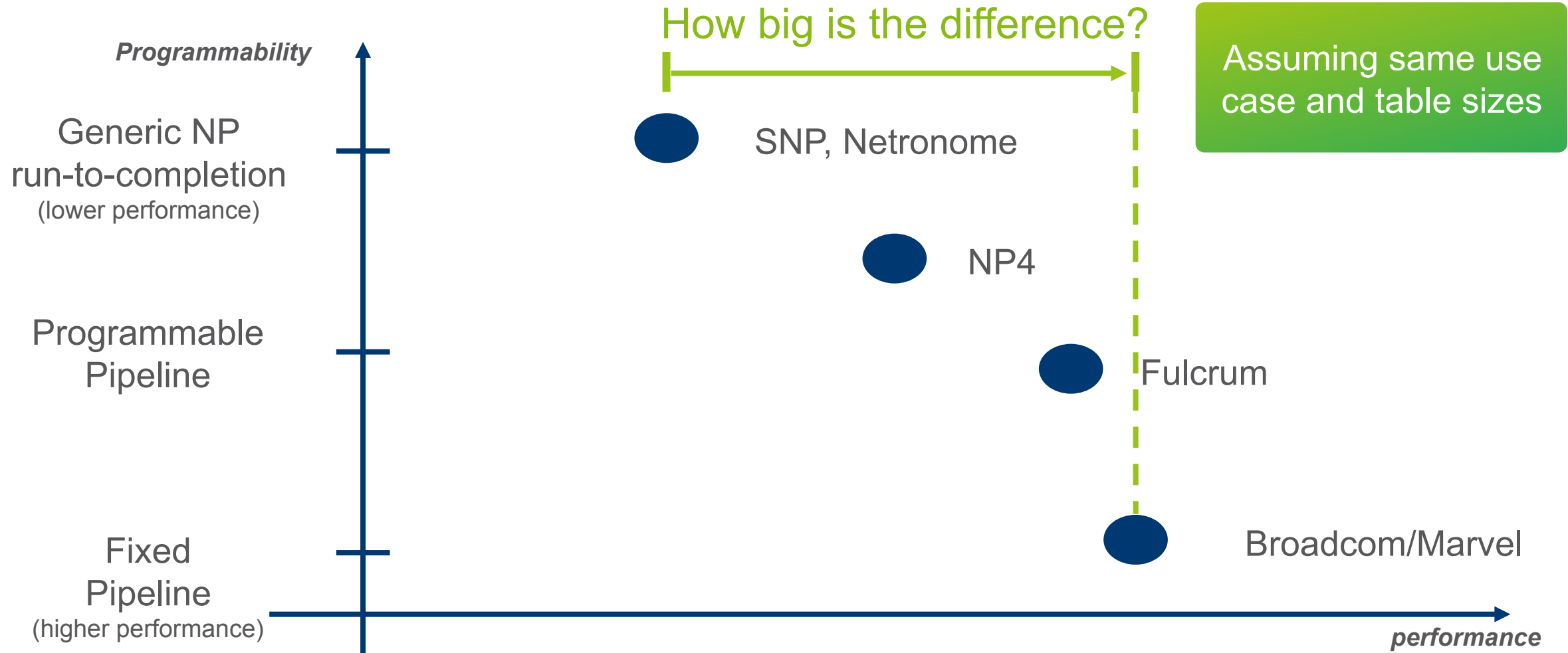
but we don't want to entirely **sacrifice performance**  
in this process!

today

Time

# Data Plane CHIP landscape

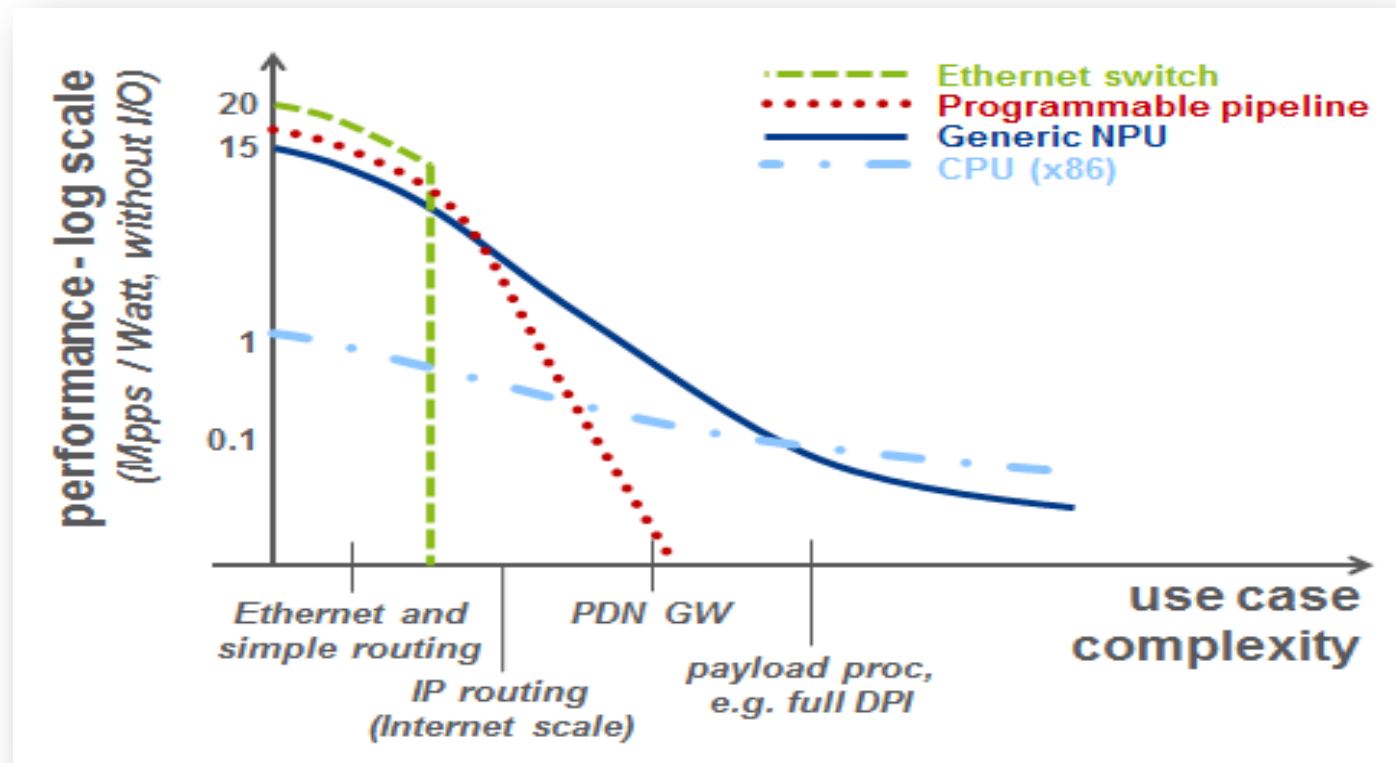
the usual way of thinking



# the Price of programmability



- › Programmability has some cost/overhead vs. performance (Mpps/Watt)
- › Statement: these costs are not huge, i.e. NPUs are close to purpose built hardware, and CPUs are also getting closer and closer to NPUs
- › Results are from measurements, modelling and calculations



# first comparison

using the product data sheet



Chip name (nm)	Gbps	Mpps	10G	Type
----------------	------	------	-----	------

So it seems

5-10x performance

difference between

But do we compare  
apples to apples?  
"con" and  
mable devices



# provider backbone bridging

## The common use case



- › Provider Backbone Bridging (PBB)  
aka. “MAC in MAC” is the most demanding Ethernet forwarding method
  - encapsulation / decapsulation, tagging, forwarding
  - good to be a common ground for comparison
- › PBB processing resource requirements:
  - 104 clock cycles
  - 25 L2 operations (depends on packet size)
  - 1 external RAM operation
- › Calculated performance
  - Based on the packet size >1 Tbps
    - › Approximately 1 Tbps with 64B packets
- › Modelling basics:
  - use case description → Assembly code → CPU and memory demands



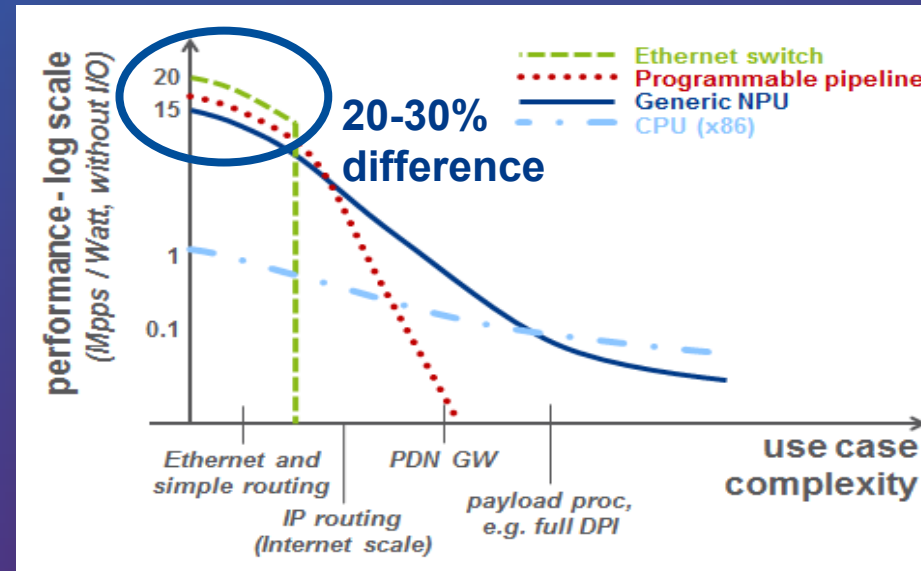
# Proper Comparison

## PBB use case results

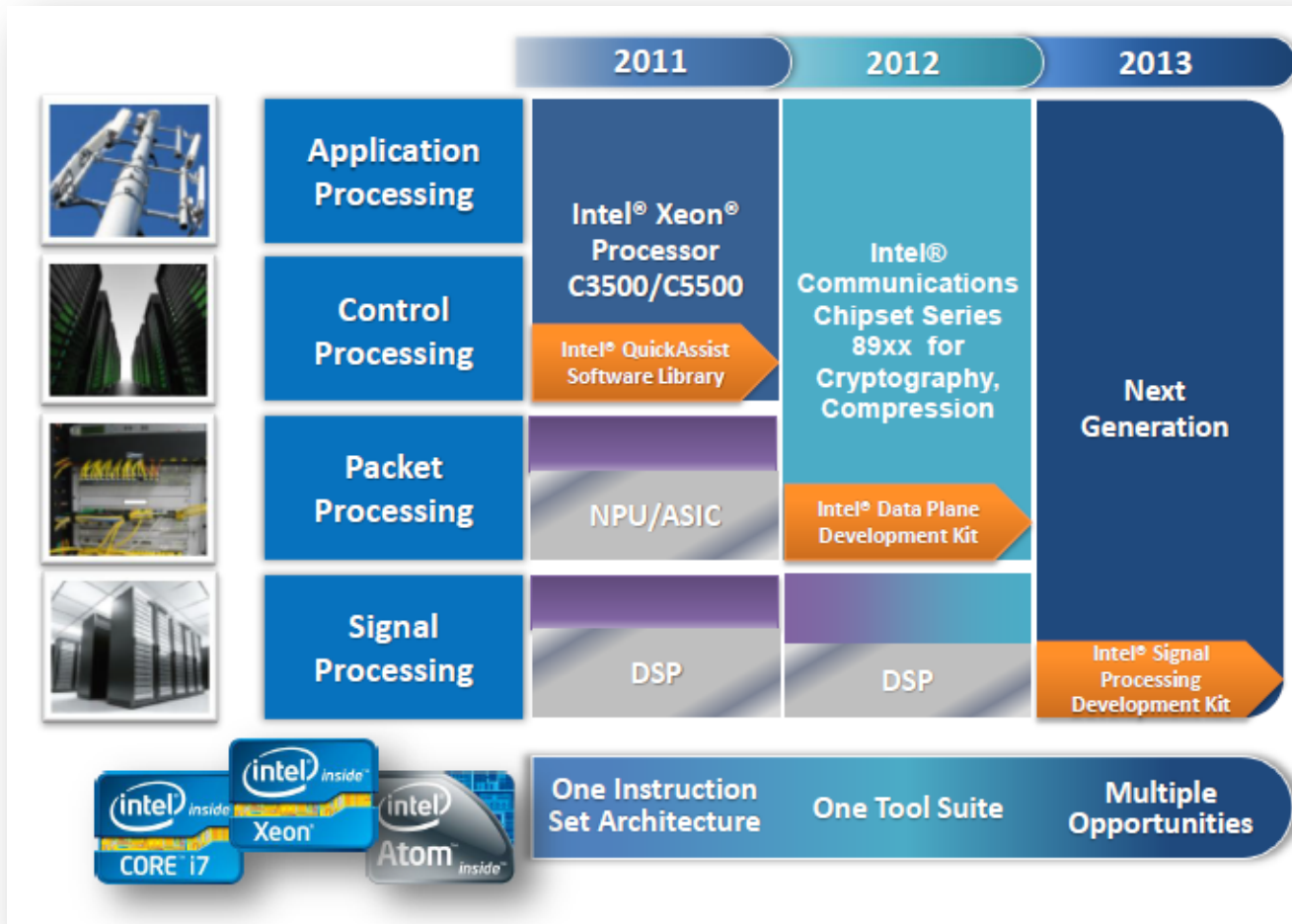


- › Results are theoretical: I/O was not considered
  - programmable chips today are designed for more complex tasks with less I/O ports

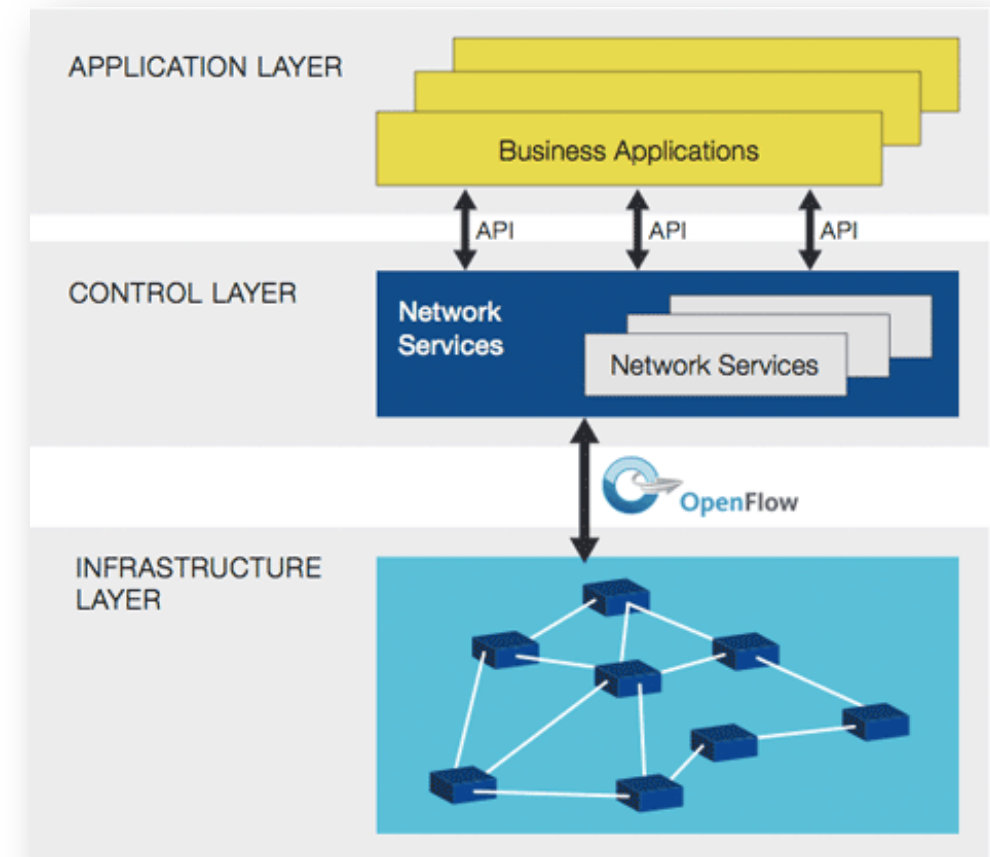
**Difference is 13-16 vs. 10-13 Mpps / Watt,  
i.e. around 1.25x instead of 10x**



# cpu and openflow for routing?



Source: [Packet Processing on Intel® Architecture](#)



Source: [Open Network Foundation](#)

# new modules for sdn router

## BGP in ODL + High performance flow switch

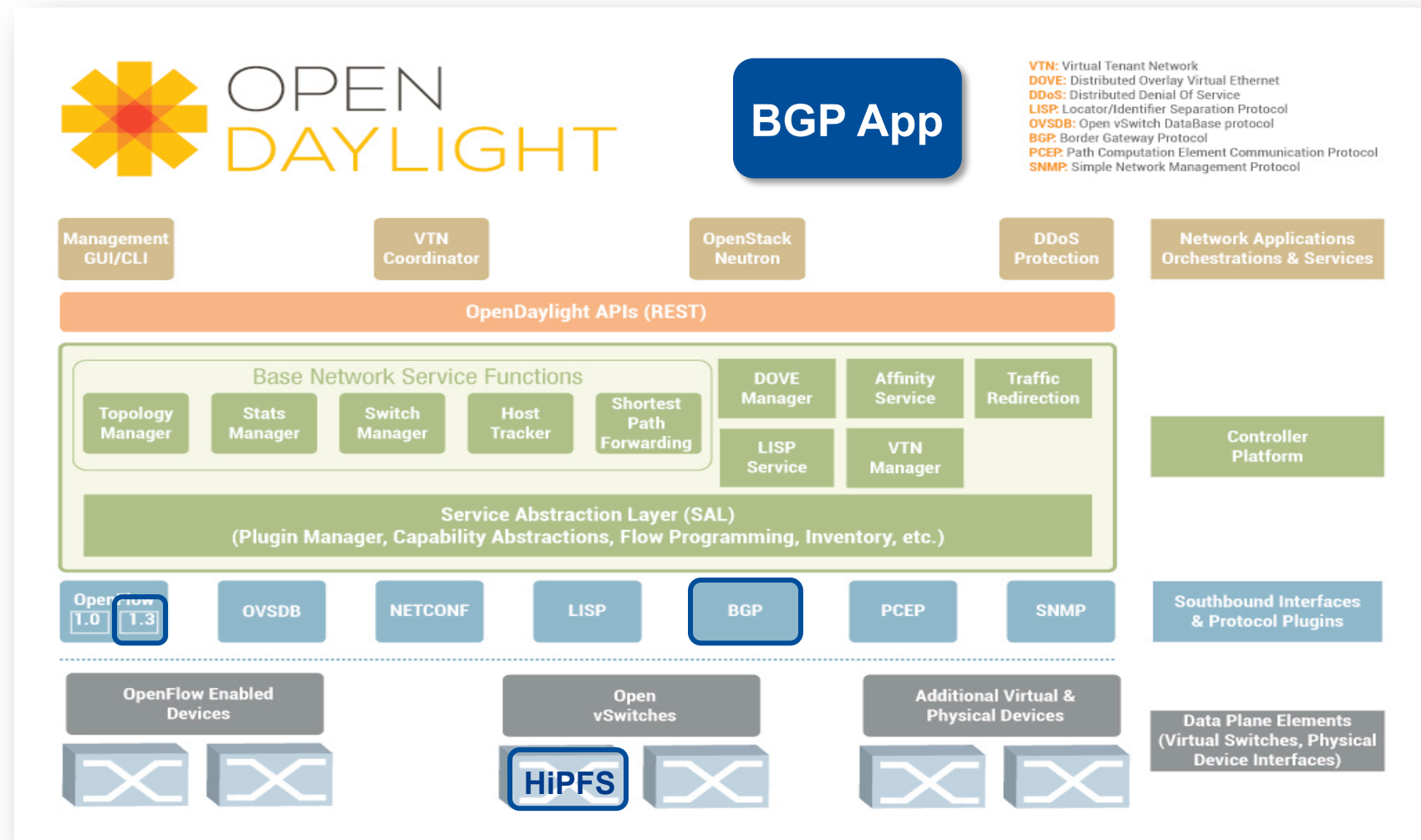


### › High Performance Flow Switch (HiPFS)

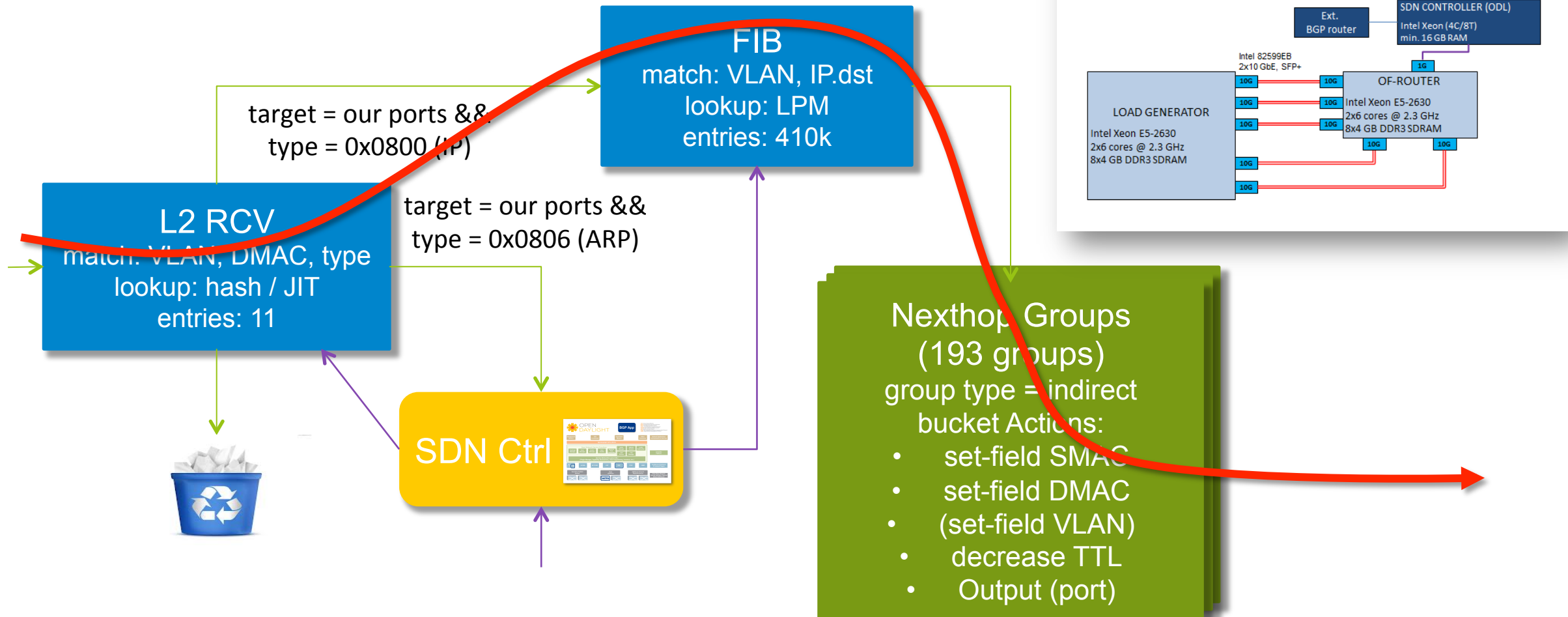
- OF 1.3.1 based
- optimizations:
  - › DPDK
  - › Longest Prefix Match
  - › Just In Time linking of most used OF flow actions

### › BGP support in ODL

- Using existing BGP module to receive external BGP messages
- New BGP App
  - › receives prefix list from BGP
  - › receives topology from OpenFlow
  - › calculates FIB
  - › downloads FIB via OF 1.3
- Using existing OF 1.3 module to communicate with switches



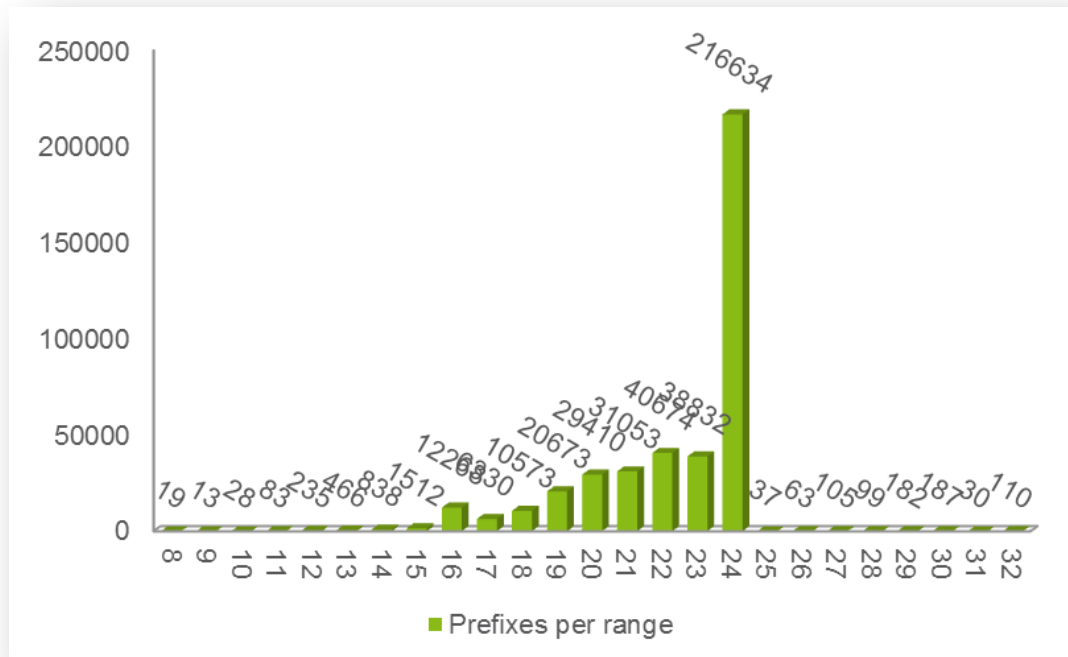
# High performance flow switch internal view



# High performance flow switch measurement results



- › Forwarding table (FIB) from a deployed access router
  - 410k prefixes, /24 prefixes dominate
  - 194 nexthops



- › Test cases
  1. 10 measurement flows with variable packet length
  2. Real-life packet trace
  3. Cache unfriendly, worst-case packet trace

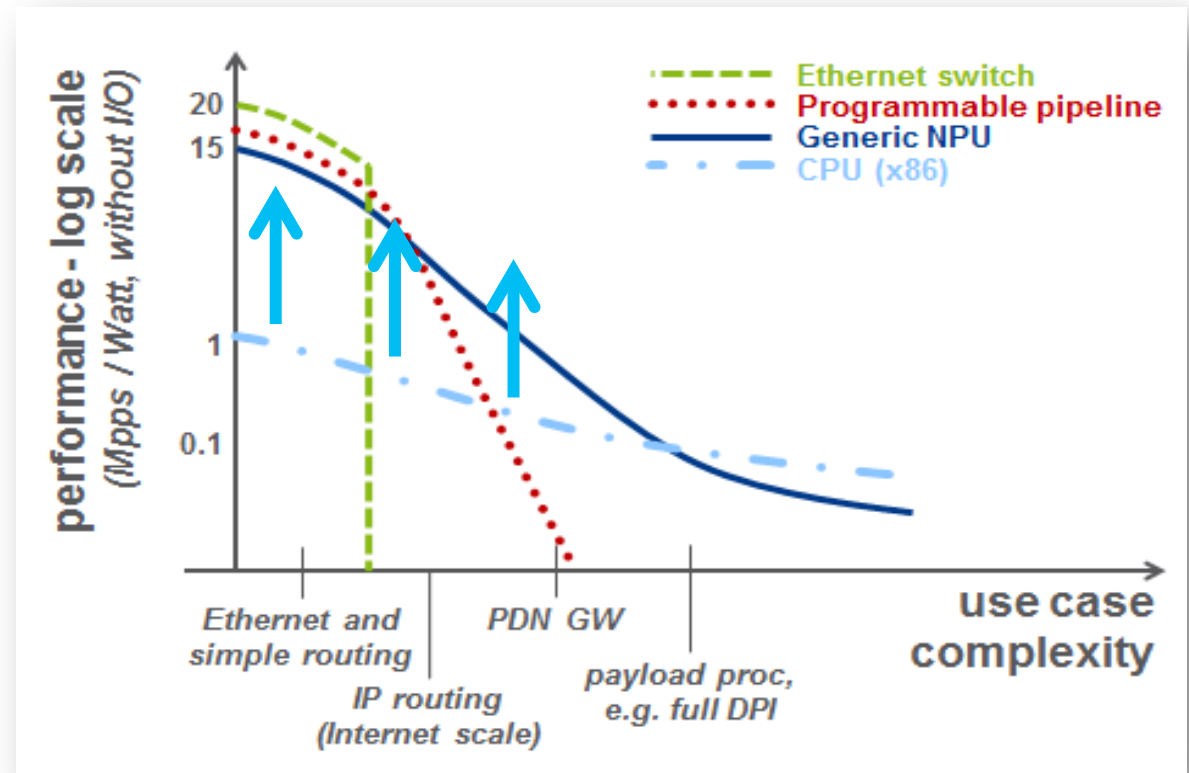
Packet size	Mpps	Gbps
64	110	75
128	82	100
256	44	100
512	23	100
Real Trace	29	100
Worst Case Trace	77	55



# cpu and openflow for routing!



- › With Intel x86 we could reach ~100 Mpps and 100 Gbps
  - Use cases: DC Gateway (all cores), hypervisor virtual switch (1-2 cores) ← easily scalable
  - Still around 12W / 10G port, but decreasing
  - Proved that the CPU (x86) curve is quite flat
    - › routing and switching does not make a big difference
- › CPUs are getting closer to NPUs
  - Big drive for power efficiency
  - More cores → better pps/W ratio
  - Characteristics are getting more and more similar for the two blue curves
- › Intel x86 seems to be suitable solution for the switch instance in the **UNIFY Universal Node**

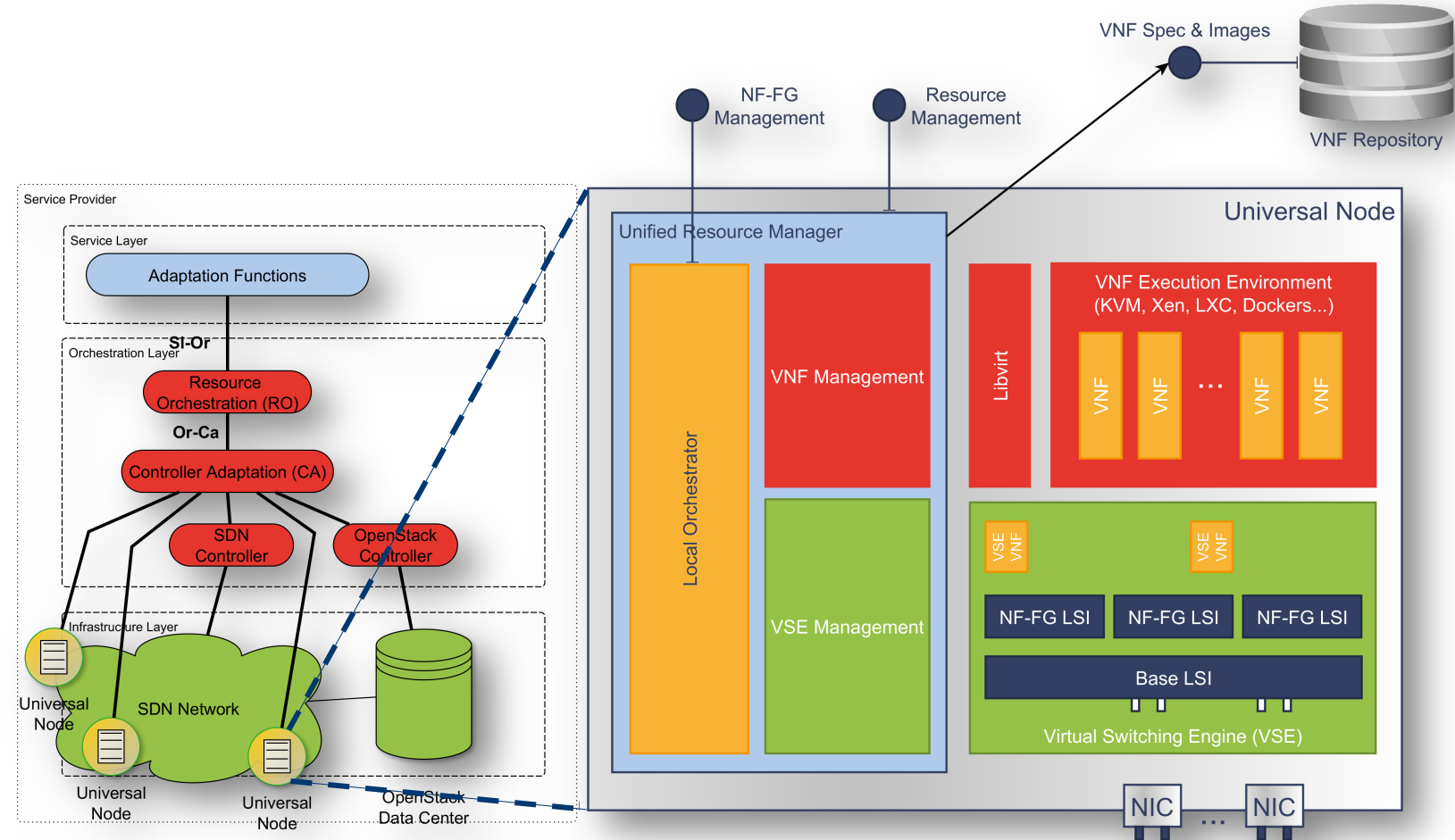


# UNIFY Universal Node

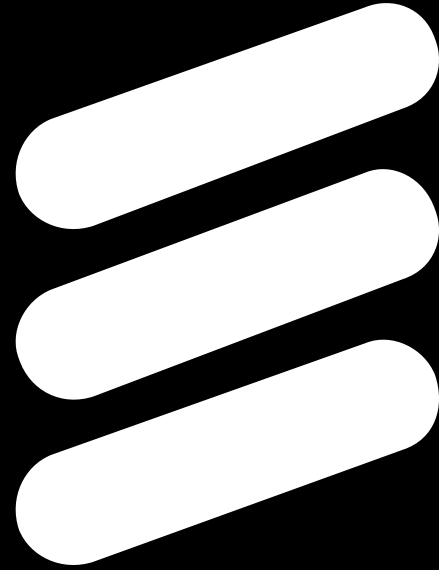
bridging the gap between compute and networking



- › The Universal Node can host VNFs as full VMs, lightweight isolated containers or enhanced logical switch instances
- › VNFs of the incoming NF-FG are logically mapped to the internal network and to traffic steering between the VNFs
- › Different optimization techniques (e.g. DPDK on x86) are used to achieve high performance in the UN Virtual Switching Engine as well as optionally in the various VNFs.



Physical networking, virtual networking (vSwitch) and VNF (compute/storage) are in the same node



**ERICSSON**