BROCADE

Advancing OpenFlow Interoperability with TTPs

Curt Beckmann

ONF ODWG Chair, ODL TTP project lead, NFV MANO stuff

Chief Technology Architect, EMEA, Brocade

Background:

Early OpenFlow

- SDN promise: open (vendor) decoupling of control / data planes
  - OpenFlow introduced as “standard low level control protocol”
  - Many vendors offered OF-enabled boxes: Problem solved!

- OF1.0 assumed a trivial packet pipeline: 1 Match-Action table
  - Supportable on many devices, but too limiting

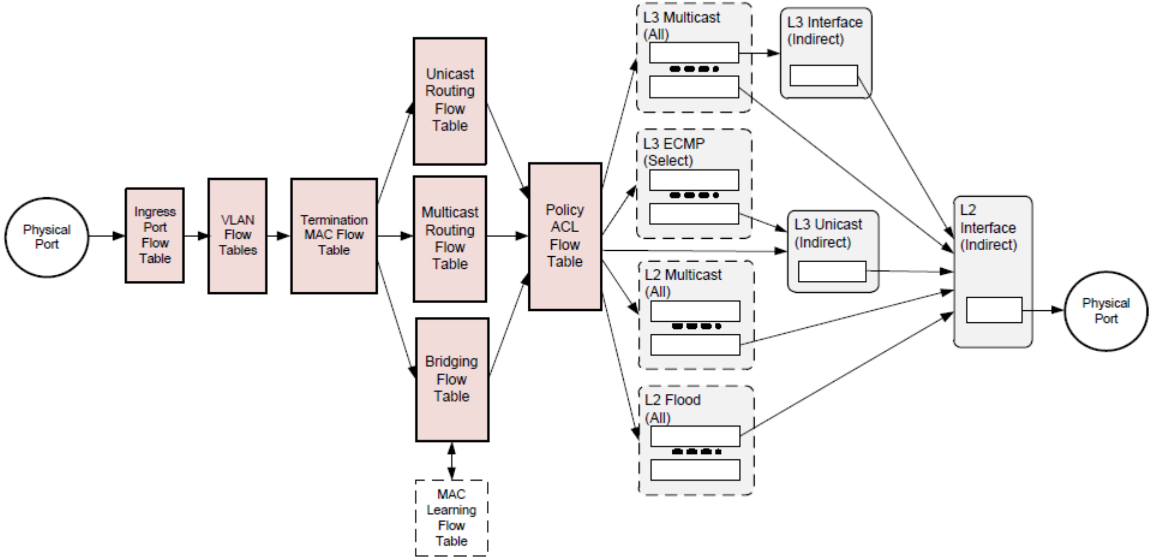
http://wyattsupply.com/products/plumbing/2-black-steel-pipe-nipple

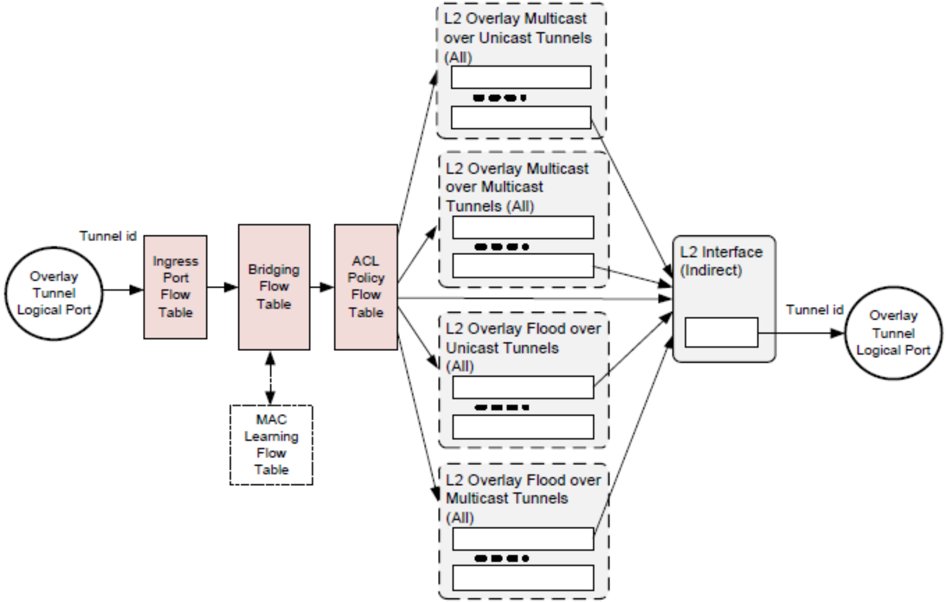
A Few Details

http://www.circleofblue.org/waternews/2011/world/mixing-art-and-technology-north-americas-largest-membrane-filtration-sewage-plant-opens-near-seattle/

- OF1.1 gave us 255 flow tables
- OF1.1 opens the door for complex packet pipelines
  - But did not acknowledge the diversity of existing, popular pipelines

Sample Pipelines




Tunnel id
Overlay Tunnel Logical Port
Ingress Port Flow Table
Bridging Flow Table
MAC Learning Flow Table
ACL Policy Flow Table
L2 Overlay Multicast over Unicast Tunnels (All)
L2 Overlay Multicast over Multicast Tunnels (All)
L2 Overlay Flood over Unicast Tunnels (All)
L2 Overlay Flood over Multicast Tunnels (All)
L2 Interface (Indirect)
Tunnel id
Overlay Tunnel Logical Port

Both are from Broadcom's OF-DPA…
See: https://github.com/Broadcom-Switch/of-dpa

Framework Gap

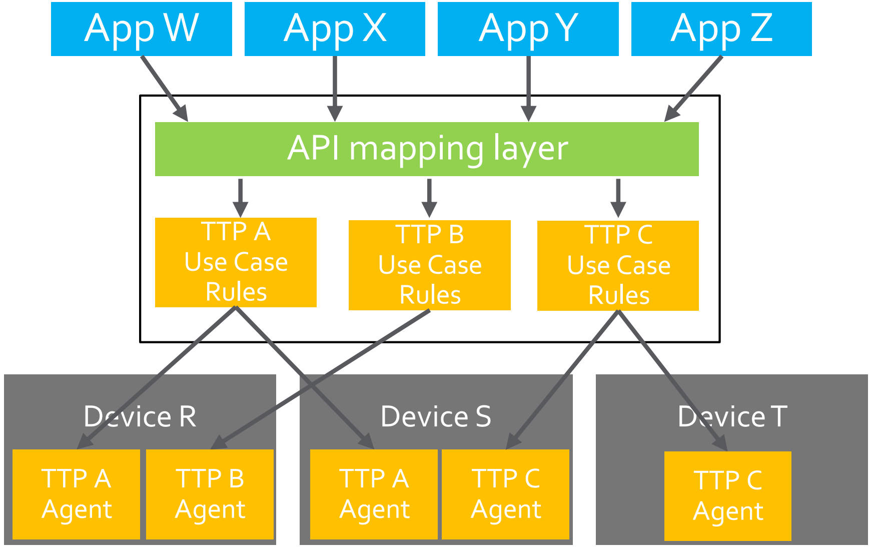
https://www.publictechnology.net/sites/www.publictechnology.net/files/styles/original_-_local_copy/entityshare/11007%3Fitok%3D4TXhvryF

- The OF framework lets the controller to send any legal OF messages
  - Device must handle them…
  - That only works with pipeline agreement
- Founders of my working group anticipated this challenge
  - We knew that OpenFlow needed to support “pipeline agreement”

Framework based on Pipeline Models

- After 1.0, OF pipeline model was no longer a **subset** of device pipelines
  - Now it is a **superset** of ASIC pipelines (and ASICs are common and useful)
- So… How to enable control of existing ASIC pipelines?
  - Run-time mapping of multi-table OpenFlow messages way too hard
  - Proposal: Figure out the mapping before run-time!
- We conceived of "Table Type Patterns" (TTPs), pipeline models
  - Plan was for switch vendors to figure out how to support some models

Early Approach: Too Switch Centric



Early approach envisioned switch vendors adding numerous TTP agents to their devices. We didn't notice these challenges

- Switches are not agile development environments
- "Uphill" given that TTPs will iterate often at first
- Don't host 3rd party code

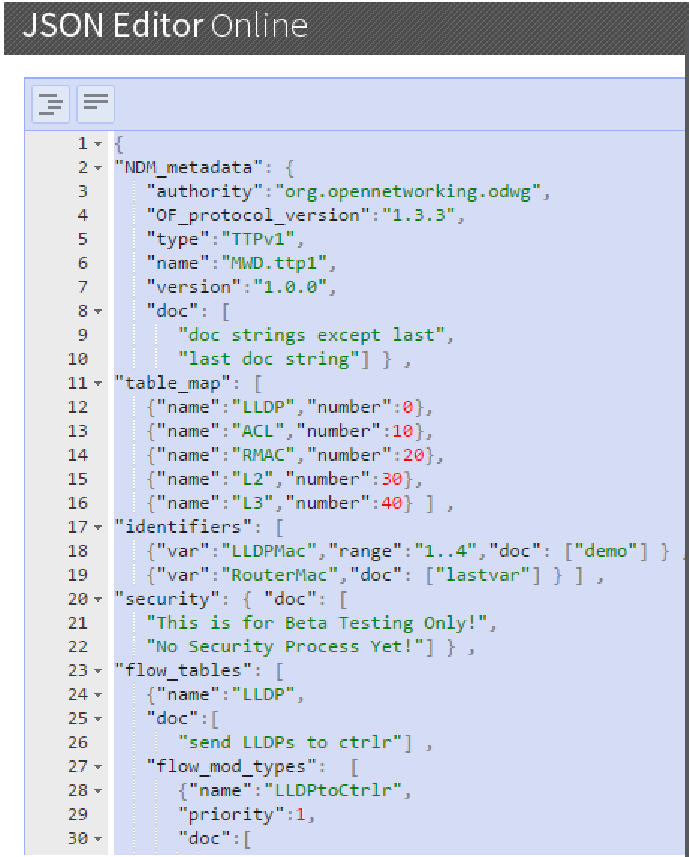
Express pipeline as “allowed OF messages”

- We expected humans to be doing most of the pipeline mapping, but we also envisioned software tools for pipeline analysis
  - So we wanted human and machine consumability
  - We biased slightly toward humans over machines

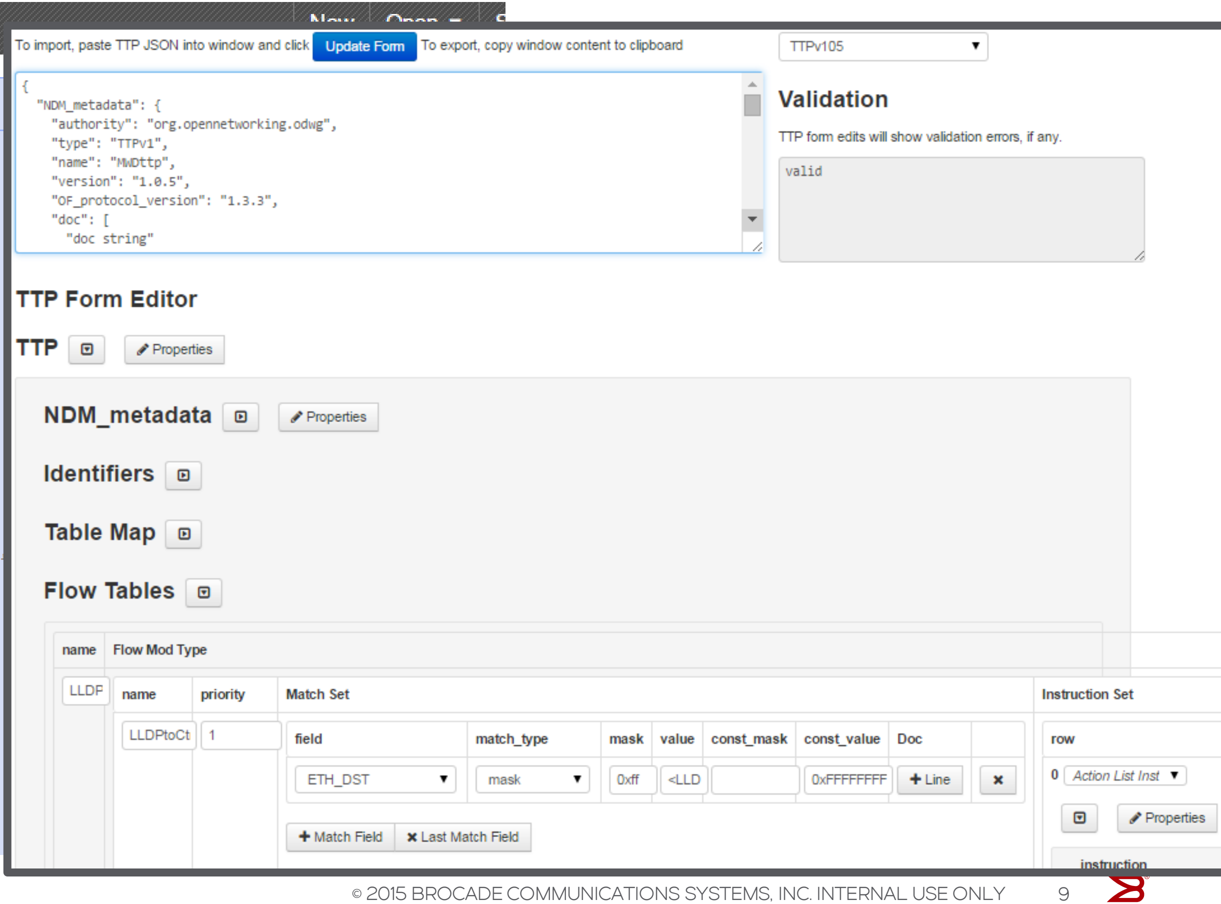
- We wanted > 1 common languages:
  - JSON, XML, YAML, whatever
- But most of our activity is using JSON

Note: P4 goes the other way: 1st focus on pipeline, do control later

TTP in JSON, and a schema-based tool



```
{
"NDM_metadata": {
   "authority":"org.opennetworking.odwg",
   "OF_protocol_version":"1.3.3",
   "type":"TTPv1",
   "name":"MWD.ttp1",
   "version":"1.0.0",
   "doc": [
      "doc strings except last",
      "last doc string"] } ,
"table_map": [
   {"name":"LLDP","number":0},
   {"name":"ACL","number":10},
   {"name":"RMAC","number":20},
   {"name":"L2","number":30},
   {"name":"L3","number":40} ] ,
"identifiers": [
   {"var":"LLDPMac","range":"1..4","doc": ["demo"] } ,
   {"var":"RouterMac","doc": ["lastvar"] } ] ,
"security": { "doc": [
   "This is for Beta Testing Only!",
   "No Security Process Yet!"] } ,
"flow_tables": [
   {"name":"LLDP",
   "doc":[
      "send LLDPs to ctrlr"] ,
   "flow_mod_types":  [
      {"name":"LLDPtoCtrlr",
      "priority":1,
      "doc":[
```



To import, paste TTP JSON into window and click **Update Form** To export, copy window content to clipboard

TTPv105 ▼

```
{
  "NDM_metadata": {
    "authority": "org.opennetworking.odwg",
    "type": "TTPv1",
    "name": "MWDttp",
    "version": "1.0.5",
    "OF_protocol_version": "1.3.3",
    "doc": [
      "doc string"
```

Validation

TTP form edits will show validation errors, if any.

```
valid
```

TTP Form Editor

TTP ⊡ Properties

NDM_metadata ⊡ Properties

Identifiers ⊡

Table Map ⊡

Flow Tables ⊡

name | Flow Mod Type

LLDF | name: LLDPtoCt | priority: 1

Match Set

field	match_type	mask	value	const_mask	const_value	Doc	
ETH_DST ▼	mask ▼	0xff	<LLD		0xFFFFFFFF	+ Line	×

+ Match Field | × Last Match Field

Instruction Set

row

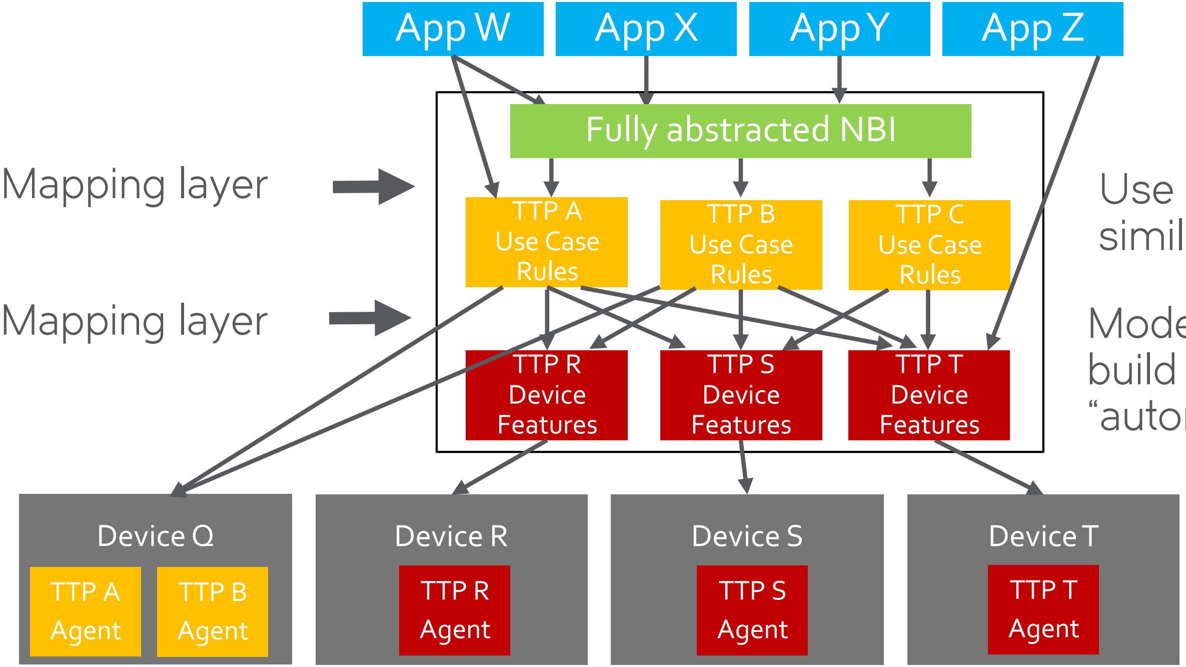
0 Action List Inst ▼

⊡ Properties

instruction

Challenges and progress intermixed

- Even before TTP spec, Broadcom produced proto TTP “OF-DPA”
  - Produced by software processing of Excel files (not human generated)
- Next, OpenDaylight was enabled (by YANG models) to import TTPs
- So… SW now generates and consumes TTPs…need to adjust TTP
  - Need to re-optimize TTP syntax for software; the tools can help humans
    - Tools like schemas and full openflow.h enum names
    - Also, schemas are fussy about their syntactic sugar…
  - Result: Schema-friendly TTPv1.1, coming soon,
- But OFDPAv2 is a device model, not mappable to other vendors



Use Case TTPs very similar to Flow Objectives

Model Driven abstraction layers build the southbound mappings "automagically"

Device supports only one vendor-centric model

Finally: The Current Problem Statement

- SDN needs scalable, hardware-independent dev platforms
  - A variety of pipeline models helps support many use cases
- Use cases can most often be mapped to > 1 ASIC pipeline
  - Each path in a “use case” pipeline model needs equivalent device path
- Mapping “use case” to “device” slow but needed to support diversity
- Can we accelerate mapping?
- OpenDaylight will help humans do the mapping
  - List all the use case path, Automate search for matching device paths
  - Help a human pick which device paths work for each use case path
- **Can this be done completely by machine? (Header Space Analysis)**

Thank you