



Evolution from OpenFlow to P4/P4Runtime

COIN Meeting IRTF 105

Jesper Eriksson, NoviFlow

July 26, 2019

www.noviflow.com

ABOUT NOVIFLOW

FOUNDED
2012

World's leading provider of SDN Network Operating Systems and solutions for programmable match-action data planes

Business Model
Software Licensing
Systems Sales

FOCUS
SDN
Cybersecurity
DCi

PRODUCTS
NOS
Whitebox switches
Controller Applications

ARCHITECTURES
NPU
Tofino

Production deployments worldwide by global network operators, Hyperscalers, large enterprises and government agencies



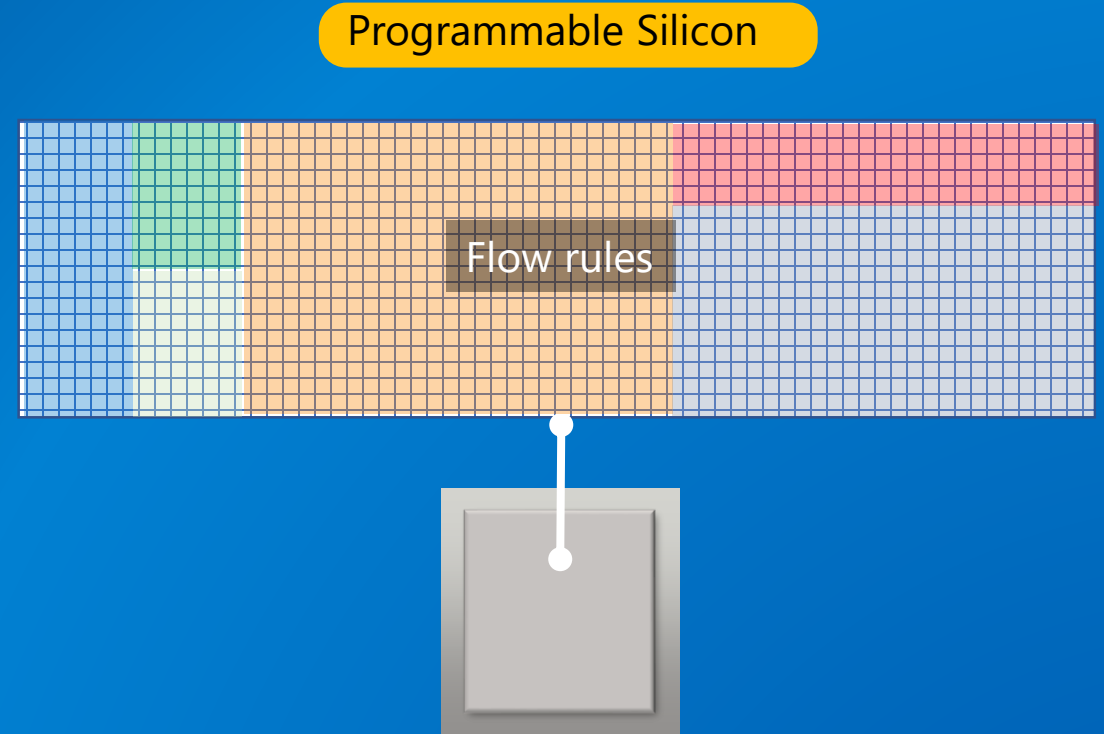
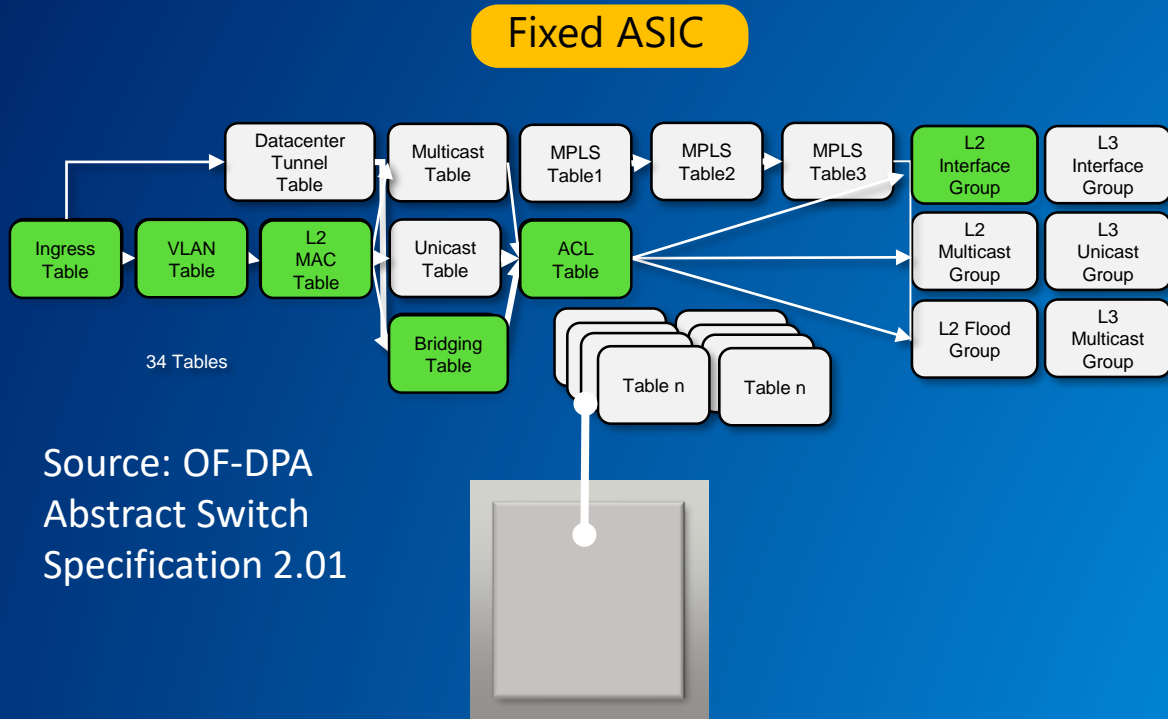
2017



2018



Match-Action Pipelines



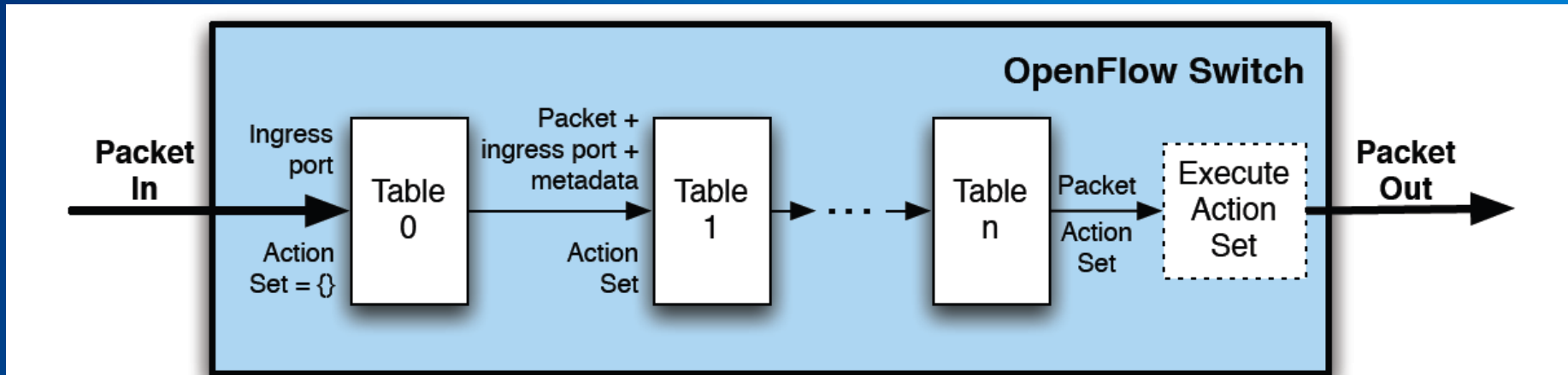
- A fixed set of match-action tables are defined in the silicon:
 - Fixed number of tables, table sizes and match fields and actions used in each table
- The application programmer tries to map the application into this fixed match-action pipeline
- **Bottoms up programming paradigm**

- No prior set of match-action tables defined in the silicon
- The application programmer creates the match-action pipeline to specifically meet the needs of the application
 - Assigns the number of tables and the size, the match fields and the actions to be used in each table
- **Top down programming paradigm**

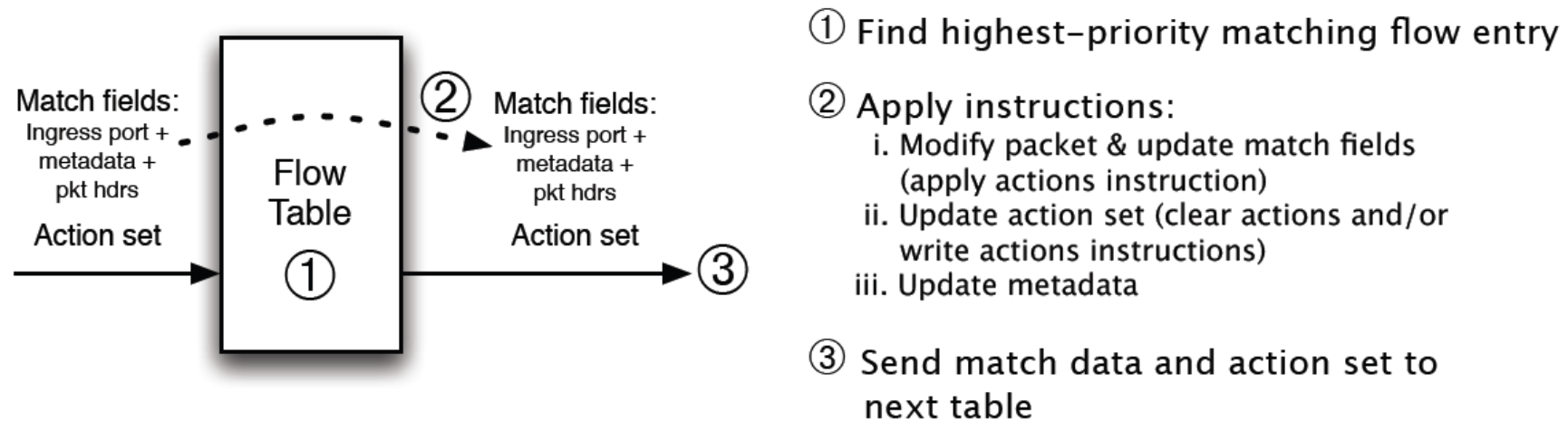
Who Cares About Programmable Match-Action Pipelines?

- Programmable Match-Action Pipelines Enables:
 - Faster introduction of new networking functionality and protocol
 - IPv6, VxLAN, SRV6, ...
 - Fast prototyping of new capabilities
 - Disaggregation of networking hardware and software
 - More like servers
 - Features are defined in the software and not in the hardware
 - No forced obsolesces of networking equipment
 - Repurposing of networking equipment

OpenFlow Match-Action Pipeline



(a) Packets are matched against multiple tables in the pipeline

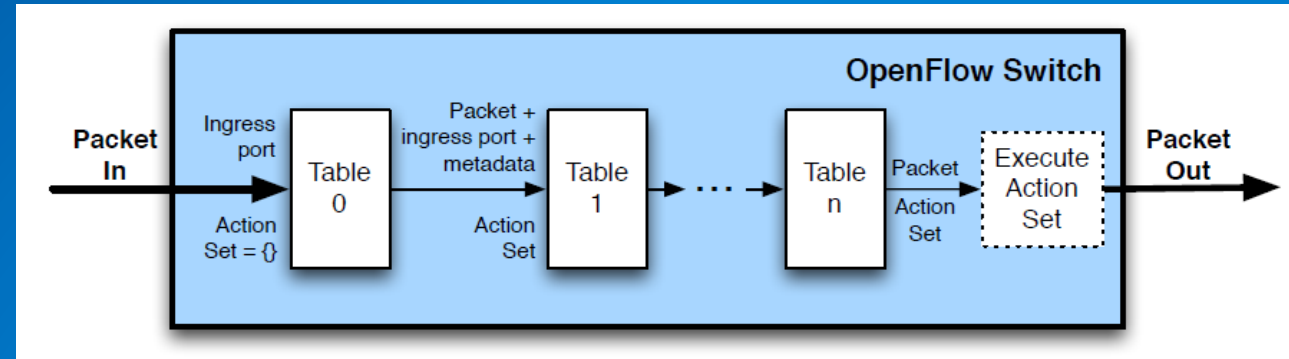


(b) Per-table packet processing

OpenFlow Match-Action Pipeline

- OpenFlow 1.4 Sample Implementation
- Provides the application programmer with a programmable match-action pipeline
- The supported match fields, instructions/actions are defined in the OpenFlow version:
 - OpenFlow 1.0, 1.1, 1.2, 1.3, 1.4, 1.5... (Protocol dependent)
 - The Experimenter Extensions allows companies to innovate:
 - Experimenter extensions for VxLAN, L2MPLS, GTP, INT,...

OpenFlow Match-Action Pipeline



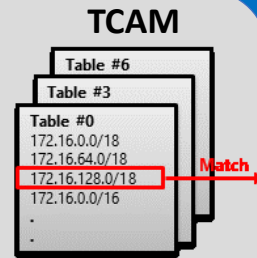
Source: OpenFlow 1.3 Specification

Fully programmable OpenFlow pipeline:

- Supports all OF 1.4 match fields (41), instructions (6) and actions (56)
- Any match field(s), action(s) and instruction(s) in any table
- Table type individually configurable:
 - Wildcard match (TCAM) (OpenFlow standard) or
 - Exact Match (DRAM)
- Each table's width and depth is individually configurable through CLI

Wild Card Matching Use Cases:

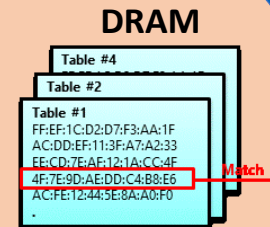
- OpenFlow standard (17/41)
- L3 Forwarding, e.g. LPM
- ACLs/Firewalls e.g. subnet matching
-



Up to **1M** flow entries with wild cards in up to **60** tables

Exact Matching Use Cases:

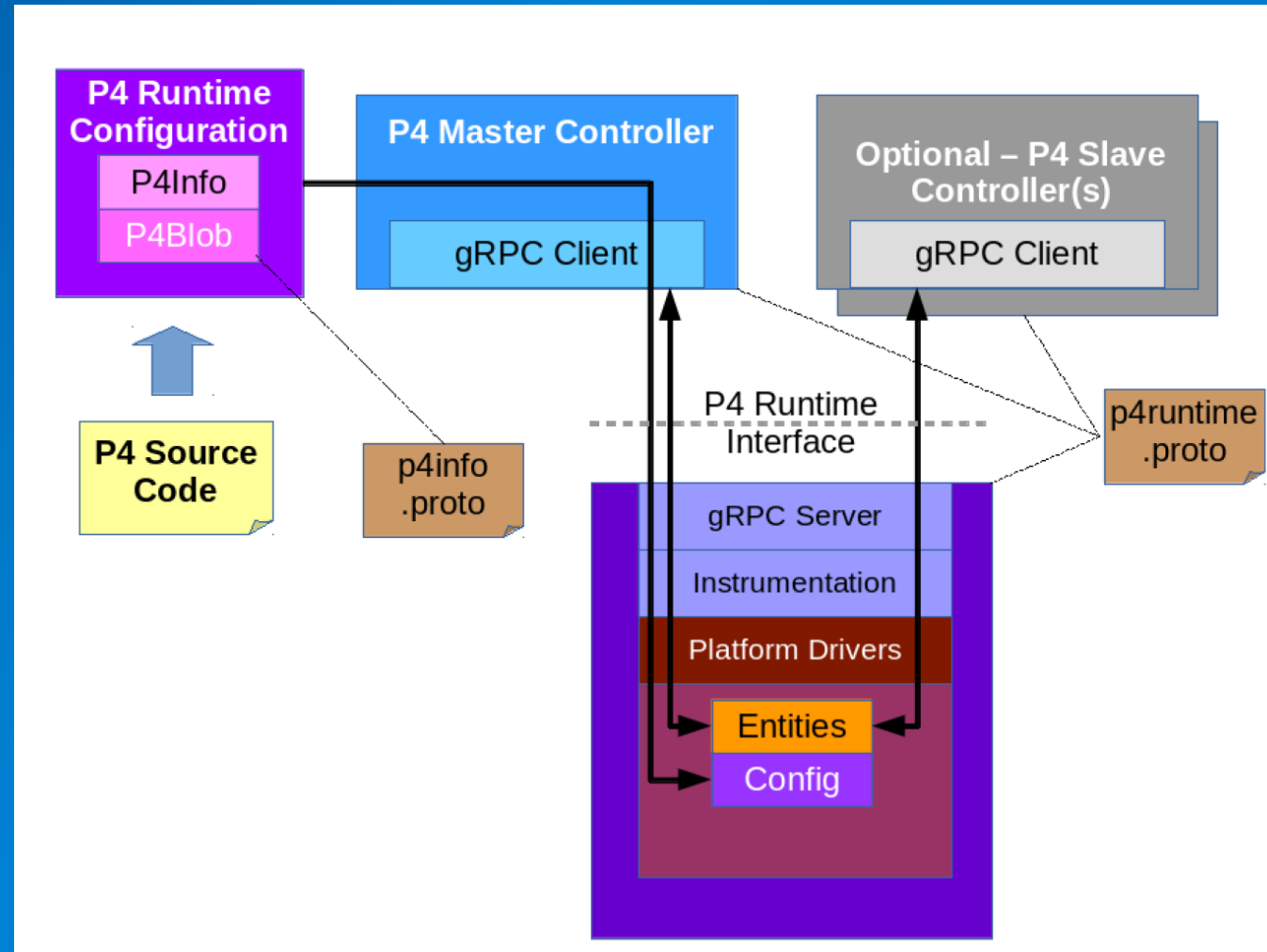
- OpenFlow standard (24/41)
- L2, MPLS Forwarding
- Segment Routing
- Service chaining
- ACLs, e.g. MAC address
- ...



Up to **6M** exact match flow entries in up to **60** tables

P4/P4Runtime Match-Action Pipeline

- P4 is a programming language used to define how a switch silicon processes packets
 - Programmable parser (match fields)
 - Programmable actions
 - Programmable match-action pipeline
- P4Runtime is an interface between a P4 Controller and a P4 programmable switch:
 - The next version of OpenFlow
 - Load a compiled P4 program into the switch silicon
 - Add/delete flow entries in the match-action tables
 - Collect statistics from the switch



Source: P4Runtime Specification

P4/P4Runtime Match-Action Pipeline

- P4 Language:
 - Header types: defines the headers
 - Parsers: defines how to parse a packet
 - Tables: contains flows
 - Actions: describes how a packet is manipulated
 - Metadata: data structures associated with each packet as it traverses the pipeline
 - Extern objects: Architecture specific constructs with well defined APIs
 - Checksum calculation
 - Registers
 - Counters
 - Meters

```
header Ethernet_h {  
    bit<48> dstAddr;  
    bit<48> srcAddr;  
    bit<16> etherType  
}
```

```
header IPv4_h {  
    bit<4> version;  
    bit<4> ihl;  
    bit<8> diffserv;  
    bit<16>totalLen;  
    bit<16>identification;  
    bit<3> flags;  
    bit<13>fragOffset;  
    bit<8> ttl;  
    bit<8> protocol;  
    bit<16>hdrChecksum;  
    bit<32>srcAddr;  
    bit<32>dstAddr;  
    varbit<320> options;  
}
```

```
table ipv4_lpm {  
    key = {  
        hdr.ipv4.dstAddr: lpm;  
    }  
    actions = {  
        ipv4_forward;  
        drop;  
        noAction;  
    }  
    size = 1024;  
    default_action = NoAction();  
}
```

Source: P4.org

Comparisons between OpenFlow and P4/P4 Runtime

- Both OpenFlow and P4/P4Runtime provides the application programmer with a programmable match-action pipeline
- Additionally, P4/P4Runtime allows the application programmer to program the parser
 - Protocol independent
 - In OpenFlow, the match fields are predefined (Note: see Experimenter Extensions below)
- Additionally, P4/P4Runtime allows the application programmer to define the actions
 - In OpenFlow, the instructions/actions are predefined (Note: see Experimenter Extensions below)
- OpenFlow supports Experimenter Extensions where the developer can define new match fields and actions
 - Used by many companies

Components of a NOS

- The OpenFlow/P4Runtime interface are only one component of what is needed in an SDN NOS
 - Configuration Management
 - Operations Management
 - Security Management
 - Extensibility
 - Telemetry



Thank You!