Update on P4
Changhoon Kim, Barefoot Networks / P4.org
A New Future: Programmable data planes come to the mainstream

• Some devices are more programmable than fixed-function ASICs
• CPUs: 10s of Gb/s
• FPGAs & NPUs: 100s of Gb/s
• Protocol-Independent Switch Architecture (PISA) ASICs: Tb/s
  • A few emerging solutions
  • Merchant silicon with fully programmable data plane with effectively zero penalty
  • In next few years this kind of data plane will be inevitable for high-speed devices
“Turning the tables”

Switch OS and Apps

Run-time API

Driver

Programmable data plane
(such as PISA, NPU, FPGA, and CPU)

“This is exactly how I want to process packets”

in P4
What does this mean?

• Data-plane changes at the velocity of S/W development and release
  • Extremely fast iteration and feature release
  • Best of breed S/W programming practices and tools leveraged

• Networking features and behaviors custom-built for your apps, policies, and architecture

• Transparent data plane
  • No more “black boxes” in the “white boxes”; no more ambiguity in forwarding behavior

Finally, create the network you want!
Enable tons of beautiful ideas!

For better and easier network management

- Re-allocate H/W resources or even simplify the data plane in the field
- Customizable visibility and telemetry for debugging and diagnostics
- Verify network behavior
- Improve OAM capabilities
For better application performance, lower cost, and more …

- Better network-fabric load-balancing to expedite job completion
- New standard or custom encapsulations and/or headers
- Improve network reliability and robustness
- Embed some middlebox functions inline
- Enhanced congestion control
- and many more …

We’ve only started to scratch the surface
P4 Language Consortium -- P4.org

- **Build an open community**
  - Open-source software – Apache license
  - A common language – Recently released P4 spec v1.1
  - Support for various devices – Physical & virtual SWs, host networking stacks, NICs, middleboxes
  - Support for various targets – PISA chips, FPGAs, NPUs, embedded or server-class CPUs

- **Enable a wealth of innovation**
  - Diverse “apps” (including proprietary ones!) while leveraging commodity targets

- **With no barrier to entry**
  - Free membership and simple software licensing
### P4.org Membership Growth

Original P4 paper authors from Barefoot, Google, Intel, Microsoft, Princeton University, and Stanford University

<table>
<thead>
<tr>
<th>Operators</th>
<th>Systems</th>
<th>Targets</th>
<th>Academia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibaba Group</td>
<td>BROCADE</td>
<td>AEPONYX</td>
<td>Intel</td>
</tr>
<tr>
<td>at&amp;t</td>
<td>CISCO</td>
<td>Barefoot Networks</td>
<td>Mellanox</td>
</tr>
<tr>
<td>Baidu</td>
<td>CorSAS</td>
<td>Broadcom</td>
<td>Marvell</td>
</tr>
<tr>
<td>COMCAST</td>
<td>Dell</td>
<td>Cavium</td>
<td>MoSys</td>
</tr>
<tr>
<td>kt</td>
<td>Hewlett Packard Enterprise</td>
<td>Centec Networks</td>
<td>Netronome</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Huawei</td>
<td>EZchip</td>
<td>NoviFlow</td>
</tr>
<tr>
<td>SK telecom</td>
<td>Inventec</td>
<td>Freescale</td>
<td>Plumgrid</td>
</tr>
<tr>
<td>Tencent</td>
<td>Juniper Networks</td>
<td>GLOBAL HITECH</td>
<td>Xilinx</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to get involved with P4 – Go to http://p4.org

• Try P4 development tools and reference programs (switch.p4 and more)
  • Including P4-programmable S/W switches and test framework
  • Exciting apps for network monitoring, analysis, diagnostics, and control

• Join the events
  • P4 tutorial for developers on May 23 – Register at P4.org
  • P4 workshops – e.g. 2016 Annual P4 Developer Workshop on May 24
  • P4 boot camps, hackathons, and more

• Join the consortium and the community
  • Join the mailing lists and participate in the discussions

• Get familiar with P4 and contribute back!
P4 evolution roadmap

• Cleanly embrace functional and architectural heterogeneity

• Enable code reuse
  • Portability – *Reuse the P4 code for various targets*
  • Architecture-language separation – *Reuse the same compiler for new targets*
  • Composability – *Write P4 code once and reuse it many times*

• More working groups

• Enabling bigger long-term impact across all parts of the network
  • Working with other open-source efforts, such as OpenSwitch, IOVisor, OVS, and DPDK
Key benefits of programmable forwarding

1. **New features**: Realize new protocols and behaviors very quickly
2. **Reduce complexity**: Remove unnecessary features and tables
3. **Efficient use of H/W resources**: Achieve biggest bang for buck
4. **Greater visibility**: New diagnostics, telemetry, OAM, etc.
5. **Modularity**: Compose forwarding behavior from libraries
6. **Portability**: Specify forwarding behavior once; compile to many devices
7. **Own your own network**: No need to wait for next chips or systems
Thank You