

NETWORK WORLD

# Features

May 5, 1986



## SDNs burst forth

Software-defined networks are not for everyone. Users with many distributed locations and large volumes of voice/data traffic will benefit most from this technology. But for many users, a combination of the dial-up network, Wats, and leased lines is less expensive.

**This page.**

## Making cents out of SDN

Many interrelated



*SPECIAL SECTION:  
Software-defined networks*

# SDNs burst forth

Software-defined networks burst into the telecommunications market last year with the promise to provide service levels for voice and data transmission that are equal to yet cheaper than those attainable with private leased-line networks.

figuring a private network, such as adding or relocating circuits, takes one to four months, depending on the carrier. In addition to being inconvenient, such changes are expensive.

All software-defined networks implement a standard seven-digit dialing scheme that makes



unbothered. moisturized. happy. in my lane.  
focused. ~~flourishing.~~

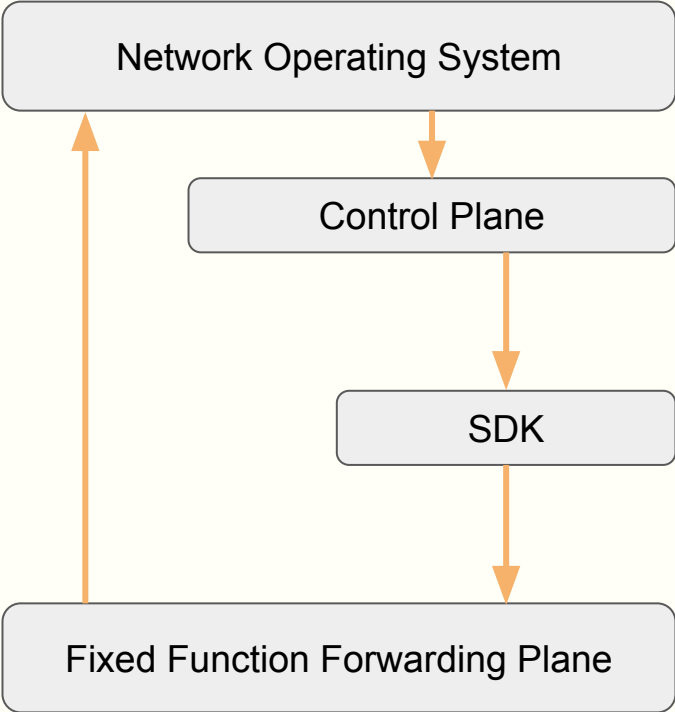
### CLI Command

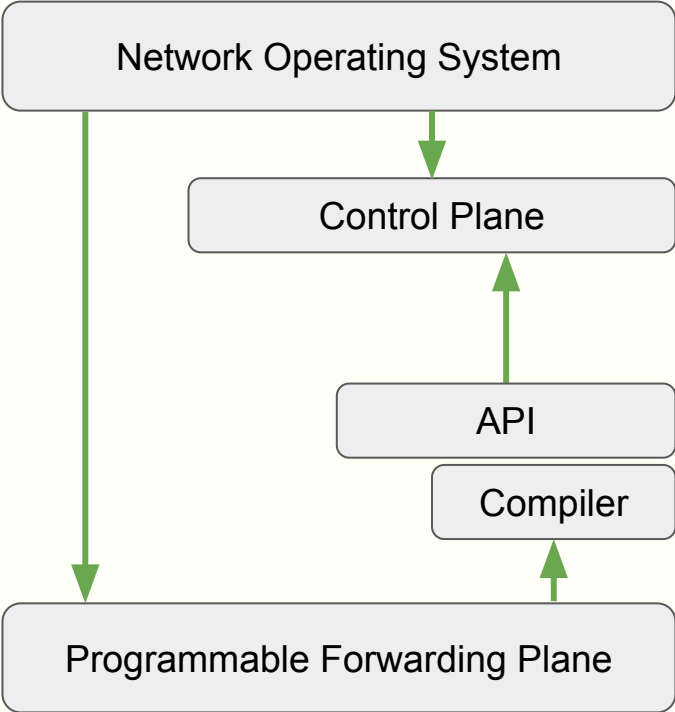
**show interfaces t3-5/1/0:0**

**show bgp neighbor 10.168.1.122**

### JUNOS XML Tags

```
<rpc>  
  <get-interface-information>  
    <interface-name>t3-5/1/0:0</interface-name>  
  </get-interface-information>  
</rpc>  
<rpc>  
  <get-bgp-neighbor-information>  
    <neighbor-address>10.168.1.122</neighbor-address>  
  </get-bgp-neighbor-information>  
</rpc>
```





# Looking Lower

## Programmable Data Planes

**Aaron A. Glenn**, Principal Network Specialist @ Predicted Paths BV

This presentation is RFC1925 compliant

*“A user interface is well-designed when the program behaves exactly how the user thought it would.” – Joel Spolsky*

# A Domain Specific Expression

We are interested in processing packets

Processing packets involves parsing headers, and manipulating bits with a side of simple logic\*

No requirement for the complexity (and pitfalls!) of a general purpose language like C++ or Rust

Those languages are for your control & management plane

\*often you need great amounts of simple logic



# The P4<sub>16</sub> Language

A domain specific language to *describe the behavior* of a packet processing forwarding plane

Express packet processing *application logic* & define capabilities of a packet *processing architecture* in the same language

Begin at [github.com/jafingerhut/p4-guide](https://github.com/jafingerhut/p4-guide) for more in-depth

Codify processing logic and underlying forwarding system capabilities through the Match-Action abstraction

# Reconfigurable Match-Action Table

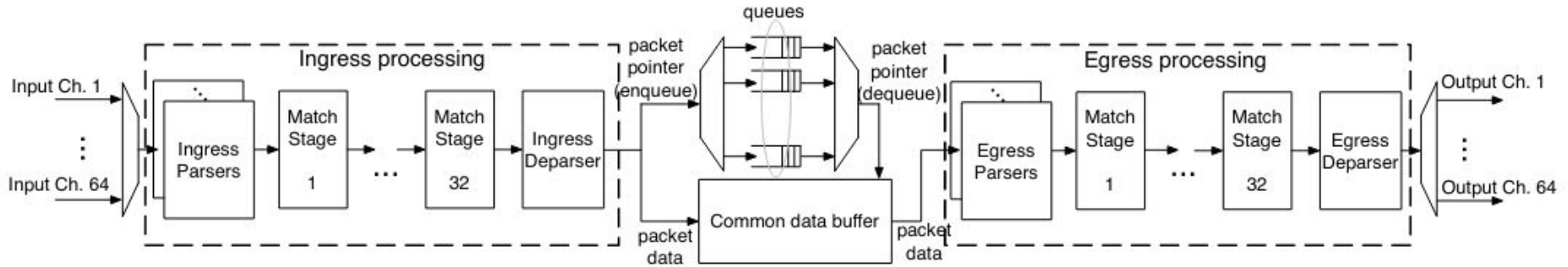
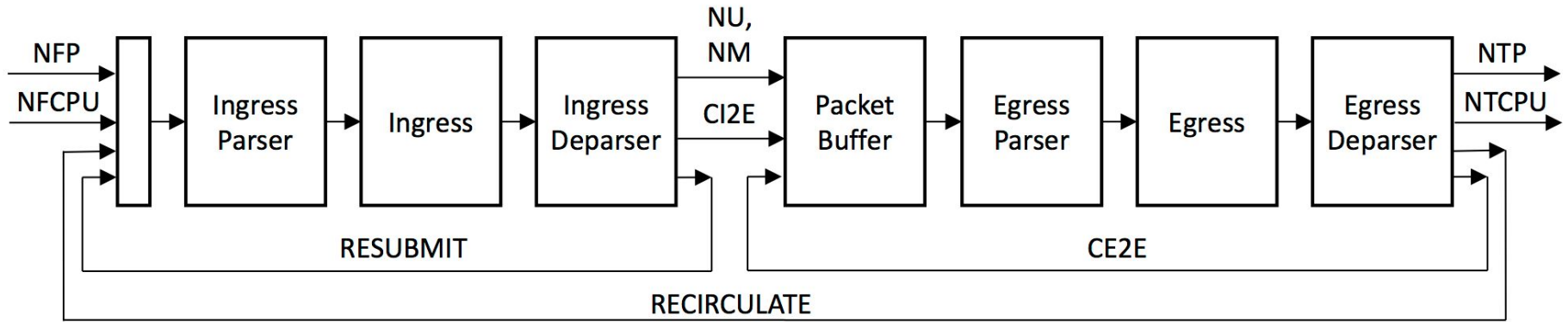


Figure 3: Switch chip architecture.

# Portable Switch Architecture

“P4 programs specify how the various programmable blocks of a target architecture are programmed and connected. The Portable Switch Architecture (PSA) is a target architecture that describes common capabilities of network switch devices that process and forward packets across multiple interface ports.” – from [p4.org/p4-spec/docs/PSA.html](https://p4.org/p4-spec/docs/PSA.html)

# Portable Switch Architecture



# Domain Specific Value Proposition

What do we want? Resilient and tractable networks

What does that mean? Understand how it operates as to configure and interact(?) with it

Infrastructure as Code – representation can be translated / compiled to different interfaces. See P4 to VPP. Domino to P4. NPL to P4.

“Artisanal MPLS” – bespoke protocols

Powerful for simplicity, differentiation, representation

But also, power.

# Reconfigurable Responsibilities

Power dynamics between operator and vendor are *very* different in a programmable data plane environment

Vendor provides device / system as well as compiler: Bad compiler, bad experience, no useful outcomes.

Road-map, resource prioritization, application support in hands of operators

Hardware vendors need to excel at software

*“You should download an MPLS stack, actually”* - Aaron A. Glenn

# The Power Problem

As bandwidth goes up, so does power draw. ASIC + optics = Watts

Network operations takes many people just to move data around

What if the network could off load common computational patterns?

A programmable network can

*“Show me someone who doesn’t want 10% or more of their general compute free’d up for general computing tasks!” - Aaron A. Glenn*

# In-Network Computing

Sapio, Amedeo, et al. "**In-network computation is a dumb idea whose time has come.**" *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. 2017.

Caching, Consensus, & Coordination

Key / Value stores

String & Pattern matching

Machine Learning!



# The Software Problem

With great power comes great responsibility...

...but also software engineering is a commercially mature (and more common) practice

P4 is incomplete! Come fill it out with us!

The Independent Software Vendor model for network protocols & stacks

# Questions & Cool Research

OK, now your turn!

“TEA: Enabling State-Intensive Network Functions on Programmable Switches”

“RedPlane: Enabling Fault-Tolerant Stateful In-Switch Applications”

“Lyra: A Cross-Platform Language and Compiler for Data Plane Programming on Heterogeneous ASICs”

“Lucid: A Language for Control in the Data Plane”

“Aquila: A Practically Usable Verification System for Production-Scale Programmable Data Planes”