Structure and Design of Software-Defined Networks

Teemu Koponen
Nicira, VMware
Warning:
I don’t design protocols. I write C++.

Translation: this is not your typical network talk.
AGENDA: 5 YEARS OF SDN

1. Fallacies and promise of SDN.
2. Quest for structures & abstractions in networking.
3. Practical, emerging abstractions:
   - Fabric and
   - Network Virtualization.
FALLACIES OF SDN
Traditional view of SDN

OpenFlow

OpenFlow

OpenFlow
Anything Wrong?
Centralization

Availability and scaling call for distribution.
Fine-grained Control

Low-level flow management is hard in large-scale.
Unified Hardware Pipeline

Vendors prefer differentiation and die surface matters.
Failure Domains

There will be network partitions.
Did I Lose My Faith in SDN?
No, exactly the opposite!
DISTRIBUTION IS A GOOD SERVANT...  
... BUT A BAD MASTER

**Data plane:**

- Spreading state across the network complicates the design.

**Control plane is the worst example:**

- Distributed algorithms are complex.
- Only indirect means for control.
FREEING OURSELVES FROM TOPOLOGY

Physical topology should not drive the control design.

Decoupling control from topology:

- Allows for simpler overall design.
- Allows for distributed systems principles.
- Allows for reconsidering the network design.
Quest for Structures & Abstractions in Networking
WHERE ARE WE TODAY?
How Did We Get Here?

A guaranteed recipe for disaster:

1. Invent a new data plane mechanism.
2. Hack a new control plane for it.

Physical topology drove the control design!
The Missing Principles of Modularity

Separation of Concerns
- Contain a problem.
- Contain its solution.
- Modules get decoupled.

Hiding Complexity
- Abstractions hide details from the rest.
- Harmful dependencies don’t creep in.

But...
DID SOMEONE SAY IT WOULD BE EASY?

- **No firm structures for the control plane after past 30 years of networking.**

- **Abstractions and structures are trivial to miss.**

  - **You have to practically stumble into them.**

  - **Exactly that happened to us.**

- **Following is not about paper designs!**
STRUCTURE I : FABRIC
MPLS: Argument for Edge/Core Split

Routing  |  Simple Forwarding  |  Routing
---------|---------------------|---------
Router   | Switch              | Router
MPLS Edge| Domain Internal     | MPLS Edge
Neighbor Domain | Domain Internal | Neighbor Domain
Hypervisors Are The New Edge

Simple Forwarding

Domain Internal

Host
Hypervisor
VM

Switch

Switch

Host
Hypervisor
VM
**Benefits: Separation of Concerns**

**Edge**
- Implements the network policy.
- Correctness & flexibility of function essential.
- Speed less.

**Fabric**
- Moves the packet across the network.
- Simplicity of function.
- Reliability and capacity critical.

Edge and fabric can evolve independently with different goals!
Clear Divide Between HW & SW

**Edge**
- Correctness & flexibility of function essential.
  - Speed less so.
  - x86

**Fabric**
- Reliability and capacity critical.
  - Simple & stable func.
  - Simple ASICs
Structure 2: Network Virtualization
Hiding Complexity

- Today configuring a policy spans all elements.
- Simple policy: A can’t talk to B.
- Why should admin consider any details between?
Network Virtualization

Virtual Switch

Virtual Router

Switch

Switch

Switch

Switch

Switch

Virtual

Physical
Generalization: Nested Virtualization

Tier 1

Tier 2

Tier 3

Tier 4
## Identifying The Scope of Virtual Switch

<table>
<thead>
<tr>
<th>Locality of Connectivity</th>
<th>Virtual switch spans an area of good connectivity.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Domains</td>
<td>Virtual switch corresponds to a single failure domain</td>
</tr>
<tr>
<td>Separation of Policies</td>
<td>Virtual switch interface hides lower level policies from above.</td>
</tr>
</tbody>
</table>
Fabric + Virtualization

Virtual Switch

Virtual Router

Virtual

Physical

Switch

Switch

Switch

Switch

Edge

Fabric

Edge
VIRTUAL SWITCHES, ROUTERS, SERVICES...
**Structures Summary**

**Fabric**
- Hardware to deliver packets across network.
- Software edge implements the semantic intelligence.

**Virtualization**
- Virtual topologies shield user from low-level details.
- Admin focuses on policies.
- Implemented at the edge.
Conjecture: Software Will Take Over
LONG TIME AGO...

Software \rightarrow Control Plane

Software \rightarrow Data Plane
More Recently...

Software → Control Plane → Data Plane → Hardware
Today In Virtualized Environments

Software → Control Plane

Hardware → Fabric Data Plane → Edge Data Plane → Software
### Clear Divide of Roles

<table>
<thead>
<tr>
<th>Hardware Fabric</th>
<th>Software Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>High aggregate bandwidth, reliability</td>
<td>Highly flexible provider for network semantics</td>
</tr>
</tbody>
</table>

A modern version of end-to-end principle!
SOFTWARE FORWARDING NEVER WORKS...

This time it’s different:

- Hardware remains responsible for aggregate bandwidth.
- Similarly, high fan-out.
- Software forwarding is a tax for servers.
- Using a few percent of CPUs not really an issue.
**Change of Design Priorities**

**Strict HW limitations (minimize die surface)**

- Control plane design adapts

**System Complexity**

**Simple control plane design**

- Software forwarding adapts (flexible matching and operations)

**Simpler System**
Networks as Systems
Classic Layered Control Planes

- Carrier Ethernet
- MPLS
- TRILL
- OSPF
- IS-IS
- STP
- Ethernet
- BGP
- Regions
- Sites
- Subnets
- WAN
Layered Control Planes in SDN
Forget the Protocols
EMPEROR’S NEW CLOTHES

Hacking with NOX requires no knowledge about protocols.

Network control at any scale is no different:

Any distributed systems developer can arrive with a design!

Remember failure domains, locality and separation of policies...

What special is left in networking after this?

We can reason about networks as plain, standard systems, not as a collection of distributed protocols!
Takeaways
**SUMMARY**

- **SDN** is an opportunity to revisit the network design.
  
  - **Fabric & network virtualization** good examples.
  
- **Classic assumptions** are quickly changing:
  
  - Software forwarding takes over hardware as the provider of network semantics.
  
  - Distributed systems replace the bag of control protocols.
  
- This is exactly what "Software" in SDN means!
Questions?