OpenFlow

overview

Joint Techs Baton Rouge

Classic Ethernet

- Originally a true broadcast medium
- Each end-system network interface card (NIC) received every packet
- Packets of interest, typically those with either a destination address of the NIC or a multicast address were then sent to the network stack

Extending Ethernet via a Bridge

- Bridge logic ensures that each NIC continues to see all multicast frames and unicast frames to that NIC's address
- Bridges "learn" which NICs are behind which port by snooping traffic
- Bridges operate autonomously, all state is soft, and for the most part they are operated as thought they are opaque

InCNTRE

GlobalNOC
Global Research Network Operations Center

Ethernet Switching

- Same thing as bridging, but implies more ports
- End systems continue to see unicast packets to their NIC's address as well as multicast traffic
- Both bridges and switches are implemented using content addressable memory (CAM)

Features

- Switches become much more sophisticated, needing to support filtering, QoS, etc.
- Underlying switch hardware starts to include TCAMs (cams where addressing can include wildcard bits), network processors, etc.
- Switches evolve from simply snooping source addresses and forwarding based on DAs to being much more “flow capable”

What is OpenFlow

- OpenFlow is a protocol for controlling the behavior of Ethernet switches
- At a basic level, the protocol specifies a pattern (called a Match Field) to which all incoming packets are compared, and what action(s) to apply to the packet (e.g., drop, modify and forward, send to controller, etc).

InCNTRE

GlobalNOC
Global Research Network Operations Center

The Pattern (Match Field)

Ingress Port
Ether src
Ether dst
Ether type
VLAN id
MPLS label
MPLS traffic class
IPv4 src
IPv4 dst
IPv4 proto / ARP code
IPv4 ToS
TCP/UDP SCTP src
TCP/UDP SCT dst
ICMP Code

InCNTRE

GlobalNOC
Global Research Network Operations Center

Pattern Matching

- This is not Perl
- For the most part, the bits in each of the Match Fields are compared to a bitmask
- The bitmask can contain a wildcard for any bit (starting in version 1.1 of OF)
- Some fields, such as vLAN ID don't use the bitmask
- Since a packet may match more than one pattern, patterns can have priorities (i.e., if it matches patterns A & B, since B has a higher priority B will be treated as the correct match)

InCNTRE

GlobalNOC
Global Research Network Operations Center

Packet Modifying

- For the most part, values of any of the Match Fields may be replaced
- IP header check some can be recalculated to account for port or IP changes
- IP TTL can be decremented (so an OF switch can be a legit router)
- MPLS and vLAN tags can be pushed/popped

InCNTRE

GlobalNOC
Global Research Network Operations Center

Next Hop

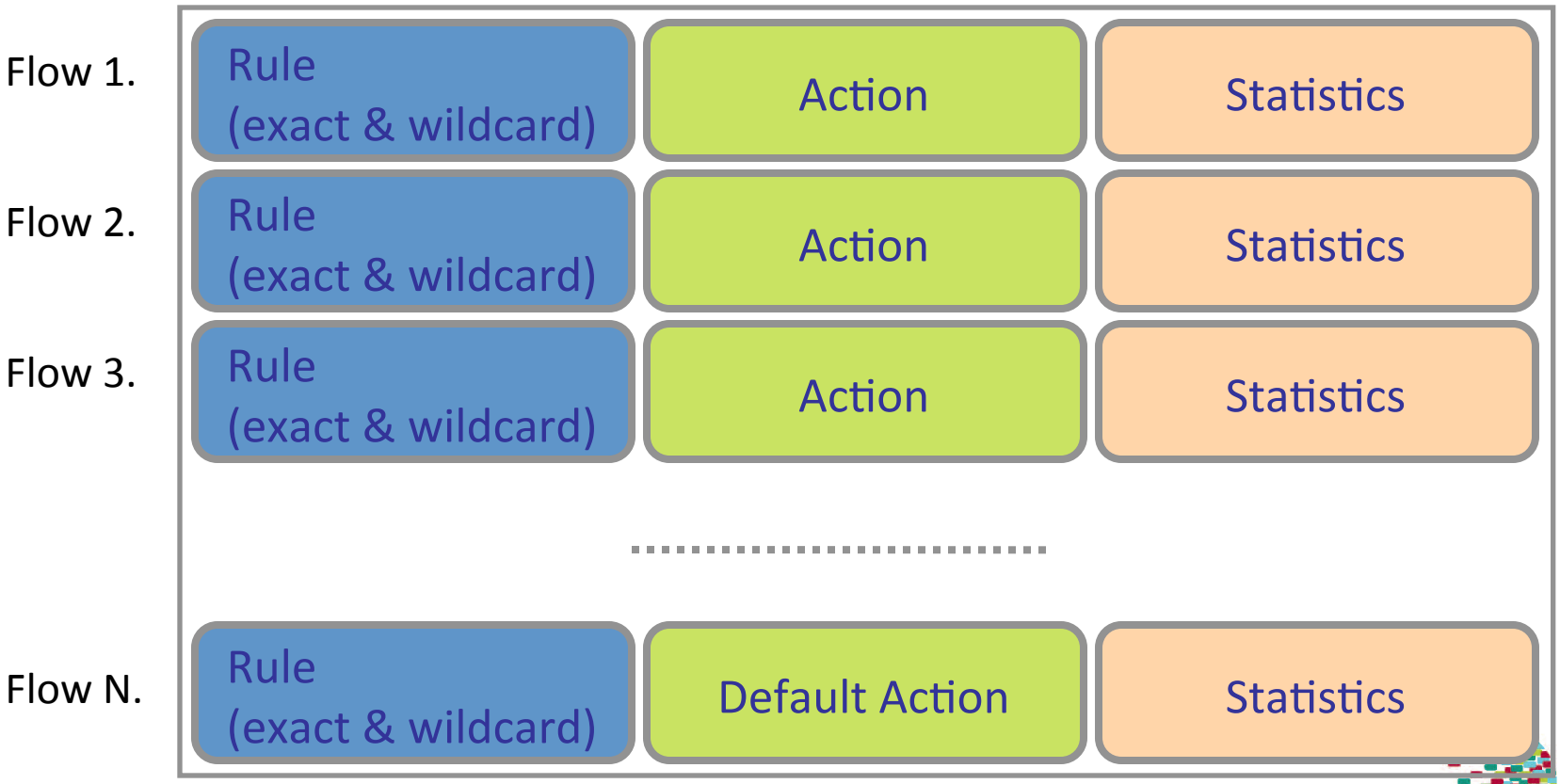
- A packet can be dropped, forwarded to one or more ports (ports can be physical or virtual)
- One standard virtual port sends the packet to the controller.
- For an OpenFlow-hybrid switch, a virtual port can point to the switches “normal” processing (i.e., outside of the OpenFlow path)

InCNTRE

GlobalNOC
Global Research Network Operations Center

OpenFlow Basics (1)

- Exploit the flow table in switches, routers, and chipsets



InCNTRE

GlobalNOC
Global Research Network Operations Center

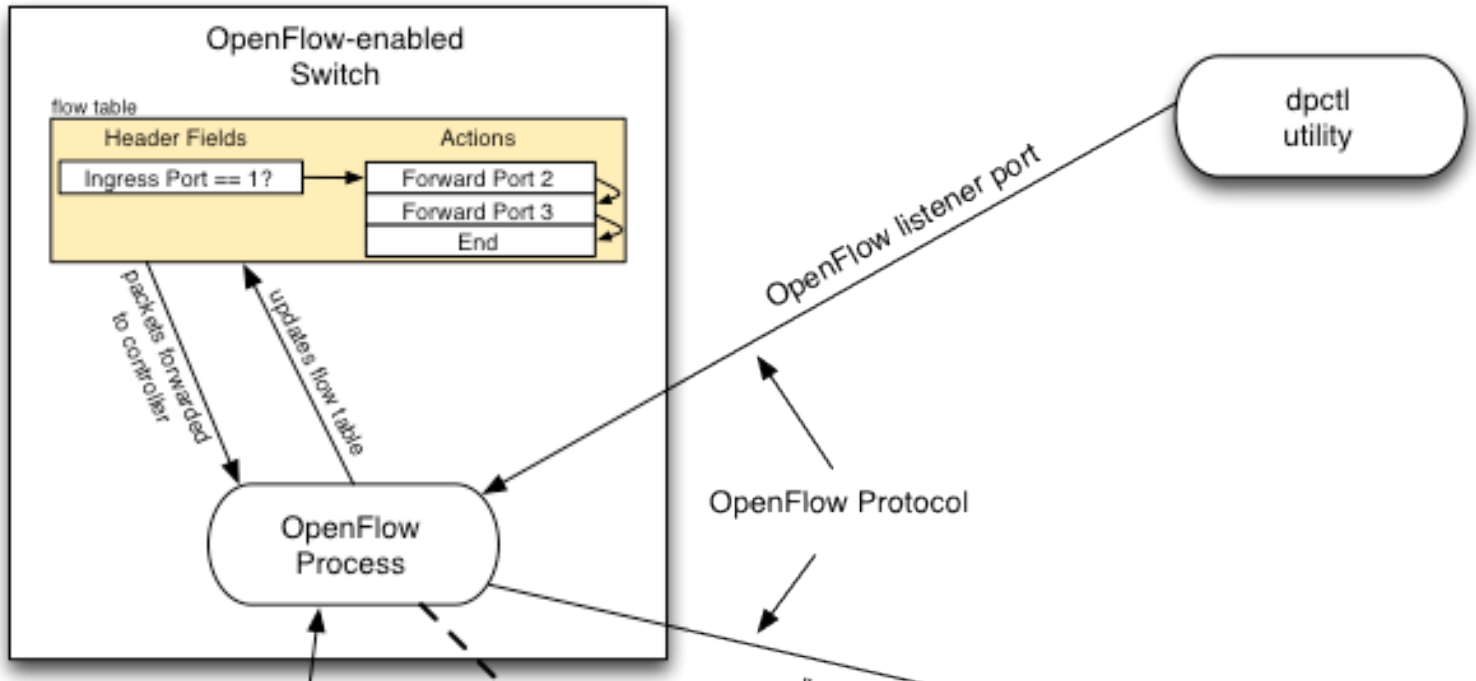
The knobs can't toast bread, but…

- The OpenFlow knobs are sufficiently fine grained to route, firewall, load-balance, fast-fail, filter, snoop, spoof, etc.
- If you want to do something with OpenFlow, you probably can write an app for that
- If eco-system matures, there will be an app for that

InCNTRE

GlobalNOC
Global Research Network Operations Center

OpenFlow Components



Most switches run a modified version
of Open vSwitch as their OpenFlow process

FlowVisor can be thought of as an
OpenFlow Controller Proxy.
OF switches connect to an instance of FlowVisor
FlowVisor then connects to OF Controllers,
exposing to each its usable flowspace