

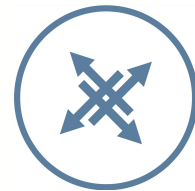
Arista 7130 update: ***Beyond Nanoseconds***

David Snowdon, November 2021
daves@arista.com

EOS on 7130

→ The world's best network operating system

Arista EOS is generally available

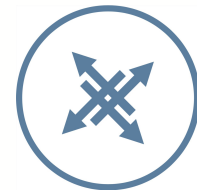


- Two GA releases so far.
 - Unified release with other platforms this quarter.
 - Ongoing feature enhancements
- MetaWatch beta release on EOS
 - Initially with a limited feature set.
 - GA release shortly.

SwitchApp

→ Our lowest latency L2/3 switch

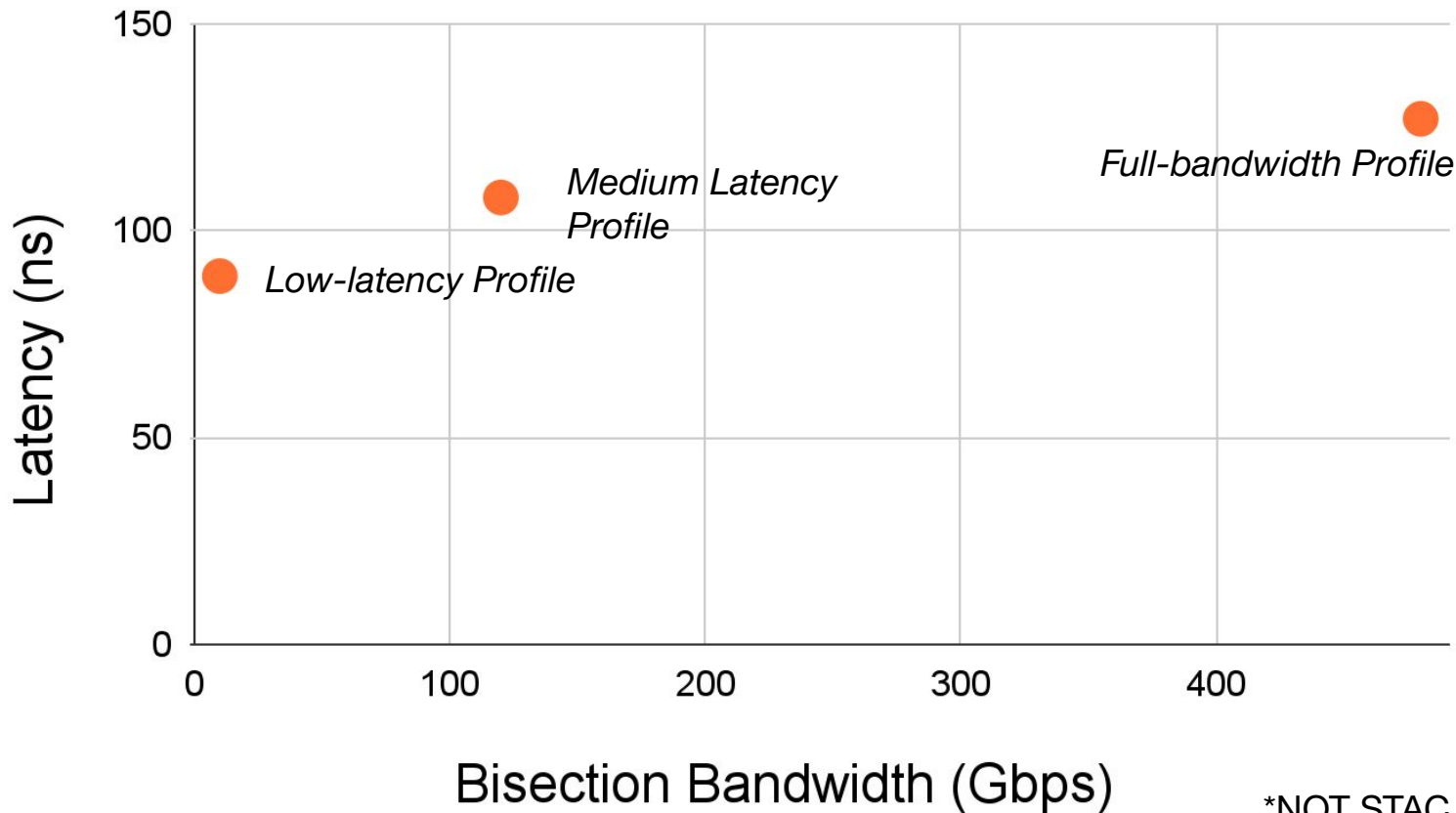
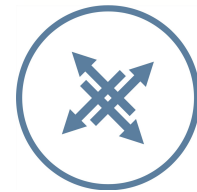
Arista SwitchApp Updates



- SwitchApp is Arista's ultra-low latency L2/3 Switch
 - Based on an FPGA with an Arista-designed pipeline.
- Two GA releases
 - New medium bandwidth profile (120G at ~108 ns)
 - Reduced latency (89 ns for 10G, 128 ns for 480G)
 - Added features: In-band management, PVST
- Upcoming release with L3 support
 - Fully integrated with EOS as a new switching ASIC
 - Static routes, BGP, PIM
 - ACLs and NAT to come.

*NOT STAC BENCHMARKS

Arista SwitchApp Profiles



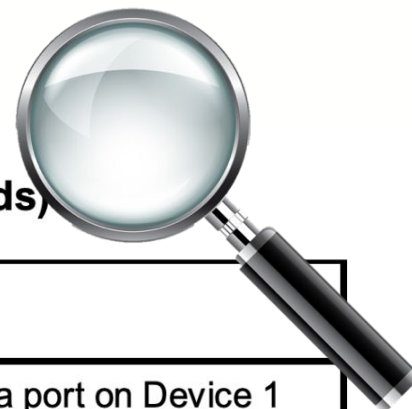
*NOT STAC BENCHMARKS

MetaWatch and White Rabbit

→ Tighter time synchronization, more convenience.

Arista 7130 STAC-TS results from last year

Port synchronization across two devices (picoseconds)



	Skew	Random error	Notes
STAC-TS.PSE2.TOTAL	-118	+/- 681	Worst case port sync between a port on Device 1 and a port on Device 2, based on the port pair with the largest total error magnitude (skew + random error).
STAC-TS.PSE2.RAND.WORST	n/a	+/- 696	Worst case random port-sync error between a port on Device 1 and a port on Device 2 (i.e., error that cannot be calibrated out).
STAC-TS.PSE2.RAND.BEST	n/a	+/- 399	Best case random port-sync error between a port on Device 1 and a port on Device 2 (i.e., error that cannot be calibrated out).

Results audited by STAC

MetaWatch and White Rabbit

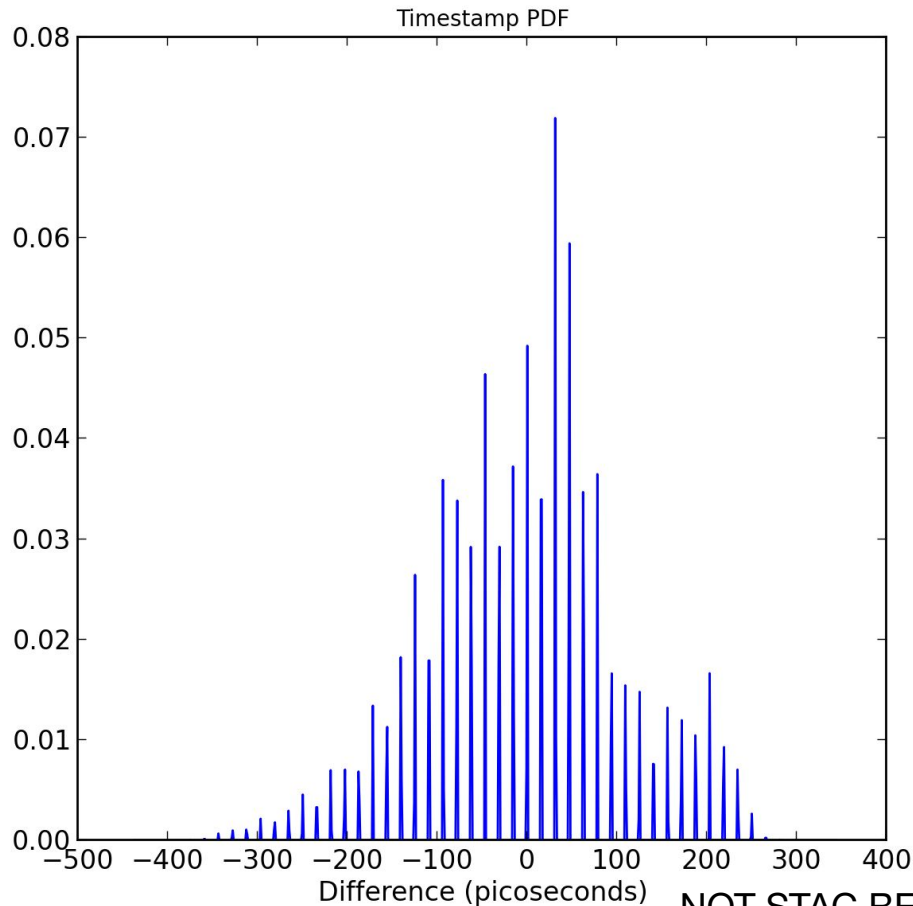
- White Rabbit is *really* accurate PTP over Synchronous Ethernet
 - Network connection over 1GbE
 - A shared frequency source across all devices.
- Arista have added the Seven Solutions HATI IP core
 - Only works with the Seven Solutions boundary clock -- WR Z16
- MetaWatch can synchronize to White Rabbit
 - Runs on 7130-48LB(S) devices (*note: no support on 7130-48L(S)*)
- Pre-release. An alpha is coming.



MetaWatch with PPS

```
*****
Iteration   :      0
Timesource  :      pps
*****
Median     :    47.000
Mid-range  :     0.500
Half-range  :   390.500
Std dev    :   107.219
Mean       :    44.173
*****
```

Half Range: **390 ps**

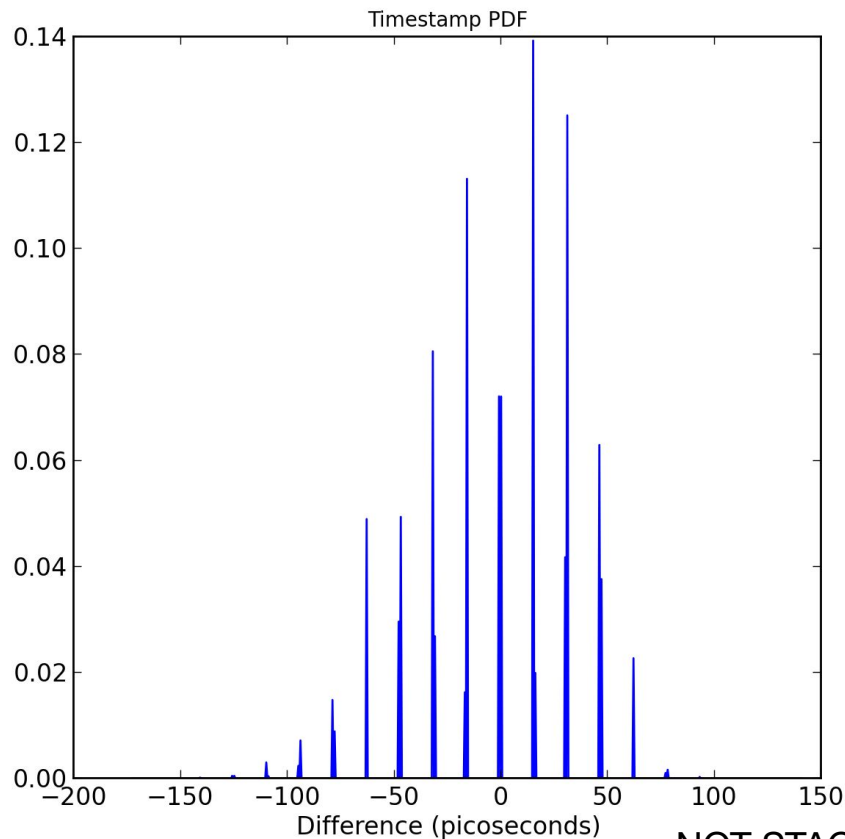


NOT STAC BENCHMARKS

MetaWatch with White Rabbit

```
*****
Iteration   :      0
Timesource  : whiterabbit
*****
Median      :   -62.000
Mid-range   :    78.000
Half-range  :   157.000
Std dev     :    36.391
Mean        :   -63.846
*****
```

Half Range: **157 ps**



NOT STAC BENCHMARKS

EOS API for FPGA Apps

→ Build applications that work for you

Arista FDK Released for EOS

Gives developers the ability to:

- Leverage the EOS infrastructure to build applications;
 - CLI, JSON API, Influx telemetry
 - State-based config and status management.
- Use Arista-built FPGA logic components;
 - These are core components we use to build Arista apps.
- Get an enterprise-grade hardware and software platform.
 - Proven, maintained, hardened
 - Manage devices and apps using CloudVision, Ansible, OpenConfig, ZTP.
 - Enterprise grade permissions management -- AAA.

You already rely on Arista for the network, now build on the same platform.

Examples -- Now open source!

- Built to work
 - ``make`` builds a functional and installable app.
 - Each example adds command to the Arista EOS CLI, JSON API.
 - Examples are subjected to the same type of tests as Arista apps.
- Arista FDK examples are now BSD 3-clause licensed.
 - Please copy them!
- Built to be helpful
 - If you don't understand, let us know...
 - Examples demonstrate FPGA and software libraries.
- Arista focus is core functionality supporting higher layers.
 - Like Enyx and Xilinx frameworks.

Name

```
> devkit
> doc
v examples
  > cliexample
  > clkgen
  > ddr4
  > macphy
  > macphy25g
  > mmp
  > muxcore
  > null
  > pcie
  > telemexample
  > tscore
    Makefile
    LICENSE.md
    README.md
> ipcores
> resources
> src
  arista_src.md5
  LICENSE.md
  README.md
```

An infrastructure example

- Adding an EOS CLI using CLI extensions gives your app:
 - a JSON RPC API -- Build your Web UI
 - Configuration management (via startup config)
 - Orchestration support (via Ansible, CloudVision, or other)

```
dmb226# — ssh us142 — 42x8
dmb226#show telemexample status
Enabled: Yes
Running: Yes
  Setting  Value
-----
  period   4.3
dmb226#
```

show command

```
dmb226# — ssh us142 — 41x10
dmb226#show telemexample status | json
{
  "running": true,
  "settings": {
    "period": 4.3
  },
  "records_sent": 0,
  "enabled": true
}
dmb226#
```

json output via CLI

```
dmb226# — ssh us142 — 44x6
dmb226(config-telemexample)#show active
telemexample
  period 4.3
  no disabled
dmb226(config-telemexample)#
dmb226#
```

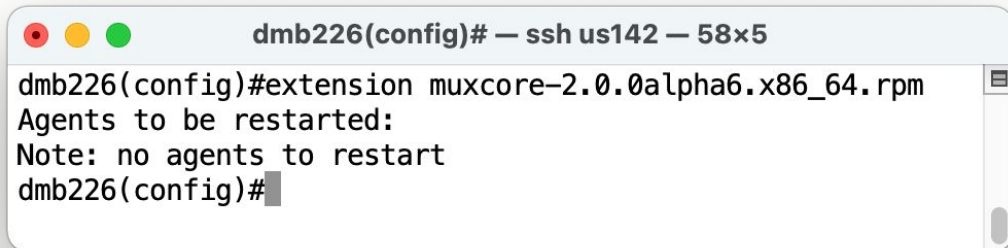
viewing the active config

Underpinned by Linux

tree /opt/apps/muxcore/ (Abbrev)

```
|-- drivers
| |-- macphy
| |   |-- driver
| |   |   |-- macphy.py
| |-- mux_ipcore
| |   |-- driver
| |   |   |-- muxcore.py
|-- eos
| |-- MuxCoreExample.yaml
| |-- MuxCoreExampleCli.py
| |-- MuxCoreExampleDaemon.py
|-- fpga
| |-- muxcore-eh_central.bit
| |-- muxcore-l.bit
| |-- muxcore-lb2.bit
...
|-- www
| |-- index.html
|-- main.js.gz
```

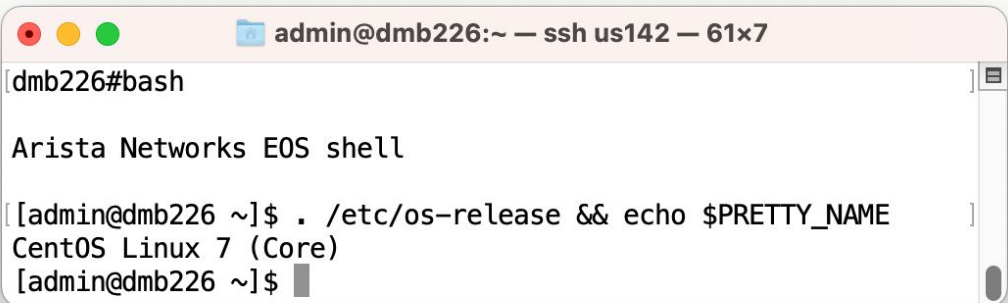
Anatomy of an installed app



A terminal window titled "dmb226(config)# — ssh us142 — 58x5". The user enters the command "extension muxcore-2.0.0alpha6.x86_64.rpm". The output shows "Agents to be restarted:" and "Note: no agents to restart". The prompt returns to "dmb226(config)#".

```
dmb226(config)# — ssh us142 — 58x5
dmb226(config)#extension muxcore-2.0.0alpha6.x86_64.rpm
Agents to be restarted:
Note: no agents to restart
dmb226(config)#
```

Installing an application



A terminal window titled "admin@dmb226:~ — ssh us142 — 61x7". The user enters "bash". The prompt changes to "dmb226#". The user enters "Arista Networks EOS shell". The prompt changes to "[admin@dmb226 ~]\$". The user enters ". /etc/os-release && echo \$PRETTY_NAME". The output is "CentOS Linux 7 (Core)". The prompt returns to "[admin@dmb226 ~]\$".

```
admin@dmb226:~ — ssh us142 — 61x7
dmb226#bash
Arista Networks EOS shell
[admin@dmb226 ~]$ . /etc/os-release && echo $PRETTY_NAME
CentOS Linux 7 (Core)
[admin@dmb226 ~]$
```

EOS is currently based on CentOS 7

Thank you!

To find out more, you could:

- Watch the video of Darrin and Joris' excellent sessions about development with the Arista FDK and ways to use 7130.
- Download these slides.
- Get in touch directly -- daves@arista.com



Thank You

www.arista.com