Programmable Network Fabrics

Nick McKeown and Sachin Katti

Lavanya Jose, Lisa Yan, Kanthi Nagaraj, Dinesh Bharadia, Mohammad Alizadeh and George Varghese
The Programmable Network Platform

Declarative approach to program the network.
Three components:

• Replacing fixed-function switch chips with reconfigurable switch chips
• P4, a language to program them
• xFabric, a system layered on top of P4 to express and realize application-layer objectives
Fixed-Function Switch Chips

Parser

L2 Stage

IPv4 Stage

IPv6 Stage

ACL Stage

Queues
Control Flow Graph

Switch Pipeline

L2 Table
IPv4 Table
IPv6 Table
ACL Table

Queues
Fixed-Function Switch Chips Are Limited

1. Can’t add new forwarding functionality
2. Can’t add new monitoring functionality
Fixed-Function Switch Chips

Control Flow Graph
Switch Pipeline
Fixed-Function Switch Chips Are Limited

1. Can’t add new forwarding functionality
2. Can’t add new monitoring functionality
3. Can’t move resources between functions
Mapping Control Flow to Reconfigurable Chip.
Reconfigurable Switch Chips

Control Flow Graph
Switch Pipeline
Protocol Independent Switch
Reconfigurability: the norm in 5 years

- Reconfigurability adds mostly to logic.
- Logic is getting relatively smaller.
- The cost of reconfigurability is going down.
- Fixed switch chip area today:
  - I/O (40%), Memory (40%),
  - Wires, Logic
Packet Forwarding Speeds

<table>
<thead>
<tr>
<th>Year</th>
<th>Gb/s (per chip)</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990</td>
<td>0.1</td>
<td>Fixed/microcoded</td>
</tr>
<tr>
<td>1995</td>
<td>1</td>
<td>Fixed/microcoded</td>
</tr>
<tr>
<td>2000</td>
<td>10</td>
<td>Fixed, closed</td>
</tr>
<tr>
<td>2005</td>
<td>100</td>
<td>Fixed, closed</td>
</tr>
<tr>
<td>2010</td>
<td>1000</td>
<td>Fixed, closed</td>
</tr>
<tr>
<td>2015</td>
<td>3.2Tb/s</td>
<td>Programmable, open</td>
</tr>
<tr>
<td>2020</td>
<td>100000</td>
<td>Programmable, open</td>
</tr>
</tbody>
</table>

- Fixed/microcoded: fixed, closed
- Fixed, closed: fixed, open
- Programmable, open: switch chip

Graph showing the increase in packet forwarding speeds from 1990 to 2020.
Fixed Function Broadcom Tomahawk: 3.2 Tbps
Reconfigurable Cavium Xpliant: 3.2 Tbps
Configuring Switch Chips

P4 code

Compiler

Compiler Target

Parser

Match Table
Action Macro

Match Table
Action Macro

Match Table
Action Macro

Match Table
Action Macro

Queues
P4 (http://p4.org/)

Parser (ANCS’13)

```
parser parse_ethernet {
  extract(ethernet);
  select(latest.etherType)
  {
    0x800 : parse_ipv4;
    0x86DD : parse_ipv6;
  }
}
```

Match Action Tables

```
table ipv4_lpm {
  reads {
    ipv4.dstAddr : lpm;
  }
  actions {
    set_next_hop;
    drop;
  }
}
```

Control Flow Graph

```
control ingress {
  apply(l2_table);
  if (valid(ipv4)) {
    apply(ipv4_table);
  }
  if (valid(ipv6)) {
    apply(ipv6_table);
  }
  apply (acl);
}
```
Which data plane? Any data plane ...
Is this enough?
What does the programmer desire?

• Our goal: make it easy for average Jo Programmer to quickly write and scale a distributed application
  – Probably doesn’t know how to use P4 and reconfigurable switch chips
  – But understands what she needs from the network in terms of her distributed application
Programmers have rich variety of preferences from their applications

• Minimize job completion time for a complex task with multiple stages

• Network goal: Minimize completion time of a collection of flows making up an application
Programmers have rich variety of preferences from their applications

- Coflows: Minimize completion time of a collection of flows making up an application
- Ramcloud: Minimize RPC flow latency
- Google’s WAN: Flexible, hierarchical, weighted bandwidth allocation
- Facebook’s memcache: Prevent bursts & incast
- Many others ...
xFabric in one slide

- Convert programmer preferences to per-flow utility functions of rate
- Network links calculate prices depending on local information: congestion and marginal utilities of flows traversing
- Senders adapt their flow rates depending on network path price and their marginal utility

→ Provably optimal network bandwidth allocation for specified preferences & fairness policy
Example: Minimize flow latency

• Programmer’s preference: Minimize flow completion time

• Convert to per flow utility function: \( x_{\downarrow i} / s_{\downarrow i} \)
  – \( x_{\downarrow i} \) is rate of flow \( i \), \( s_{\downarrow i} \) is size of flow \( i \)

• Network’s goal:

\[
\max \sum_{i} x_{\downarrow i} / s_{\downarrow i} \\
\text{s.t } R x \leq C \quad \text{(link capacity constraints)}
\]

• Encodes intuition that providing the maximum rate to the smallest flows maximizes overall network utility
What happens at the sender?

xFabric senders use standard DCTCP but add the following fields to the packet header:

• Network price: sum of prices along packet’s path, filled in by the switches

• Residue: difference between marginal utility at current rate and previous path price

• Weight: inverse of the utility function at the current path price
What happens at each link?

For each packet in the queue:

• Switches allocate bandwidth across all flows according to their relative weight from the packet header using WFQ

• Switches add the current price of the link in the network price header for each outgoing packet

• Update link price for the next epoch based on link utilization and flow utility residues
xFabric’s Guarantee

• The above protocol provably maximizes the overall network utility for any convex per-flow utility function
  → Optimal realization of declarative programmer preferences and operator fairness policy

• Single protocol capable of accommodating a large range of bandwidth allocation goals by appropriately choosing the utility functions
  – Flow latency, co-flow latency, proportional fairness, per-tenant weighted allocation, tail latency
xFabric & P4

xFabric

P4 code

Compiler

Compiler Targets

Reconfigurable stack at the end-hosts

DCTCP

Parser

Match Table

Match Table

Match Table

Match Table

Weight Update

Residue Update

NIC

Reconfigurable switches (both SW edge and HW switches)

Parser

Match Table

Match Table

Match Table

Match Table

Min Residue est.

Utilization est.

Update link price

Add price to hdr.

Queues
Programmable Network Platform

• Declarative interface to program the network according to application layer requirements
• Raises the productivity floor for the average programmer
• Open questions:
  – P4 compilers (Lavanya’s talk)
  – End-host datapath reconfigurability
  – In-band WFQ at the switches
  – Translating complex programmer preferences to utility functions