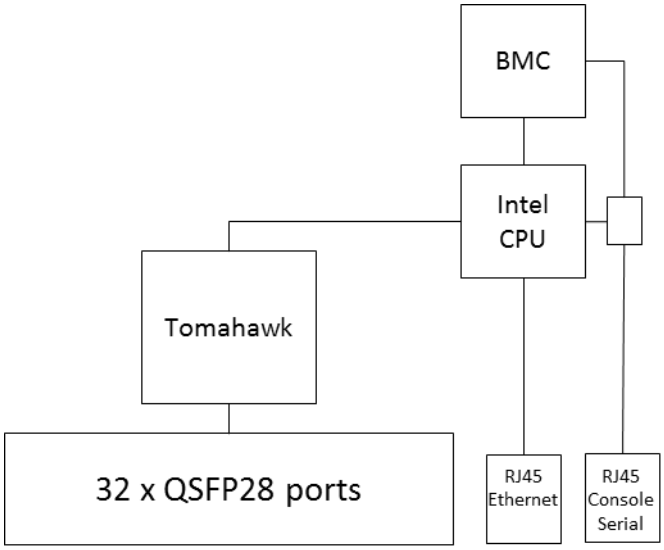
Platina PSW-3001-32C Trial Configuration Guide
Last Update 6/13/2017

Quick Start Guide

Please refer to the quick start guide included with the unit for hardware package/unpackage instructions, MAC addresses, port labeling, field replaceable unit labeling, and other general instructions.

General Architecture

The PSW-3001-32C is a 32x100GE switch with hardware forwarding up to 3.2Tbps via the Broadcom Tomahawk ASIC. Built into the switch is a 4-core Intel Broadwell DE CPU with 16GB RAM and 128GB SSD. There is also a separate BMC ARM processor with 2GB of DRAM and 2GB of uSD. The high level block diagram is as follows:



BMC Processor

BMC processor controls the PSU, board voltage rails, fan speed, and other GPIO on the board. For the trial, there is nothing to configure on the BMC. Everything, including temperature and fan speed, is automated.

Intel Processor

After power up, the RJ45 console serial should default to Intel CPU console. The prompt is the Linux prompt. Everything needed for using, configuring, and monitoring the unit can be done from here.

Linux

Linux is the operating system for PSW-3001-32C. The unit is pre-loaded with stock Debian Jesse and kernel version 4.11.0. The admin account is "root"; password "platina". All standard Linux command and usage applies as is, including configuration of Ethernet interfaces, installation of new apps, etc.

By default eth0 is the RJ45 Management Ethernet port on the front panel.

Note: In this trial version, Platina has made the following kernel modifications.
- uio_pci_dma: for accessing PCI devices from userspace
These modifications will be removed eventually in later revision of the SW.

Platina GOES app

Platina software, GOES, is a user space application that runs in Linux. A copy of the GOES binary is at /root/goes-platina-mk1

GOES is pre-installed on each switch. To manually install GOES (i.e. SW upgrade/downgrade, re-install) enter the following command:

```
sudo /root/goes-platina-mk1 install
```

The 'install' command will install GOES at /usr/bin/goes, by default.

Once installed, GOES will startup automatically on reboot going forward.

You can verify GOES is running properly by entering

```
goes status
```

and look for the following:

```
GOES status
====================
  PCI              - OK
  Kernel module    - OK
  Check daemons:
    goes-daemons   - OK
    redisd         - OK
    i2cd           - OK
    uptimed        - OK
    vnetd          - OK
    qsfp           - OK
  Check Redis      - OK
  Check vnet       - OK
```

To uninstall goes, enter:

```
sudo goes uninstall
```

To stop goes without doing a full uninstall enter:

```
sudo goes stop
```

To start up goes again, enter:

```
sudo goes start
```

Each stop/start will reset all ASIC configuration/memory and reinitialize ASIC.

GOES is an open source project developed by Platina. To see the source code, visit

https://github.com/platinasystems/go

There is an addition private repository needed to compile the source. To inquire if you meet the license requirement to access the private repository, email support@platinasystems.com.

Admin Privilege

Running goes start/stop/install/uninstall require superuser privilege in Linux. The Quick Start guide includes default password for root access preconfigured on the unit.

For other 'goes…' commands, it is sufficient to be just part of the 'adm' group to be able to execute the command. A new user account should be added to the 'adm' group if goes command access is required, otherwise goes commands will need to be invoked with sudo.

Redis Database/Interface

GOES includes a Redis server daemon. Any configuration and stats not normally associated with Linux can be found in this Redis database. The GOES Redis server is a standard Redis server listening on port 6379 of the loopback interface and eth0. The database can be accessed remotely anywhere via standard Redis-client and Redis-cli using the IP address of eth0.

To invoke Redis commands directly from the Linux CLI, precede the Redis command with "goes", otherwise standard Redis command format should be used from a Redis client. In this guide, the command examples will assume the user is invoking from the switch's Linux CLI.

To see a list of all parameters in this Redis database, enter:

From Linux CLI: `goes hgetall platina`

From a Redis client: `hgetall platina`

"platina" is the hash for which all of the unit's configurations and stats are stored under. The field/value associated with platina appear in the output as field:value.

You can view just a particular set of fields using hget. For example to get all configuration/stats associated with front panel port eth-1-1, enter:

```
goes hget platina eth-1-1
```

Redis database is updated every 5 seconds by default for stats counters. The update interval for the switch ASIC stats can be changed. For example, to change the interval to 1 second enter:

```
goes hset platina vnet.pollInterval 1
```

Included in the GOES Redis interface is a convenience command hdelta. hdelta returns only the non-zero differences from the last Redis update and includes rate calculation if applicable. hdelta runs continuously until stopped by ctrl-c. hdelta is not a standard Redis command and must be invoked with the following command via CLI.

```
goes hdelta
```

Since the commands above are invoked at the standard Linux command line prompt, 'grep' or other Linux tools can be used to further filter the output.

QSFP28 Ports

Once GOES is installed and running, all front panel ports will show up as normal eth interfaces in Linux. By default they are eth-1-1, eth-2-1, … , eth-32-1. The format is eth-<port_number>-<sub-port_number> where port_number corresponds to the 32 front panel ports and sub-port_number corresponds to optionally configured breakout ports within each port.

All 32 QSFP28 eth interfaces can be configured via Linux using standard Linux methods, e.g. ifconfig, ip addr, /etc/network/interfaces, etc.

Interface configurations not available in Linux, such as interface speed and media type, can be done via the Redis database.

Set Media Type and Speed

To set the media type and speed on eth-1-1 to copper (i.e. DAC cable) and 100g fixed speed for example, enter

```
goes hset platina vnet.eth-1-1.media copper
goes hset platina vnet.eth-1-1.speed 100g
```

To set the speed to autoneg enter

```
goes hset platina vnet.eth-1-1.speed auto
```

Note: In this trial version, media copper must be set before speed in order for link training and receive equalization to work optimally.

Note: In this trial version, the only speed options supported are auto, 100g, and 40g.

The settings do not persist across goes stop/start or reboot. To configure permanently, add the configuration in /etc/goes/start as described below.

Persistent Configuration

Each time GOES starts up, it will read the configuration file /etc/goes/start. Each line in the configuration file has the same syntax as a Redis command. Linux commands can also be included in /etc/goes/start by prefixing the command with "!"

Example /etc/goes/start file:

```
#!/usr/bin/goes
hset -q platina vnet.eth-1-1.media copper
hset -q platina vnet.eth-1-1.speed auto
hset -q platina vnet.eth-2-1.media copper
hset -q platina vnet.eth-2-1.speed auto
! ifup --allow vnet -a
```

In this example, hset does the same thing as if done at the Linux prompt prefixed by “goes”. The “-q” is an optional parameter to quiet informational CLI output prints from each command. In the example above, “ifup” is a standard Linux command and works in conjunction with the following /etc/network/interfaces file:

```
auto eth0
iface eth0 inet static
  address 192.168.101.127
  netmask 255.255.255.0
  gateway 192.168.101.2
  dns-nameservers 8.8.8.8 8.8.4.4

allow-vnet eth-1-1
iface eth-1-1 inet static
  address 10.0.1.7
  netmask 255.255.255.0

allow-vnet eth-2-1
iface eth-2-1 inet static
  address 10.0.2.7
  netmask 255.255.255.0
```

There is also a /etc/goes/stop configuration file that gets executed when GOES stop (or gets uninstalled).

Example /etc/goes/stop file:

```
#!/usr/bin/goes
! ifdown --allow vnet -a
```

BGP

The trial unit does not come pre-installed with BGP. Any BGP protocol stack can be installed directly onto Linux as if installing on a server. The GOES daemon will handle any translation between the Linux kernel and ASIC. All FIB/RIB can be obtained directly from Linux or the BGP stack.

To get the FIB from the ASIC for verification, enter:

```
goes vnet show ip fib
```

List of Known Issues

For list of known issues, please visit

https://github.com/platinasystems/go/issues

Support

Any question or to report new issues, please email

support@platinasystems.com

Appendix 1: Redis Fields Guide

The examples in the appendix show standard Redis commands. When directly on the switch's Linux CLI, add "goes" in front of the Redis command.

From Redis-client:

```
hgetall platina
```

From switch Linux prompt:

```
goes hgetall platina
```

To get multiple field names that contain a specific string, use hget platina <string>, for example:

```
goes hget platina eth-1-1
```

When using redis-cli, it is recommended to connect using the --raw parameter so that newline characters between the multiple fields are parsed properly. For example:

```
redis-cli --raw -h  <ipv4_address> hget platina eth-1-1
```

Fields that can be set in Platina's Redis:

Most of the fields in the Platina Redis are read-only, but some can be set. If a set is successful, Redis will return an integer 1. Otherwise Redis will return an error message.

Media Type

Each port can be configured as copper or fiber mode. Copper mode is for QSFP28 media such as direct attach cable (DAC) where the switch ASIC serdes is driving the line directly. Fiber mode is for all other QSFP28 optical pluggable modules where ASIC serdes interacts typically with a CDR at the local QSFP28 module. 100GE autoneg function and link training is only available in copper mode.

Example:

```
hset platina vnet.eth-1-1.media copper
```

```
hget platina vnet.eth-1-1.media
copper
```

Speed

Each QSFP port supports 1GE, 10GE, 20GE, 40GE, 50GE, and 100GE speeds, either in single port mode or breakout port mode. In the trial version, only 40GE and 100GE single port modes are supported. If a speed is specified, the port will be fixed at that speed. If set to "auto" (applicable only if port is in copper media mode), the port will negotiate with neighbor switch to establish the speed.

Example:

```
hset platina eth-1-1.speed auto
```

```
hget platina eth-1-1.speed
auto

hset platina eth-1-1.speed 100g

hget platina eth-1-1.speed
100g
```

Stats Counter Update Interval

Stats counters such as transmit/receive packet counters, packet drops, etc. are maintained real time in hardware, but updated to Redis data at a fixed interval. This interval defaults to 5 seconds. This interval can be configured, the minimum being 1 second. Other asynchronous event, such as line up/down, are updated to Redis real time as the event occur.

To change the Redis counter update interval.

Example:

```
hset platina vnet.pollInterval 1

hget platina vnet.pollInterval 1
vnet.pollInterval: 1
```

Read-only Redis Fields

These fields are read only. Attempts to set them will return error messages.

Packages

Packages shows the software version numbers of GOES. These version numbers match the github commit/versions in https://github.com/platinasystems from which the GOES binary is built.

Example

```
hget platina packages | grep versions
  version: bef5c09d9b9d5203c8c9de5688183709bf1b77f2
  version: d0e64c647b86e234434f7bdf55ba52df777bb261
  version: a7405309d5953fa8e08096613db1e27400fc4a8a
```

Physical link state (as reported by switch ASIC)

vnet.[interface].link indicates the physical link state as reported by the switch ASIC. The value is "true" if link is up, "false" if link is down.

Example:

```
goes hget platina vnet.eth-3-1.link
false
```

Admin state of the link (as reported by Linux)

vnet.[interface].admin indicates the admin state as reported by the Linux interface. The value is "true" if link is up, "false" if link is down. This is the same link state as reported by Linux commands such as "ip link show [interface]". Admin down of an interface from Linux will automatically force link down on the interface at the ASIC level.

Example:

```
goes platina hget vnet.eth-3-1.admin
true
```

Linux Packet/Byte counters

Deprecated from redis

The Linux interface counter (e.g. via ip link or ifconfig) counts packets that have been sent/receive from/to the ASIC to/from the CPU only. Packets that are locally received and forwarded in ASIC are not included in the Linux interface counter. To get the ASIC level interface counters, use vnet.[interface].port redis commands instead.

```
ip -s link show eth-1-1
2612: eth-1-1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 4352 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 5000
    link/ether 02:46:8a:00:03:aa brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    738        7        0       0       0       0
```

Port Counters

vnet.[interface].port… are counters that reflect the interface counters as reported by the switch ASIC. These counters count the number of packets/bytes that are coming in or going out of the switch ASIC's, as well as counters for various types of recognized packet types, for the corresponding front panel ports.

Example: (assume launching from switches local prompt; otherwise hget each field 1 at a time)

```
goes hget platina vnet.eth-2-1.port
vnet.eth-2-1.port_rx_1024_to_1518_byte_packets: 0
vnet.eth-2-1.port_rx_128_to_255_byte_packets: 161930000680
vnet.eth-2-1.port_rx_1519_to_1522_byte_vlan_packets: 0
vnet.eth-2-1.port_rx_1519_to_2047_byte_packets: 0
vnet.eth-2-1.port_rx_1tag_vlan_packets: 0
vnet.eth-2-1.port_rx_2048_to_4096_byte_packets: 0
vnet.eth-2-1.port_rx_256_to_511_byte_packets: 0
vnet.eth-2-1.port_rx_2tag_vlan_packets: 0
vnet.eth-2-1.port_rx_4096_to_9216_byte_packets: 0
vnet.eth-2-1.port_rx_512_to_1023_byte_packets: 0
vnet.eth-2-1.port_rx_64_byte_packets: 0
vnet.eth-2-1.port_rx_65_to_127_byte_packets: 0
vnet.eth-2-1.port_rx_802.3_length_error_packets: 0
vnet.eth-2-1.port_rx_9217_to_16383_byte_packets: 0
vnet.eth-2-1.port_rx_alignment_error_packets: 0
vnet.eth-2-1.port_rx_broadcast_packets: 0
vnet.eth-2-1.port_rx_bytes: 26880380211152
vnet.eth-2-1.port_rx_code_error_packets: 0
vnet.eth-2-1.port_rx_control_packets: 0
vnet.eth-2-1.port_rx_crc_error_packets: 0
vnet.eth-2-1.port_rx_eee_lpi_duration: 0
```

```
vnet.eth-2-1.port_rx_eee_lpi_events: 0
vnet.eth-2-1.port_rx_false_carrier_events: 0
vnet.eth-2-1.port_rx_flow_control_packets: 0
vnet.eth-2-1.port_rx_fragment_packets: 0
vnet.eth-2-1.port_rx_good_packets: 161930001015
vnet.eth-2-1.port_rx_jabber_packets: 0
vnet.eth-2-1.port_rx_mac_sec_crc_matched_packets: 0
vnet.eth-2-1.port_rx_mtu_check_error_packets: 0
vnet.eth-2-1.port_rx_multicast_packets: 0
vnet.eth-2-1.port_rx_oversize_packets: 0
vnet.eth-2-1.port_rx_packets: 161930000775
vnet.eth-2-1.port_rx_pfc_packets: 0
vnet.eth-2-1.port_rx_pfc_priority_0: 0
vnet.eth-2-1.port_rx_pfc_priority_1: 0
vnet.eth-2-1.port_rx_pfc_priority_2: 0
vnet.eth-2-1.port_rx_pfc_priority_3: 0
vnet.eth-2-1.port_rx_pfc_priority_4: 0
vnet.eth-2-1.port_rx_pfc_priority_5: 0
vnet.eth-2-1.port_rx_pfc_priority_6: 0
vnet.eth-2-1.port_rx_pfc_priority_7: 0
vnet.eth-2-1.port_rx_promiscuous_packets: 161930000975
vnet.eth-2-1.port_rx_runt_bytes: 0
vnet.eth-2-1.port_rx_runt_packets: 0
vnet.eth-2-1.port_rx_src_address_not_unicast_packets: 0
vnet.eth-2-1.port_rx_truncated_packets: 0
vnet.eth-2-1.port_rx_undersize_packets: 0
vnet.eth-2-1.port_rx_unicast_packets: 161930000785
vnet.eth-2-1.port_rx_unsupported_dst_address_control_packets: 0
vnet.eth-2-1.port_rx_unsupported_opcode_control_packets: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_0: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_1: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_2: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_3: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_4: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_5: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_6: 0
vnet.eth-2-1.port_rx_xon_to_xoff_priority_7: 0
vnet.eth-2-1.port_tx_1024_to_1518_byte_packets: 0
vnet.eth-2-1.port_tx_128_to_255_byte_packets: 161872616443
vnet.eth-2-1.port_tx_1519_to_1522_byte_vlan_packets: 0
vnet.eth-2-1.port_tx_1519_to_2047_byte_packets: 0
vnet.eth-2-1.port_tx_1tag_vlan_packets: 0
vnet.eth-2-1.port_tx_2048_to_4096_byte_packets: 0
vnet.eth-2-1.port_tx_256_to_511_byte_packets: 0
vnet.eth-2-1.port_tx_2tag_vlan_packets: 0
vnet.eth-2-1.port_tx_4096_to_9216_byte_packets: 0
vnet.eth-2-1.port_tx_512_to_1023_byte_packets: 0
vnet.eth-2-1.port_tx_64_byte_packets: 0
vnet.eth-2-1.port_tx_65_to_127_byte_packets: 5
vnet.eth-2-1.port_tx_9217_to_16383_byte_packets: 0
vnet.eth-2-1.port_tx_broadcast_packets: 0
vnet.eth-2-1.port_tx_bytes: 26870854403676
vnet.eth-2-1.port_tx_control_packets: 0
vnet.eth-2-1.port_tx_eee_lpi_duration: 0
vnet.eth-2-1.port_tx_eee_lpi_events: 0
vnet.eth-2-1.port_tx_excessive_collision_packets: 0
vnet.eth-2-1.port_tx_fcs_errors: 0
vnet.eth-2-1.port_tx_fifo_underrun_packets: 0
vnet.eth-2-1.port_tx_flow_control_packets: 0
vnet.eth-2-1.port_tx_fragments: 0
vnet.eth-2-1.port_tx_good_packets: 161872616533
vnet.eth-2-1.port_tx_jabber_packets: 0
vnet.eth-2-1.port_tx_late_collision_packets: 0
vnet.eth-2-1.port_tx_multicast_packets: 7
vnet.eth-2-1.port_tx_multiple_collision_packets: 0
vnet.eth-2-1.port_tx_multiple_deferral_packets: 0
vnet.eth-2-1.port_tx_oversize: 0
vnet.eth-2-1.port_tx_packets: 161872616542
vnet.eth-2-1.port_tx_pfc_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_0_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_1_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_2_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_3_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_4_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_5_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_6_packets: 0
vnet.eth-2-1.port_tx_pfc_priority_7_packets: 0
vnet.eth-2-1.port_tx_runt_packets: 0
vnet.eth-2-1.port_tx_single_collision_packets: 0
vnet.eth-2-1.port_tx_single_deferral_packets: 0
vnet.eth-2-1.port_tx_system_error_packets: 0
vnet.eth-2-1.port_tx_total_collisions: 0
vnet.eth-2-1.port_tx_unicast_packets: 161872616546
```

Interface MMU Counters

vnet.[interface].mmu… are counters that reflect drops at the switch ASICs memory management unit (MMU). This is applicable to the Broadcom Tomahawk switch ASIC that employs the MMU to manage/switch traffic between 4 packet processing pipelines. The [interface] is the egress port, and the queues are queues associated with that port. For example, if "vnet.eth-2-1.mmu_multicast_tx_cos1_drop_packets" is non-zero, that means packets destined for eth-2-1 is dropping at the MMU's multicast, priority 1 queue.

Example:

```
goes hget platina vnet.eth-2-1.mmu
vnet.eth-2-1.mmu_multicast_tx_cos0_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos0_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos1_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos1_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos2_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos2_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos3_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos3_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos4_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos4_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos5_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos5_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos6_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos6_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_cos7_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_cos7_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_qm_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_qm_drop_packets: 0
vnet.eth-2-1.mmu_multicast_tx_sc_drop_bytes: 0
vnet.eth-2-1.mmu_multicast_tx_sc_drop_packets: 0
vnet.eth-2-1.mmu_rx_threshold_drop_bytes: 0
vnet.eth-2-1.mmu_rx_threshold_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos0_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos0_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos1_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos1_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos2_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos2_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos3_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos3_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos4_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos4_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos5_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos5_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos6_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos6_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_cos7_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_cos7_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_qm_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_qm_drop_packets: 0
vnet.eth-2-1.mmu_unicast_tx_sc_drop_bytes: 0
vnet.eth-2-1.mmu_unicast_tx_sc_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos0_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos1_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos2_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos3_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos4_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos5_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos6_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_cos7_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_qm_drop_packets: 0
vnet.eth-2-1.mmu_wred_queue_sc_drop_packets: 0
```

Interface Pipe Counters

vnet.[interface].rx_pipe… and vnet.[interface].tx_pipe… are counters that reflect counters at the switch ASICs packet processing pipelines. The rx_pipe… reflect counters for the ingress pipeline, and

tx_pipe… reflect counters for the egress pipeline. The [interface] is the egress port, and the queues are queues associated with that port. For example if “vnet.eth-2-1.tx_pipe_multicast_tx_cos0_drop_packets” is non-zero, that means packets destined for eth-2-1 is dropping at the egress pipeline’s multicast, priority 1 queue.

Example:

goes hget platina vnet.eth-2-1.rx_pipe

```
vnet.eth-2-1.rx_pipe_debug_3: 0
vnet.eth-2-1.rx_pipe_debug_4: 0
vnet.eth-2-1.rx_pipe_debug_5: 0
vnet.eth-2-1.rx_pipe_debug_6: 0
vnet.eth-2-1.rx_pipe_debug_7: 0
vnet.eth-2-1.rx_pipe_debug_8: 0
vnet.eth-2-1.rx_pipe_dst_discard_drops: 0
vnet.eth-2-1.rx_pipe_ecn_counter: 0
vnet.eth-2-1.rx_pipe_hi_gig_broadcast_packets: 0
vnet.eth-2-1.rx_pipe_hi_gig_control_packets: 0
vnet.eth-2-1.rx_pipe_hi_gig_l2_multicast_packets: 0
vnet.eth-2-1.rx_pipe_hi_gig_l3_multicast_packets: 0
vnet.eth-2-1.rx_pipe_hi_gig_unknown_hgi_packets: 0
vnet.eth-2-1.rx_pipe_hi_gig_unknown_opcode_packets: 0
vnet.eth-2-1.rx_pipe_ibp_discard_cbp_full_drops: 0
vnet.eth-2-1.rx_pipe_ip4_header_errors: 0
vnet.eth-2-1.rx_pipe_ip4_l3_drops: 0
vnet.eth-2-1.rx_pipe_ip4_l3_packets: 197000110884
vnet.eth-2-1.rx_pipe_ip4_routed_multicast_packets: 0
vnet.eth-2-1.rx_pipe_ip6_header_errors: 0
vnet.eth-2-1.rx_pipe_ip6_l3_drops: 0
vnet.eth-2-1.rx_pipe_ip6_l3_packets: 0
vnet.eth-2-1.rx_pipe_ip6_routed_multicast_packets: 0
vnet.eth-2-1.rx_pipe_l3_interface_bytes: 0
vnet.eth-2-1.rx_pipe_l3_interface_packets: 0
vnet.eth-2-1.rx_pipe_niv_forwarding_error_drops: 0
vnet.eth-2-1.rx_pipe_niv_frame_error_drops: 0
vnet.eth-2-1.rx_pipe_port_table_bytes: 31921075930770
vnet.eth-2-1.rx_pipe_port_table_packets: 197043678585
vnet.eth-2-1.rx_pipe_spanning_tree_state_not_forwarding_drops: 0
vnet.eth-2-1.rx_pipe_trill_non_trill_drops: 0
vnet.eth-2-1.rx_pipe_trill_packets: 0
vnet.eth-2-1.rx_pipe_trill_trill_drops: 0
vnet.eth-2-1.rx_pipe_unicast_packets: 197044262157
vnet.eth-2-1.rx_pipe_unknown_vlan_drops: 0
vnet.eth-2-1.rx_pipe_vlan_tagged_packets: 0
vnet.eth-2-1.rx_pipe_zero_port_bitmap_drops: 44150971
```

goes hget platina vnet.eth-2-1.tx_pipe

```
vnet.eth-2-1.tx_pipe_cpu_queue_0_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_0_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_10_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_10_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_11_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_11_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_12_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_12_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_13_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_13_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_14_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_14_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_15_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_15_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_16_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_16_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_17_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_17_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_18_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_18_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_19_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_19_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_1_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_1_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_20_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_20_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_21_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_21_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_22_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_22_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_23_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_23_packets: 0
```

```
vnet.eth-2-1.tx_pipe_cpu_queue_24_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_24_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_25_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_25_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_26_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_26_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_27_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_27_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_28_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_28_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_29_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_29_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_2_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_2_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_30_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_30_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_31_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_31_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_32_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_32_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_33_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_33_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_34_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_34_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_35_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_35_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_36_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_36_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_37_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_37_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_38_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_38_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_39_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_39_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_3_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_3_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_40_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_40_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_41_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_41_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_42_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_42_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_43_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_43_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_44_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_44_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_45_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_45_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_46_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_46_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_47_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_47_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_4_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_4_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_5_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_5_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_6_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_6_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_7_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_7_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_8_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_8_packets: 0
vnet.eth-2-1.tx_pipe_cpu_queue_9_bytes: 0
vnet.eth-2-1.tx_pipe_cpu_queue_9_packets: 0
vnet.eth-2-1.tx_pipe_debug_8: 0
vnet.eth-2-1.tx_pipe_debug_9: 0
vnet.eth-2-1.tx_pipe_debug_a: 0
vnet.eth-2-1.tx_pipe_debug_b: 0
vnet.eth-2-1.tx_pipe_ecn_errors: 0
vnet.eth-2-1.tx_pipe_invalid_vlan_drops: 0
vnet.eth-2-1.tx_pipe_ip4_unicast_aged_and_dropped_packets: 0
vnet.eth-2-1.tx_pipe_ip4_unicast_packets: 270900830162
vnet.eth-2-1.tx_pipe_ip_length_check_drops: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos0_bytes: 738
vnet.eth-2-1.tx_pipe_multicast_queue_cos0_packets: 7
vnet.eth-2-1.tx_pipe_multicast_queue_cos1_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos1_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos2_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos2_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos3_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos3_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos4_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos4_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos5_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos5_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos6_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos6_packets: 0
```

```
vnet.eth-2-1.tx_pipe_multicast_queue_cos7_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_cos7_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_qm_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_qm_packets: 0
vnet.eth-2-1.tx_pipe_multicast_queue_sc_bytes: 0
vnet.eth-2-1.tx_pipe_multicast_queue_sc_packets: 0
vnet.eth-2-1.tx_pipe_packet_aged_drops: 0
vnet.eth-2-1.tx_pipe_packets_dropped: 0
vnet.eth-2-1.tx_pipe_port_table_bytes: 43885935264330
vnet.eth-2-1.tx_pipe_port_table_packets: 270900834965
vnet.eth-2-1.tx_pipe_purge_cell_error_drops: 0
vnet.eth-2-1.tx_pipe_spanning_tree_state_not_forwarding_drops: 0
vnet.eth-2-1.tx_pipe_trill_access_port_drops: 0
vnet.eth-2-1.tx_pipe_trill_non_trill_drops: 0
vnet.eth-2-1.tx_pipe_trill_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos0_bytes: 43885934763102
vnet.eth-2-1.tx_pipe_unicast_queue_cos0_packets: 270900831871
vnet.eth-2-1.tx_pipe_unicast_queue_cos1_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos1_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos2_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos2_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos3_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos3_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos4_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos4_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos5_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos5_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos6_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos6_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos7_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_cos7_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_qm_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_qm_packets: 0
vnet.eth-2-1.tx_pipe_unicast_queue_sc_bytes: 0
vnet.eth-2-1.tx_pipe_unicast_queue_sc_packets: 0
vnet.eth-2-1.tx_pipe_vlan_tagged_packets: 0
```

fe1 counters

fe1 counters captures counters not associated with any front panel interface. Examples are loopback port and cpu ports as seen back the switch ASIC.

Example:

```
goes hget platina fe1
vnet.fe1-cpu.drops: 0
vnet.fe1-cpu.punts: 0
vnet.fe1-cpu.rx_bytes: 0
vnet.fe1-cpu.rx_packets: 0
vnet.fe1-cpu.rx_pipe_l3_interface_bytes: 0
vnet.fe1-cpu.rx_pipe_l3_interface_packets: 0
vnet.fe1-cpu.tx_bytes: 0
vnet.fe1-cpu.tx_packets: 0
vnet.fe1-pipe0-loopback.drops: 0
vnet.fe1-pipe0-loopback.punts: 0
vnet.fe1-pipe0-loopback.rx_bytes: 0
vnet.fe1-pipe0-loopback.rx_packets: 0
vnet.fe1-pipe0-loopback.rx_pipe_l3_interface_bytes: 0
vnet.fe1-pipe0-loopback.rx_pipe_l3_interface_packets: 0
vnet.fe1-pipe0-loopback.tx_bytes: 0
vnet.fe1-pipe0-loopback.tx_packets: 0
vnet.fe1-pipe1-loopback.drops: 0
vnet.fe1-pipe1-loopback.punts: 0
vnet.fe1-pipe1-loopback.rx_bytes: 0
vnet.fe1-pipe1-loopback.rx_packets: 0
vnet.fe1-pipe1-loopback.rx_pipe_l3_interface_bytes: 0
vnet.fe1-pipe1-loopback.rx_pipe_l3_interface_packets: 0
vnet.fe1-pipe1-loopback.tx_bytes: 0
vnet.fe1-pipe1-loopback.tx_packets: 0
vnet.fe1-pipe2-loopback.drops: 0
vnet.fe1-pipe2-loopback.punts: 0
vnet.fe1-pipe2-loopback.rx_bytes: 0
vnet.fe1-pipe2-loopback.rx_packets: 0
vnet.fe1-pipe2-loopback.rx_pipe_l3_interface_bytes: 0
vnet.fe1-pipe2-loopback.rx_pipe_l3_interface_packets: 0
vnet.fe1-pipe2-loopback.tx_bytes: 0
vnet.fe1-pipe2-loopback.tx_packets: 0
vnet.fe1-pipe3-loopback.drops: 0
vnet.fe1-pipe3-loopback.punts: 0
```

```
vnet.fe1-pipe3-loopback.rx_bytes: 0
vnet.fe1-pipe3-loopback.rx_packets: 0
vnet.fe1-pipe3-loopback.rx_pipe_l3_interface_bytes: 0
vnet.fe1-pipe3-loopback.rx_pipe_l3_interface_packets: 0
vnet.fe1-pipe3-loopback.tx_bytes: 0
vnet.fe1-pipe3-loopback.tx_packets: 0
```

Eth0 stats

Eth0 is the RJ45 management ethernet port that goes straight from the CPU to/from the front panel RJ45. All the eth0 stats are available in Linux also.

Example:

goes hget platina eth0

```
eth0.[172.30.0.101/24].address: 172.30.0.101
eth0.[172.30.0.101/24].broadcast: 172.30.0.255
eth0.[172.30.0.101/24].flags: PERMANENT
eth0.[172.30.0.101/24].label: eth0
eth0.[172.30.0.101/24].local: 172.30.0.101
eth0.address: 02:46:8a:00:07:56
eth0.admin: up
eth0.broadcast: ff:ff:ff:ff:ff:ff
eth0.carrier: 1
eth0.carrier_changes: 2
eth0.fe80::46:8aff:fe00:756/64.address: fe80::46:8aff:fe00:756
eth0.fe80::46:8aff:fe00:756/64.flags: PERMANENT
eth0.group: 0
eth0.gso_max_segs: 65535
eth0.gso_max_size: 65536
eth0.ifname: eth0
eth0.link: up
eth0.linkmode: 0
eth0.may-broadcast: true
eth0.mtu: 1500
eth0.multicast-encap: enabled
eth0.num_rx_queues: 8
eth0.num_tx_queues: 8
eth0.operstate: up
eth0.promiscuity: 0
eth0.proto_down: 0
eth0.qdisc: mq
eth0.running: yes
eth0.rx_bytes: 1813615176
eth0.rx_packets: 19885573
eth0.tx_bytes: 3639168708
eth0.tx_packets: 26745607
eth0.txqlen: 1000
```

Loopback port stats

lo is the Linux loopback port. All the lo stats are available in Linux also.

Example:

goes hgetall platina lo.

```
lo.2001:5:3::12/128.address: 2001:5:3::12
lo.2001:5:3::12/128.flags: PERMANENT
lo.::1/128.address: ::1
lo.::1/128.flags: PERMANENT
lo.[12.12.12.12/32].address: 12.12.12.12
lo.[12.12.12.12/32].flags: PERMANENT
lo.[12.12.12.12/32].label: lo
lo.[12.12.12.12/32].local: 12.12.12.12
lo.[127.0.0.1/8].address: 127.0.0.1
lo.[127.0.0.1/8].flags: PERMANENT
lo.[127.0.0.1/8].label: lo
lo.[127.0.0.1/8].local: 127.0.0.1
lo.address: 00:00:00:00:00:00
lo.admin: up
lo.broadcast: 00:00:00:00:00:00
lo.carrier: 1
```

```
lo.carrier_changes: 0
lo.group: 0
lo.gso_max_segs: 65535
lo.gso_max_size: 65536
lo.ifname: lo
lo.link: up
lo.linkmode: 0
lo.loopback: enabled
lo.mtu: 65536
lo.num_rx_queues: 1
lo.num_tx_queues: 1
lo.operstate: unknown
lo.promiscuity: 0
lo.proto_down: 0
lo.qdisc: noqueue
lo.running: yes
lo.rx_bytes: 49408330
lo.rx_packets: 49046
lo.tx_bytes: 49408330
lo.tx_packets: 49046
lo.txqlen: 1
```

EEPROM contents

These Redis fields show the content of the EEPROM that includes manufacturing information on the specific hardware unit such as the part number and serial number.

Example:

```
goes hget platina eeprom
eeprom.BaseEthernetAddress: 50:18:4c:00:13:04
eeprom.Crc: 0xbff00bd2
eeprom.DeviceVersion: 0x0a
eeprom.ManufactureDate: 2016/11/11 11:11:11
eeprom.NEthernetAddress: 134
eeprom.Onie.Data: "TlvInfo\x00" 0x546c76496e666f00
eeprom.Onie.Version: 0x01
eeprom.PartNumber: PS-3001-32C
eeprom.PlatformName: X86-BDE-4C-16GB-128GB
eeprom.SerialNumber: HAY16B7300003
eeprom.Vendor: Platina Systems
eeprom.VendorExtension: "\x00\x00\xbceP\x01\x00Q\x01\x02R\x01\nS\x0e900-000002-000"
```

Appendix 2: BMC Redis

The BMC runs a separate Redis server that provides configuration and status of the hardware platform including power supplies, fan trays, and environmental monitoring.

Accessing the BMC redis server should be done via Redis client. For convenience, the redis-tools package is pre-installed in the switch's Linux ("sudo apt-get install redis-tools") and includes the redis-cli command that can be used to access the BMC redis from the switch's Linux CLI.

The BMC Redis server listens on port 6379 of the BMC's eth0 IPv4 address and IPv6 link local address.

Connecting via IPv4

By default the BMC eth0 IPv4 address is 192.168.101.100. To connect from the switch's Linux CLI:

```
root@platina:~# redis-cli -h 192.168.101.100
192.168.101.100:6379> hget platina machine
"platina-mk1-bmc"
```

Connecting via IPv6 Link Local

The BMC eth0 IPv6 link local address can be directly translated from the BMC eth0 MAC address. The BMC eth0 MAC address is simply the base MAC address of the system which can be retrieved from the switch's Linux CLI:

```
root@platina:~# goes hget platina eeprom.BaseEthernetAddress
50:18:4c:00:13:04
```

Convert to 6c:ec:5a:07:c8:30 MAC address to IPv6 link local: fe80::5218:4cff:fe00:1304 and connect to BMC redis:

```
root@platina:~# redis-cli -h fe80::5218:4cff:fe00:1304%eth0
[fe80::5218:4cff:fe00:1304%eth0]:6379> hget platina machine
"platina-mk1-bmc"
```

Notable BMC Redis Fields

Temperature status

```
root@platina:~# redis-cli --raw -h fe80::5218:4cff:fe00:1304%eth0 hget platina temp
bmc.temperature.units.C: 37.01
hwmon.front.temp.units.C: 49.000
hwmon.rear.temp.units.C: 54.000
psu1.temp1.units.C: 33.375
psu1.temp2.units.C: 37.812
```

Fan and PSU status

```
root@platina:~# redis-cli --raw -h fe80::5218:4cff:fe00:1304%eth0 hget platina status
fan_tray.1.status: ok.front->back
fan_tray.2.status: ok.front->back
fan_tray.3.status: ok.front->back
fan_tray.4.status: ok.front->back
psu1.status: powered_on
psu2.status: not_installed
```

Fan Speed

```
root@platina:~# redis-cli --raw -h fe80::5218:4cff:fe00:1304%eth0 hget platina fan_tray
fan_tray.1.1.speed.units.rpm: 7031
fan_tray.1.2.speed.units.rpm: 7031
fan_tray.1.status: ok.front->back
fan_tray.2.1.speed.units.rpm: 7031
fan_tray.2.2.speed.units.rpm: 6490
fan_tray.2.status: ok.front->back
fan_tray.3.1.speed.units.rpm: 7031
```

```
fan_tray.3.2.speed.units.rpm: 7031
fan_tray.3.status: ok.front->back
fan_tray.4.1.speed.units.rpm: 7031
fan_tray.4.2.speed.units.rpm: 6490
fan_tray.4.status: ok.front->back
fan_tray.duty: 0x4d
fan_tray.speed: auto
```

PSU Information

```
root@platina:~# redis-cli --raw -h fe80::5218:4cff:fe00:1304%eth0 hget platina psu
psu1.admin.state: enabled
psu1.eeprom:
0100000010900f5010819c54757202020cb47572d4352505335353020ca585858585858585858c3585846ce505
3555134303030303037475720c0c0c1000000000000000002f00021822c42602390319052823b036504620672f3f
0c1f94c20000001f01020d09e701b0047404ec047800e803c8af01820d274982b0047404ec0478000000b80b20202
0202020202020202020202020202020202020202020202020202020202020202020202020202020202020202020
2020202020202020202020202020202020202020202020202020202020202020202020202020202020202020202
0202020202020202020202020202020202020202020202020
psu1.fan_speed.units.rpm: 4020
psu1.i_out.units.A: 10.688
psu1.mfg_id: Great Wall
psu1.mfg_model: CRPS550
psu1.p_in.units.W: 141.000
psu1.p_out.units.W: 129.000
psu1.status: powered_on
psu1.temp1.units.C: 33.375
psu1.temp2.units.C: 37.812
psu1.v_in.units.V: 117.000
psu1.v_out.units.V: 12.047
psu2.admin.state: enabled
psu2.status: not_installed
```

Power Monitor Information

```
root@platina:~# redis-cli --raw -h fe80::5218:4cff:fe00:1304%eth0 hget platina vmon
vmon.1v0.tha.units.V: 1.041
vmon.1v0.thc.units.V: 1.014
vmon.1v2.ethx.units.V: 1.187
vmon.1v25.sys.units.V: 1.24
vmon.1v8.sys.units.V: 1.805
vmon.3v3.bmc.units.V: 3.28
vmon.3v3.sb.units.V: 3.344
vmon.3v3.sys.units.V: 3.298
vmon.3v8.bmc.units.V: 3.826
vmon.5v.sb.units.V: 4.926
vmon.poweroff.events: 1970-01-01T23:30:34Z.1970-01-01T23:32:28Z.1970-01-01T23:40:17Z.1970-01-
01T23:50:54Z.1970-01-01T23:52:04Z
```