

# P51: High Performance Networking

## Lecture 6: Programmable network devices

# High Throughput Interfaces

# Performance Limitations

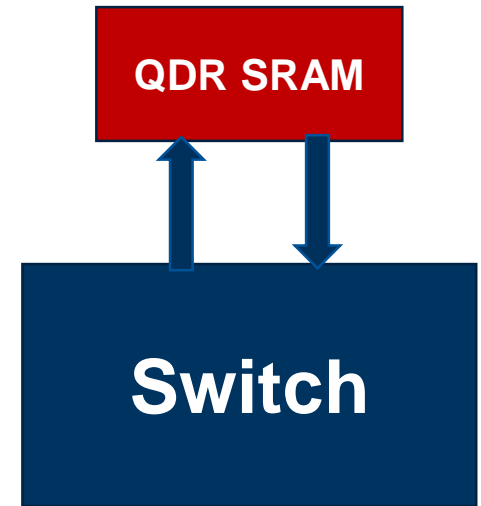
- So far we discussed performance limitations due to:
  - Data path
  - Network Interfaces
- Other common critical paths include:
  - Memory interfaces
    - Lookup tables, packet buffers
  - Host interfaces
    - PCIe, DMA engine

# Memory Interfaces

- On chip memories
  - Advantage: fast access time
  - Disadvantage: limited size (10's of MB)
- Off chip memory:
  - Advantage: large size (up to many GB)
  - Disadvantage: access time, cost, area, power
- New technologies
  - Offer mid-way solutions

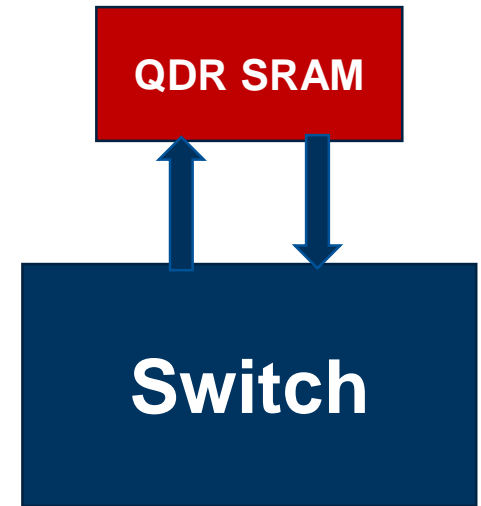
# Example: QDR-IV SRAM

- Does 4 operations every clock: 2 READs, 2 WRITEs
- Constant latency
- Maximum random transaction rate: 2132 MT/s
- Maximum bandwidth: 153.3Gbps
- Maximum density: 144Mb
- Example applications: Statistics, head-tail cache, descriptors lists



# Example: QDR-IV SRAM

- Does 4 operations every clock: 2 READs, 2 WRITEs
  - *DDR4 DRAM: 2 operations every clock*
- Constant latency
  - *DDR4 DRAM: variable latency*
- Maximum random transaction rate: 2132 MT/s
  - *DDR4 DRAM: 20MT/s (worst case!  $t_{RC} \sim 50ns$ )*
    - DDR4 theoretical best case 3200MT/s
- Maximum bandwidth: 153.3Gbps
  - *DDR4 DRAM maximum bandwidth: 102.4Gbps (for 32b (2x16) bus)*
- Maximum density: 144Mb
  - **DDR4 maximum density: 16Gb**
- Example applications: Statistics, head-tail cache, descriptors lists
  - *No longer applicable: packet buffer*

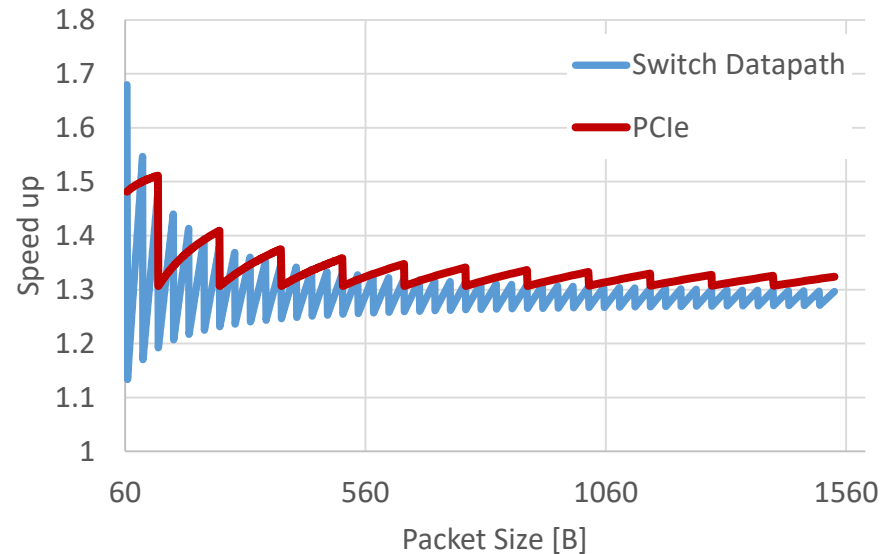


# Random Memory Access

- Random access is a “killer” when accessing DRAM based memories
  - Due to strong timing constraints
- Examples: rules access, packet buffer access
- DRAMs perform well (better) when there is strong locality or when accessing large chunks of data
  - E.g. large cache lines, files etc.
  - Large enough to hide timing constraints
    - E.g. for 3200MT/s, 64b bus: 50ns~ 1KB

# Example: PCI Express Gen 3, x8

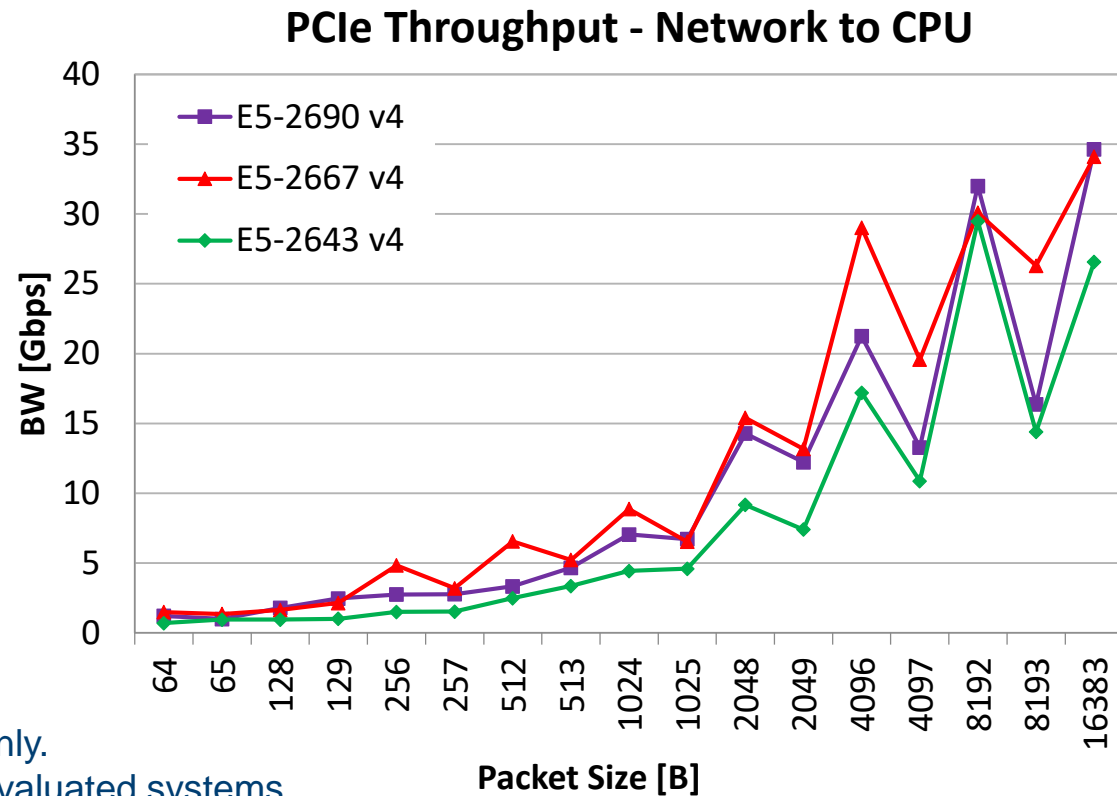
- The theoretical performance profile:
- PCIe Gen 3 – each lane runs at 8Gbps
- ~97% link utilization (128/130 coding, control overheads)
- Data overhead – 24B-28B (including headers and CRC)
- Configurable MTU (e.g., 128B, 256B, ...)





# Example: PCI Express Gen 3, x8

- Actual throughput on VC709, using Xilinx reference project: (same FPGA as NetFPGA SUME)
- This is so far from the performance profile...
- Why?



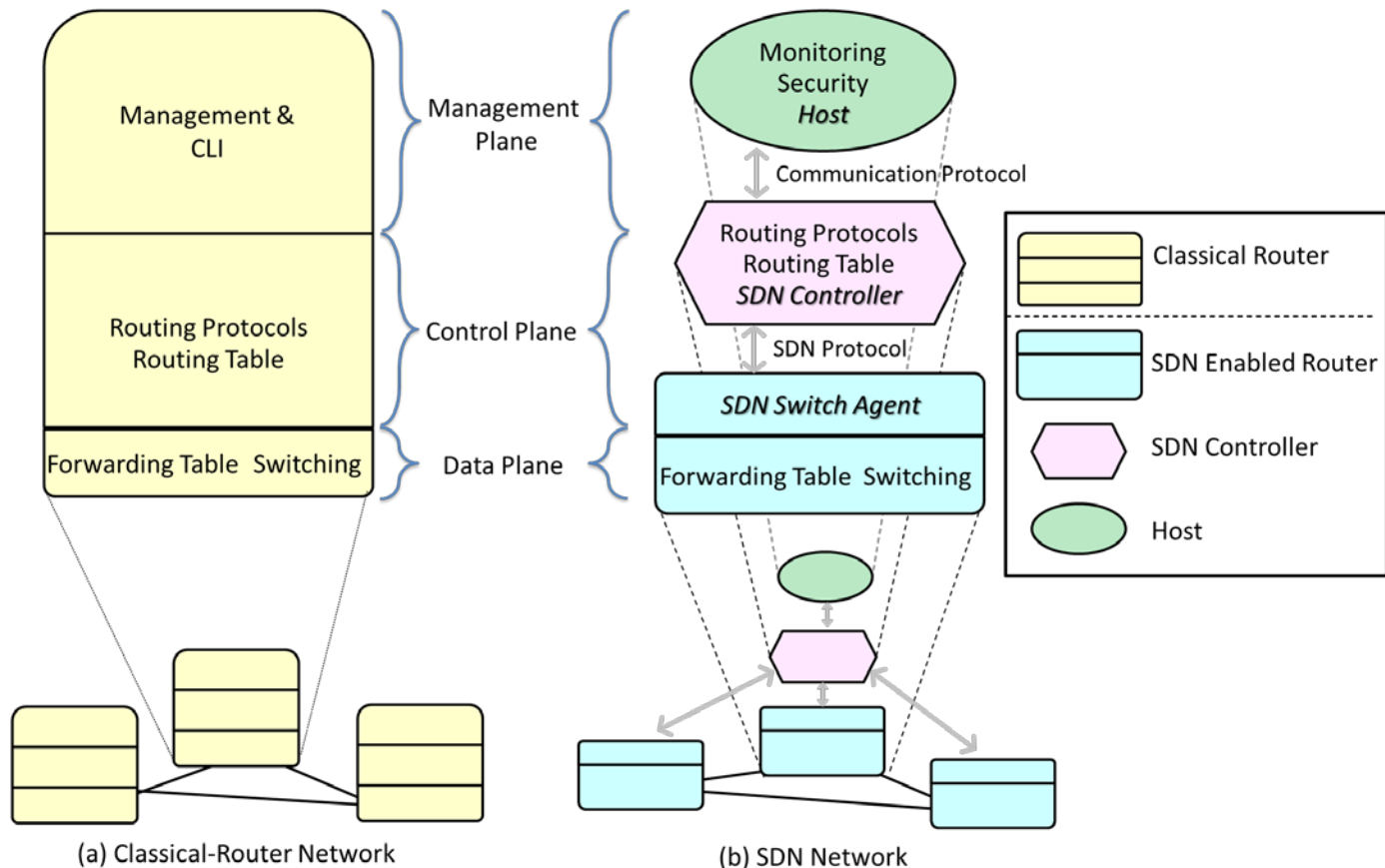
Note: the graph is for illustration purposes only.  
There were slight differences between the evaluated systems.

# Software Defined Networks

We will not discuss SDN...

# Software Defined Networking (SDN)

Key Idea: Separation of Data and Control Planes



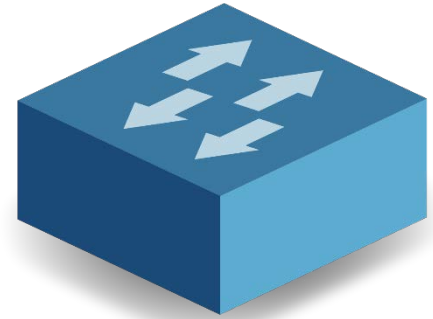
# Software Defined Networking (SDN)

- SDN is about control and manageability
- Attending to challenges in:
  - Controlling large scale networks
  - Different underlying hardware
  - Device complexity
  - ...
- The data plane is simple, the “smartness” is in the control plane
  - Focused on the packet processing

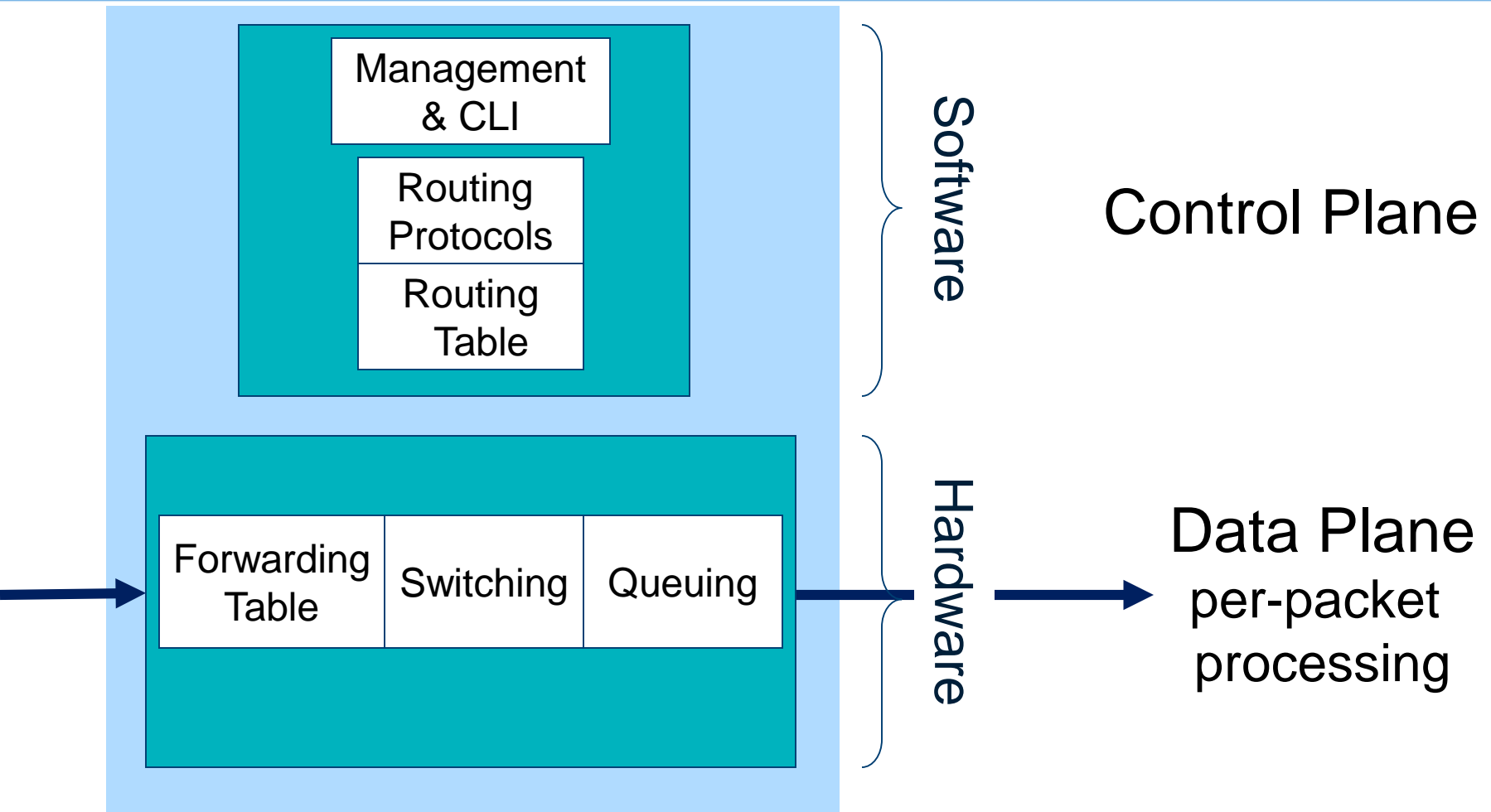
# Programmable Network Devices

# A bit of history...

- The role of a switch is to connect multiple LAN segments
- Operates on Layer 2
- Supports a single operation: Forwarding
- If you want to do more:
  - Layer 3 is handled by the software
  - Protocol processing is handled by another device (NPU / PPU)
- Valid until mid-2000's
  - E.g. 2002's "state of the art" Broadcom Strata XGS, 8x10GE

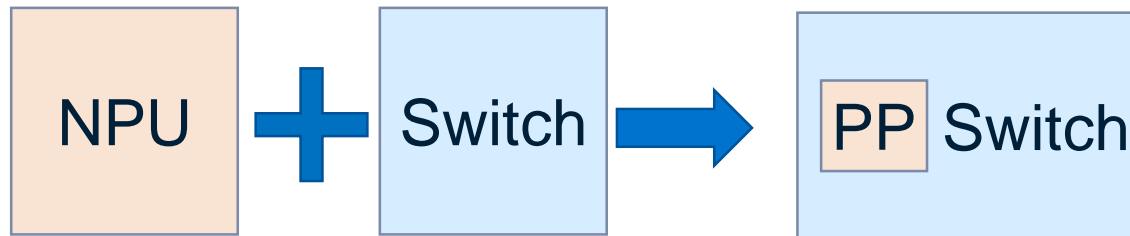


# Basic components of an IP router (originally)



# A bit of recent history...

- Mid-2000's to start-2010's:
  - Fixed function switches
  - Integration of functions: same trend as with CPUs
    - Why use NPU + Switch if you can use just a switch?
    - For limited applications





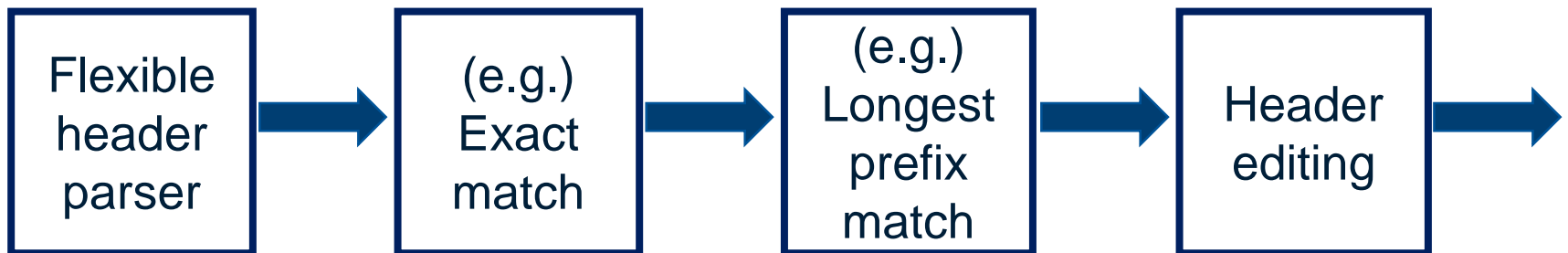
# A bit of recent history...

- Mid-2000's to start-2010's:
  - Fixed function switches
  - Supporting multiple (pre-defined) protocols
    - E.g. Layer 3 switching
  - Fixed pipeline (example only):



# A bit of recent history...

- Start-2010's – Recent years:
  - Partly / fully flexible switches
  - Support \*many\* protocols
  - Flexibility in selecting the protocols, memories used, header size,...



# Programmable network devices

- Partly / fully programmable
  - Mostly focused on the header processing
  - But starting to attend also to queueing / switching / TM / ...
- Support ANY protocol
- Pipeline is “programmable”
  - But within given resource limitations

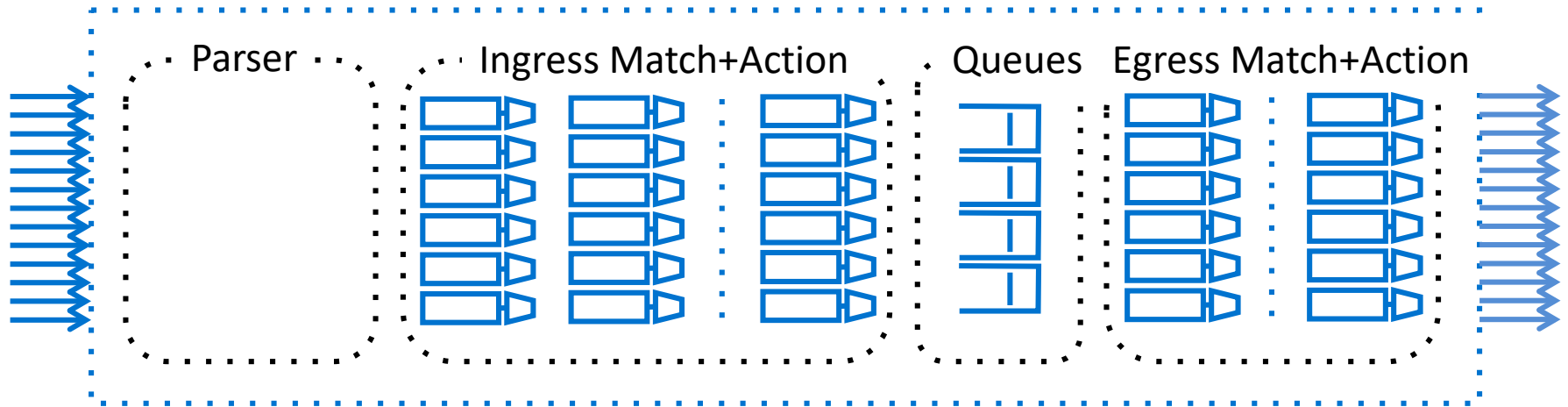
# Programmable network devices

## Advantages:

- New Features – Add new protocols
- Reduce device complexity – e.g., Implement only required protocols.
- Flexible use of resources
- SW style development – better innovation, fix data-plane bugs in the field

# Reconfigurable Match-Action Model

- RMT – Reconfigurable Match-Action Model



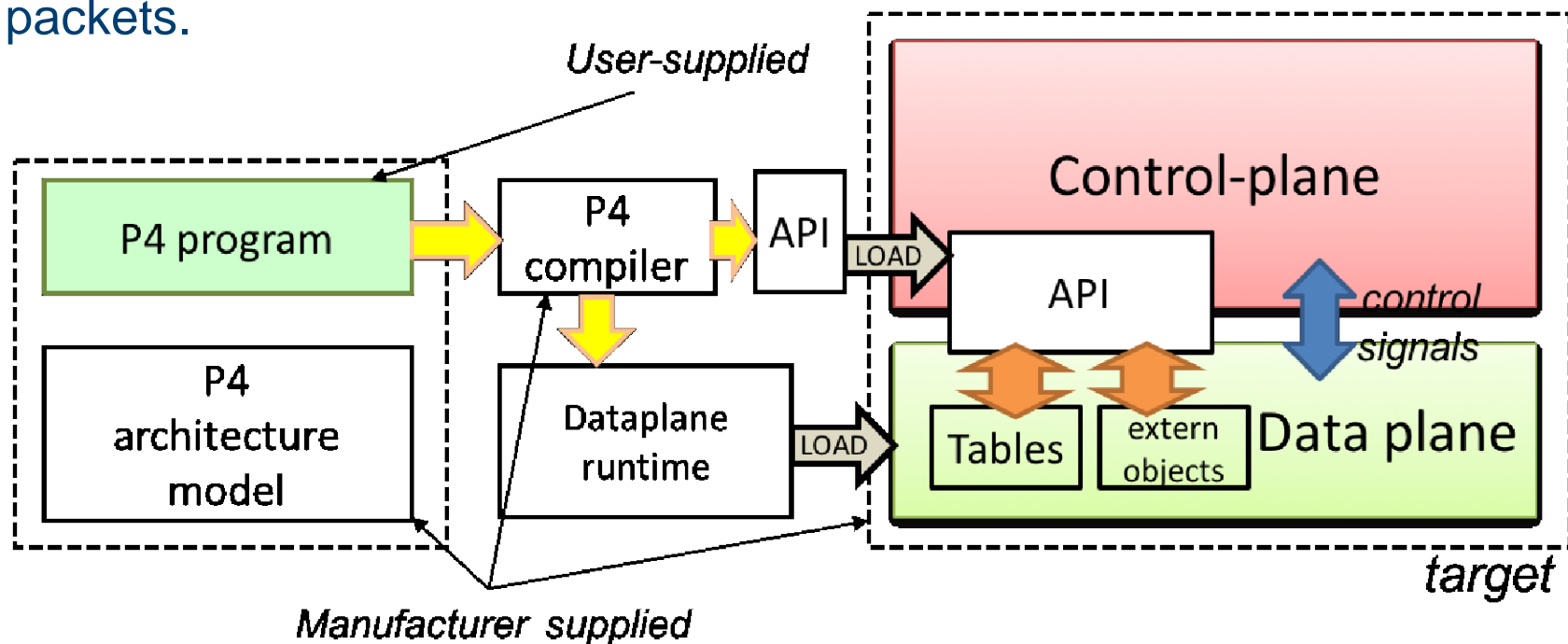
- Bosshart, Pat, et al. "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN." *SIGCOMM* 2013.

# Programmable network devices

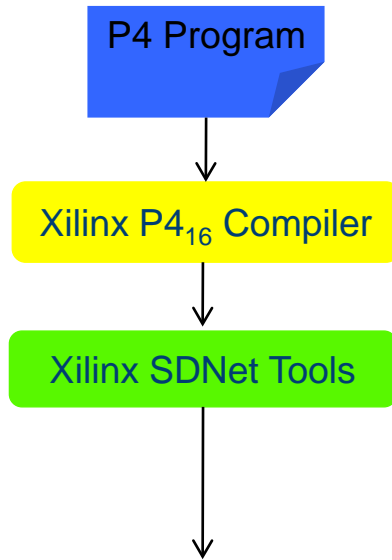
- How do you programme a network device?
- Requires:
  - Programming language
  - Compilers
  - Architecture
    - Underlying hardware support
- We will discuss one popular option, but there are more

# P4

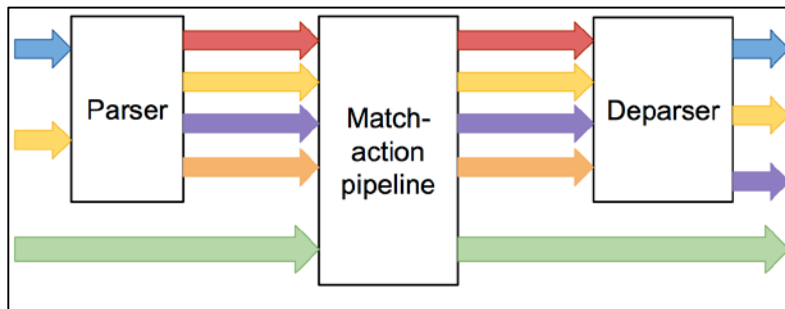
- [www.p4.org](http://www.p4.org)
- A declarative language
- Telling forwarding-plane devices (switches, NICs, ...) how to process packets.



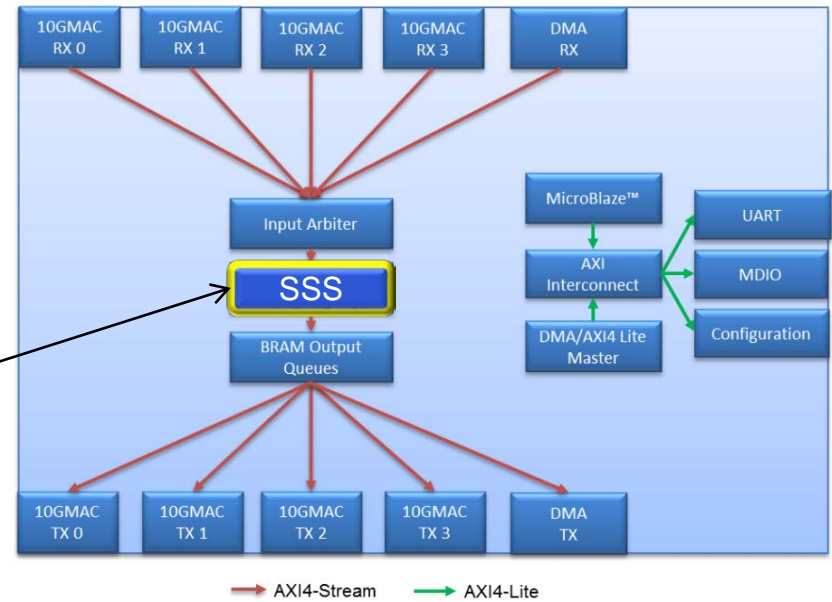
# Example: P4 on NetFPGA (P4-NetFPGA)



*SimpleSumeSwitch Architecture*

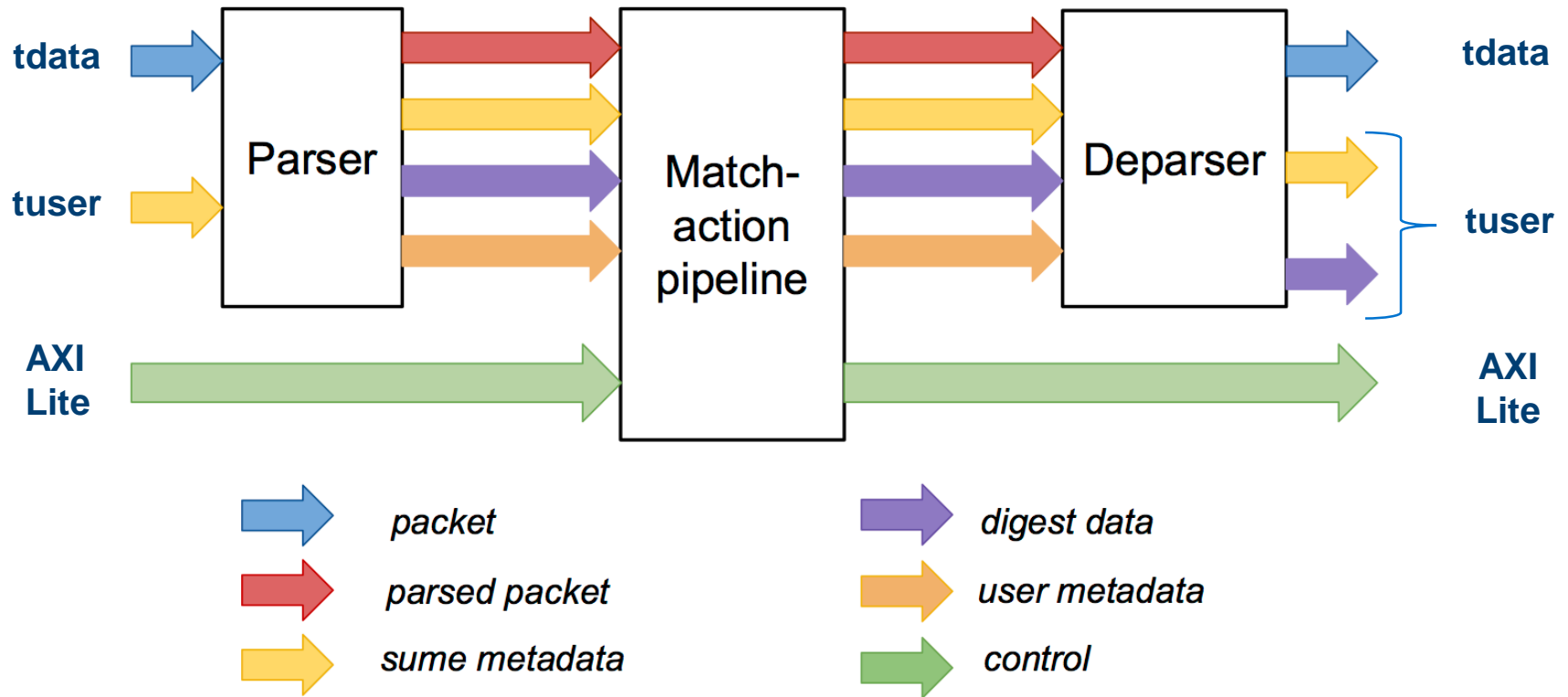


NetFPGA Reference Switch





# SimpleSumeSwitch Architecture Model for SUME Target

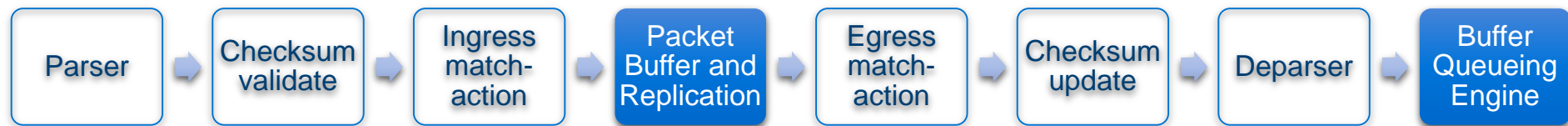


P4 used to describe parser, match-action pipeline, and deparser

# P4 PSA: Portable Switch Architecture

- Composability
  - Example: Multiple functions in a single pipeline
- Portability
  - Example: Apply a function consistently across a network
- Comparability
  - Example: Compare functions implementation, A vs. B

## Pipeline



## Externs



# P4 – Examples Use Cases

- Network telemetry (INT)
- New protocols (e.g., NDP)
- Layer 4 load balancing
- In Network Caching (NetCache) –  $\times 10$  throughput
- Consensus Protocols (NetPaxos) –  $\times 10,000$  throughput
- Tic-Tac-Toe


# In Network Computing

- Idea: move services and applications from the host to the network
- Somewhat similar terms:
  - Network as a Service (NaaS)
  - Hardware acceleration (but network specific)
- Implementations:
  - Smart NICs
  - Programmable Switches
- Different platforms support different languages

# In Network Computing - Examples

- Machine learning
- Graph processing
- Key-value store
- Security (e.g., DDoS detection)
- Big data analytics
- Stream processing
  
- But nothing is for free (cost, power, space, ...)

# P51 Summary

- Architecture of high-performance network devices
  - High throughput devices
  - Low latency devices
  - Programmable network devices
- 
- This is just a little glimpse into high performance networking...
  - Reminder: 6/3, same place and time – Special talk:  
Steve Pope (CTO, Solarflare) – High performance end-host networking