

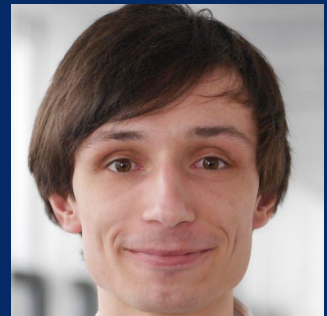


Stratum Techinar

July 27, 2021 | 9am PDT



Brian O'Connor
Stratum Technical Lead



Maximilian Pudelko
Stratum TST Member

Outline



- Quick Stratum Intro
- Stratum Release Highlights (21.03 & 21.06)
 - API Enhancements
 - Tofino enhancements
 - Debugging enhancements
- Stratum Roadmap
 - Supporting production deployments
 - Stratum as an application
- Switch OS Landscape
 - Bringing Stratum capabilities to other switch OSes
 - Areas of intersection

What is Stratum?



Open source, production targeted, **thin** switch OS

Thin means bring your own *data plane pipeline* and *control plane apps*

- There's no routing (BGP), trunking (LACP), etc. by default
- Pairs with *fabric.p4*, *ONOS*, and *Trellis* for a complete networking stack

Stratum provides rich, programmatic interfaces

- **P4**: specify the data plane pipeline
 - Works with fixed-function and programmable targets
- **P4Runtime**: program the data plane at runtime
 - Works on the box (distributed) or off the box (SDN)
- **OpenConfig**: model ASIC and platform configuration
 - Provided, but extensible
- **gNMI**: configure and monitor the data plane
 - Enables subscriptions and streaming telemetry





Stratum Release Highlights



Release Process and Cadence



- ✓ 20.06
- ✓ 20.09
- ✓ 20.12
- ✓ 21.03
- ✓ 21.06 (latest)
- 21.09 (planned)

Focus of today's talk

- “Live at Head”
 - Be fast moving and adaptive
 - Development happens on **main**
- Quarterly releases
 - Primarily, tagged for deployments
 - Release name format: **year.month**
 - Release tag format: **YYYY-MM-DD**
 - Automated script takes care of building and uploading artifacts

Release Notes and Artifacts



Release notes and binaries can be found on the GitHub release page:

<https://github.com/stratum/stratum/releases>

- Debian packages are included in the “Assets” section of the GitHub release page
- Docker images are uploaded to DockerHub

2021-06-30

Latest release

2021-06-30 99375ca







Compare


Verified

bocon13 released this 24 days ago · 20 commits to main since this release

We are pleased to announce the fifth Stratum release: 2021-06-30 (or 21.06 for short)

Assets 15

 stratum-bcm-21.06-openssa-amd64.deb	25.3 MB
 stratum-bcm-21.06-sdkt-amd64.deb	7.46 MB
 stratum-bf-21.06-9.2.0-amd64.deb	14.3 MB
 stratum-bf-21.06-9.3.1-amd64.deb	14.5 MB
 stratum-bf-21.06-9.3.2-amd64.deb	14.5 MB
 stratum-bf-21.06-9.4.0-amd64.deb	14.6 MB
 stratum-bf-21.06-9.5.0-amd64.deb	14.8 MB
 stratum-bf-21.06-9.2.0-amd64.deb	14.3 MB
 stratum-bf-21.06-9.3.1-amd64.deb	14.5 MB
 stratum-bf-21.06-9.3.2-amd64.deb	14.5 MB
 stratum-bf-21.06-9.4.0-amd64.deb	14.5 MB
 stratum-bf-21.06-9.5.0-amd64.deb	14.8 MB
 stratum_bmv2_deb.deb	7.18 MB
 Source code (zip)	
 Source code (tar.gz)	

 Search for great content (e.g., mysql) Explore Repositories Organizations Help

stratumproject > Repositories > stratum-bf-rt > 21.06-9.5.0



stratumproject/stratum-bf-rt:21.06-9.5.0

DIGEST: sha256:9288d3d9675b3b7bc9d653d8491c9966d8dc47f5e87809996d636f1ae34f170a

OS/ARCH

linux/amd64

COMPRESSED SIZE

63.49 MB

LAST PUSHED

18 days ago by onbuilder



API Enhancements



- TCP Port 28000 is no longer listening by default
 - Use IANA standard ports: **9339** for gNMI and **9559** for P4Runtime
 - Can be overridden using the **-external_stratum_urls** flag
 - Default: **0.0.0.0:9339,0.0.0.0:9559**
- P4Runtime version is now 1.3.0
 - Biggest change is move to canonical byte strings
- gNMI: several fixes (*see Release Notes for details*)
 - Log level can now be update at runtime:
/system/logging/console/config/severity

P4Runtime Canonical Binary Strings



Previously, Stratum used full width binary strings for P4Runtime integer values.

- `bit<32>`: `\x00\x00\x00\x01`
- `bit<16>`: `\x00\x02`
- This was required by several targets, including BMv2 and Tofino.

Now, Stratum uses canonical binary strings as recommended by the [P4Runtime specification](#). Padding (leading zeros) is stripped in this format.

- `bit<32>`: `\x01`
- `bit<16>`: `\x02`
- Stratum takes care of padding the canonical strings into the full width strings that are required by the SDK.

Either full width or canonical strings can be written to Stratum, but reads will always return canonical strings.

- **Use canonical strings if you want to ensure read-write symmetry**

Changing the Log Level at Runtime with gNMI



Log level is initialized via [glog](#) flags

- 2 knobs: **-minloglevel** for severity and **-v** for verbosity
- Default: INFO (0) and Verbosity 0

Log level can be updated at runtime using gNMI set on `/system/logging/console/config/severity`

Example: Getting and Setting the log level

```
docker run --rm --network=host \  
  stratumproject/stratum-tools gnmi_cli \  
  --grpc_addr=localhost:9339 \  
  get \  
  '/system/logging/console/config/severity'
```

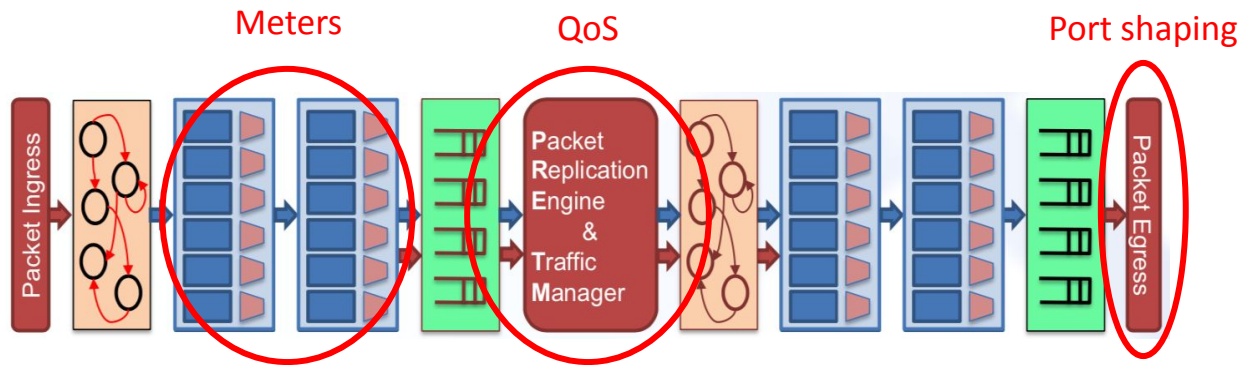
```
docker run --rm --network=host \  
  stratumproject/stratum-tools gnmi_cli \  
  --grpc_addr=localhost:9339 \  
  set --string_val="DEBUG" \  
  '/system/logging/console/config/severity'
```

gNMI Log Level	glog Severity	glog Verbosity
CRITICAL	FATAL (3)	0
ERROR	ERROR (2)	0
WARNING	WARNING (1)	0
NOTICE	INFO (0)	0
INFORMATIONAL	INFO (0)	1
DEBUG	INFO (0)	2

Tofino Enhancements



Taking full advantage of the Tofino ASIC capabilities beyond match-action tables

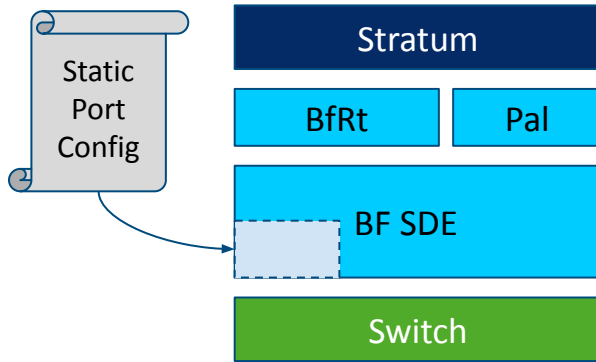


- P4RT Meters
 - Dynamically color-mark flows according to byte or packet limits
- Expose traffic manager (TM) APIs in ChassisConfig
 - Direct, unmodeled access to application buffer pools, PPGs and queues
 - Might get refined into a model, e.g. OpenConfig, over time
 - Allows implementation of advanced Quality of Service (QoS) rules
- Egress port shaping
 - Per port limits to prevent overloading downstream links

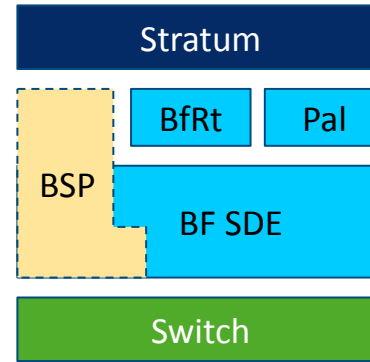
Tofino Enhancements



Stratum now includes BSP support on select switches



or



- Configuration provided out-of-band
- Fixed after startup (e.g. transceiver types and SerDes values)
- Platform and model (SKU) specific

- BSP discovers and dynamically sets configuration depending on available hardware
- Out-of-the-box support for optical transceivers
- Exposes advanced platform features (LEDs, SerDes, ...)
- Independent from ONLP

Debugging Enhancements



Continue to increase Stratum's useability for non-developers

- All released Docker images include labels to trace their origin
- Binaries are stamped with build-time information
- Easy remote access script to the Tofino bfshell
- Many auxiliary tools collected and released in one convenient package
 - Contains: **gnmi_cli**, **bf_pipeline_builder**, **stratum_replay**, and more
 - Docker image: [stratumproject/stratum-tools](https://stratumproject.org/stratum-tools)
 - Debian package: [stratum-tools.deb](https://stratumproject.org/stratum-tools.deb)

Debugging Enhancements



[Live Demo]



Stratum Roadmap

2021 Technical Roadmap for Stratum



1. Support production users
 - QoS model [**initial / experimental support**]
 - Tofino features (scheduler, traffic manager) [**experimental support**]
 - External API enhancements (P4Runtime, gNMI) [**enhancements ongoing**]
 - Testing (pre-merge hardware validation, CI system)
2. Portability
 - Standardized config (move from Chassis Config to OpenConfig)
 - Platform API integration / plugins (focus on transceivers and ASIC) [**first step with BSP**]
 - Run / test on other switch OSES
3. Usability
 - Documentation **and tooling** [**improved**]
 - Tofino model usability (Mininet integration, P4 program linking) [**in progress**]
 - Build and release automation [**improved**]

Run Stratum Anywhere



- Reducing dependency on ONLP gives users more choices
 - Ubuntu, Debian, Network Grade Linux (NGL), SONiC, etc.
- Stratum packages already support multiple kernels
 - Adding support for more is easy and automated
- BSP support improves portability for Tofino platforms
 - Supported for the Wedge100BF platforms as of 21.06
 - Vendors can build Stratum packages for their platforms
 - Exploring ways to package BSPs for multiple platforms into a universal Stratum package (like we do for kernel modules)



Switch OS Landscape

Bring Stratum Capabilities to Other Switch OSES

PINS: P4 Integrated Network Stack

- Google, Intel, and ONF are working with the community to model SAI in P4 and bring P4Runtime to SONiC
- Supports a hybrid control plane
- Provides an incremental, opt-in path towards SDN
- Allows for SAI extensions to be modeled in P4
- Design docs available soon; MVP will be part of SONiC's November 2021 release

Overlapping Community Projects

- P4Runtime Server (looking to provide a standalone common library)
- Testing and Validation

Bring Stratum Capabilities to Other Switch OSES

P4 OvS

- Intel, Orange, and ONF are working with the community to bring P4 capabilities to OvS
- P4Runtime API with translation to and from OpenFlow
- P4 used to precisely model the pipeline
- Support for P4-enabled targets: eBPF, DPDK, Smart NIC, FGPA, Tofino
- Planning a PoC on DPDK for later this year

Overlapping Community Projects

- P4Runtime Server (starting with Stratum's)
- P4 to Intel Table API translation (sharing code with **stratum_bfrt**)

P4 Switch OS Landscape



Stratum: bring your own pipeline and control plane

- Great for pure-play SDN networks and highly customized pipelines
- Extremely lightweight, high performance, and low footprint

PINS (SONiC): hybrid control plane with path to SAI extensibility

- Great for traditional network users that want some SDN capabilities
- Seamless transition for SONiC users

P4 OvS: optimize OvS by running on P4-enabled targets

- Great for OvS users with OpenFlow control planes and OvS DB config
- Designed for the server with a wide range of supported targets (CPU to switching ASIC)

P4 Switch OS Landscape



- P4 is the common language for pipeline definition and extension
- Common SDN-ready interfaces: P4Runtime & gNMI
- Shared code, libraries, and test cases
 - P4 compiler, P4Runtime Service, P4 to Table API
 - Tools: test frameworks, ptf, P4Runtime Shell, etc.
- Lots of community overlap (companies and individuals)
- Improved collaboration and code reuse benefits everyone
 - Cross-pollination of ideas
 - Reduced development time and maintenance burden

Getting Involved



How do I get involved with Stratum?

- Send an [email to stratum-dev](#) or a [Slack message to #stratum-dev](#)
- Join the [weekly technical steering team \(TST\) meeting on Thursdays at 12:30pm Pacific](#)
- Send a pull request with your changes to the [Stratum Github repo](#)
- Try the [NG-SDN tutorial](#)
- Add BSP support for your Tofino platform and let us know!

PINS is currently an ONF Member's Only project and will be opening later this year.

P4 OvS is currently incubating with a small engineering team.





Thank You

Follow Up Links:

<https://opennetworking.org/stratum/>

<https://github.com/stratum/stratum>

Links



Stratum

<https://opennetworking.org/stratum/>

<https://github.com/stratum/stratum>

PINS

<https://opennetworking.org/pins/>

[PINS – Extending SDN and P4 Programmability](#) (Mythil Raman & Konstantin Weitz, Google) - [slides](#), [video](#)

P4 OvS

[Converting OpenFlow to P4](#) (James Choi, Intel) - [PPTX](#), [video](#)

[p4proto: Cooking OVS with P4 Spice!](#) (Namrata Limaye and Deb Chatterjee, Intel) - [PPTX](#), [video](#)

[vswitch.p4: Why OVS Needs P4](#) (Dan Daly, Intel) - [PPTX](#), [video](#)