Lessons from the OpenFlow experience

CEF 2017

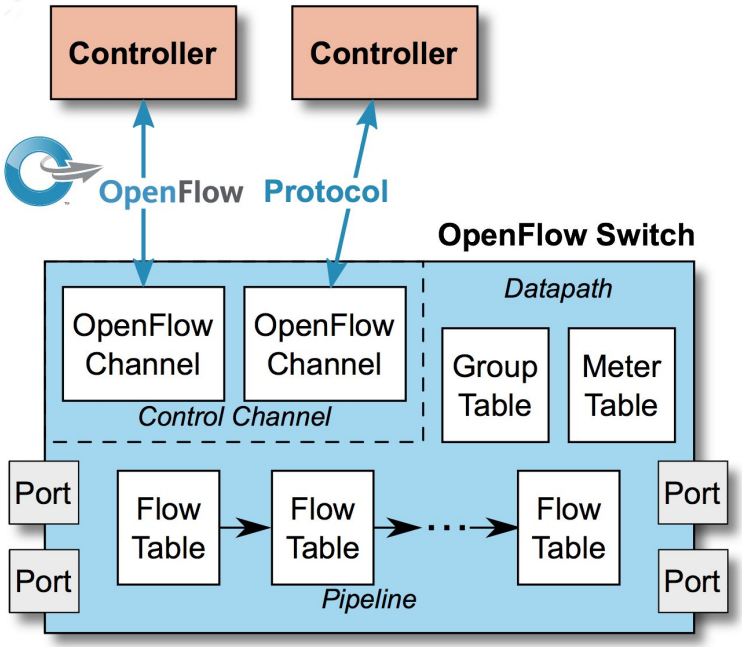
Simeon Miteff <smiteff@lbl.gov>

What is OpenFlow?

Standard and open abstraction to a match-action packet switch pipeline.

Think of it like L2/L3/L4 static routes, and a remote API to add and delete them, or tunnel packets, get statistics, etc.

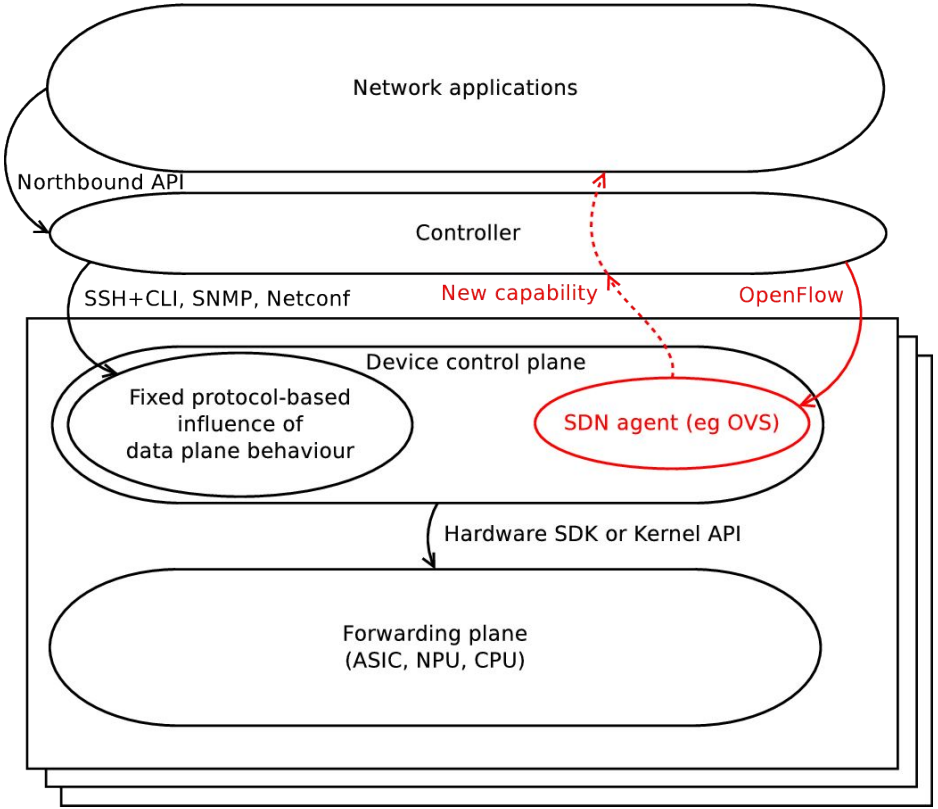
Other contenders in this space: ForCES (IETF) and proprietary vendor APIs



What is interesting about OpenFlow?

- Exposes low level match-action pipeline to the operator
- Supports wildcard match types
- Implicit deny/drop
- Open standard

The bit the runs on the switch is called the OpenFlow Agent (OFA) - the shim between the ASIC and your control plane.



High level shared objectives

What goals might next-gen packet and optical have in common?

- Collapse the stack
  - Why? (less is more, maybe cheaper, easier to maintain)
  - Challenge: optical is even more vertically integrated

- Go multi-vendor
  - Motive: optical has longer lifecycle, higher cost to fork-lift
  - Take advantage of innovations in specific components
  - Low hanging fruits: OLA example

Open+standard+programmable interfaces

- Vendor lock-in and the "single neck to ring" fallacy
  - Does your mobile provider reward you with excellent service once you sign a contract?

vs

- Multiple non-standard interfaces
  - Implicit in going multi-vendor (“manager of managers”) problem

Dis-aggregation

- Standard interfaces → interchangeable and interoperable components
  - *“low coupling and high cohesion”* as a design pattern for complex systems

- Vendors want to differentiate and compete
  - Disaggregated/standardised systems focus vendor innovation inside the component
  - Competition on the component level is a threat, <u>but also an opportunity</u>

OpenFlow: a short history

- Pre-history: Stanford, Ethane and NetFPGAs (2006 to 2008)
- MVP standard and the era of soft-switching (2009 to ~2014)
- Operator deployment and vendors catch up (2015 to now)

What went right with OpenFlow

- **Speed**: ONF got to a minimum-viable spec (OF1.3) in < 3 years

(whether 3 years is fast, and the outcome is good, is debatable!)

- **Novelty**: OpenFlow has unique capability
- **Viability**: was actually possible to implement in hardware (eventually)
- **Running code**: OVS as reference implementation and some startup success

Field		Description
OXM_OF_IN_PORT	*Required in ingress*	Ingress port. This may be a physical or logical port.
OXM_OF_ACTSET_OUTPUT	*Required in egress*	Egress port from action set.
OXM_OF_ETH_DST	*Required*	Ethernet destination address. Can use arbitrary bitmask
OXM_OF_ETH_SRC	*Required*	Ethernet source address. Can use arbitrary bitmask
OXM_OF_ETH_TYPE	*Required*	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_IP_PROTO	*Required*	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	*Required*	IPv4 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV4_DST	*Required*	IPv4 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_SRC	*Required*	IPv6 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_DST	*Required*	IPv6 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_TCP_SRC	*Required*	TCP source port
OXM_OF_TCP_DST	*Required*	TCP destination port
OXM_OF_UDP_SRC	*Required*	UDP source port
OXM_OF_UDP_DST	*Required*	UDP destination port

What went wrong with OpenFlow

- Speed came at the cost of **"standards collision"** (see: J. Halpern paper)
- ONF purist attitude did not allow for transition (step-function was a gamble)
- Misunderstanding "rough consensus and running code"::
  - "*The series of undeployable prototypes*"
  - Should standards bodies manage open-source?
  - *Open* is not sufficient, must be *useful*
- **OVS** did not address the general case
- **Unusable vendor implementations**

Non-Goals

The project does not aim to do the following:

1. It does not aim to create a Generally Available (GA) product. It will not undergo typical product Quality Assurance (QA), nor will it be ready for production or interoperate with other networks and network control planes. Instead the project will support certain elements that help in the 'productization' of SDN ideas, like ease-of-configuration, visibility and troubleshooting.
2. The project also does not aim to deliver any specific service like a full-blown bandwidth-managed TE service, a VPN/VPLS/VPWS service, or an NFV service. It will however, support the core capabilities required to build such services (and others) on top – see extensibility choices below.
3. Finally, the project does *not* aim to exclude the use of other parts, in the data plane as well as the control plane. In the interest of time and limited resources, choices will be made for building the system. However such choices should be replaceable

OpenFlow lessons

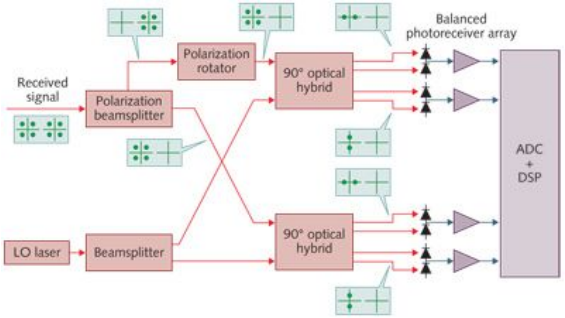
A view on getting to production deployment:

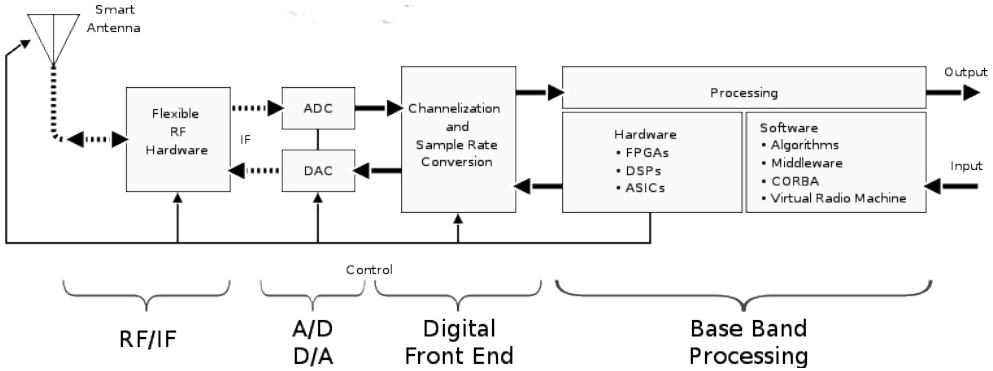
- The "running code" must be deployable in operation
- Without tests, you get "CIO+datasheet+checkbox"-compliant implementations
- Underdog vendors have an incentive to break ranks
- Sales incentive + public exposure of quality/capability

versus

Optical similarities/differences

- Your SNMP is called TL1, do your programmers like it?
  - Do your programmers really like REST+JSON better?
- Open (optical) is more operator-driven (good)
  - Are you creating a space for innovation like OpenFlow did?
- Transponders have more magic in the chips (bad)
  - Seems like an opportunity: software defined transponder?





Conclusion

On the software that can be run in production: **be extremely unambitious.**

When dealing with vendors:

- Standards compliance is best proven with open software
- *“The willing, Destiny guides them. The unwilling, Destiny drags them.”*
  - Seneca
- Money talks (create win-win situations for vendors)

End and acknowledgements

This presentation is a synthesis of wisdom I have gleaned from talking to **Inder Monga** and **Josh Bailey** (thanks guys!).

If you're interested in deployment-ready packet-SDN, please check out the FAUCET conference in Berkeley from 18-20 October (directly after Internet2 TechEx in San Francisco):

http://conference.faucet.nz/