

# DASH bmv2 Model Testing Update

## DASH Working Group

### July 27, 2022

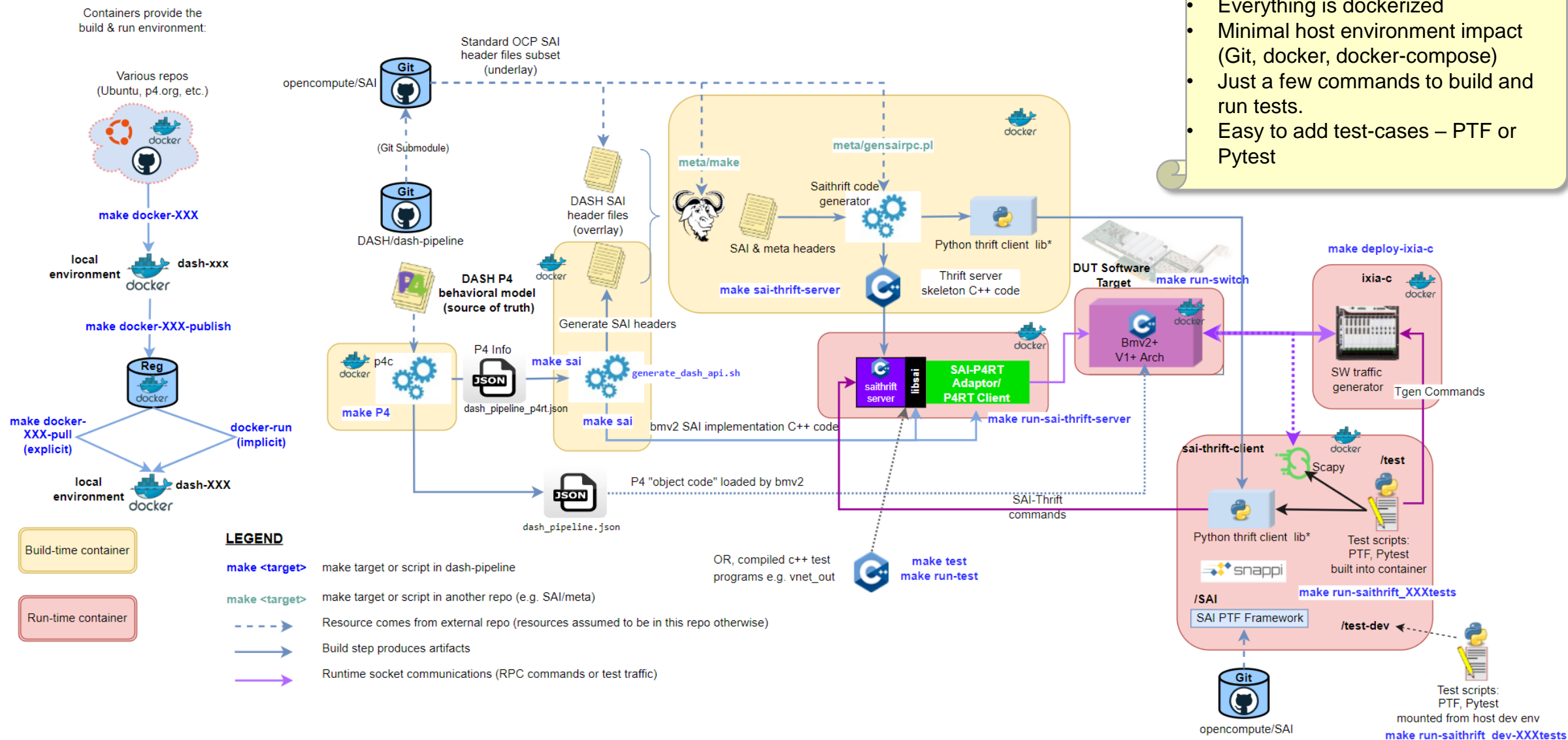
Chris Sommers  
2022-07-27

## Progress Since Last update (2022-06-08)

- Complete workflow: Build -> Test with 3 commands
  - Compile P4 code (was already working)
  - libsai headers & adaptor enhancements (added sai\_api\_query() etc.)
  - Build saithrift server & client autogeneration (using autogen framework from Intel)
  - Issues & PRs filed w/ SAI project to handle SAI extension headers in sairpcgen
  - Build saithrift-client docker: contains all tools & test scripts, self-contained
  - PTF and Pytest framework integration
  - Import all existing SAI PTF framework and test cases (via submodule)
  - Wrote a few exemplar test-cases for PTF and Pytest. Easy to mimic for new cases.
  - READMEs, READMEs, READMEs...
  - Dockerfile streamlining – *smaller, faster, cheaper!* Removed almost all host dependencies.
  - ACR (Azure Container Registry) created, migration to this is TODO
  - Identified some gaps but have workarounds for now
  - Lots of help from community trying out dev branches – thanks everyone!

# BMV2 Build/Test Workflow is implemented!

- Everything is dockerized
- Minimal host environment impact (Git, docker, docker-compose)
- Just a few commands to build and run tests.
- Easy to add test-cases – PTF or Pytest



# Get up and running in a few minutes!

## Quick-start

### Prerequisites

See [Installing Prerequisites](#) for details.

- Ubuntu 20.04, bare-metal or VM
- 2 CPU cores minimum, 7GB RAM, 14Gb HD; same as [free Azure 2-core GitHub runner instances](#),
- git - tested with version 2.25.1
- docker
- [docker-compose](#) (1.29.2 or later)

### Clone this repo

```
git clone <repo URL>
cd DASH
```

Optional - if you require a particular dev branch:

```
git checkout <branch>
```

Init (clone) the SAI submodule:

```
git submodule update --init # NOTE --recursive not needed (yet)
```

## I feel lucky!

Eager to see it work? First [clone this repo](#), then do the following:

In first terminal (console will print bmv2 logs):

```
cd dash-pipeline
make clean && make all run-switch
```

1

Three make commands to build & test

The above procedure takes awhile since it has to pull docker images (once) and build some code.

In second terminal (console will print saithrift server logs):

```
make run-saithrift-server
```

2

In third terminal (console will print test results):

```
make run-all-tests
```

3

When you're done, do:

```
make kill-all      # just to stop the daemons
                   # you can redo "run" commands w/o rebuilding
```



# CI Pipeline – Working!

Azure / DASHPublic

Edit PinsUnwatch33Fork42

<> CodeIssues44Pull requests19ActionsProjectsWikiSecurityInsights

WorkflowsNew workflow

All workflows

DASH-BMV2-CI

DASH-docker-bmv2-bldr-buil...

DASH-docker-grpc1.43.2-buil...

DASH-docker-p4c-bmv2-buil...

DASH-docker-saithrift-bldr-b...

DASH-docker-saithrift-client-...

DASH-docker-saithrift-client-i...

All workflowsShowing runs from all workflows

Filter workflow runs

157 workflow runs

	Event	Status	Branch	Actor
saithrift client & server integration with DASH bmv2 DASH-docker-saithrift-client-image #1: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 15m 30s	...	
saithrift client & server integration with DASH bmv2 DASH-docker-saithrift-bldr-build-image #23: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 12m 43s	...	
saithrift client & server integration with DASH bmv2 DASH-docker-saithrift-client-bldr-image #1: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 17m 9s	...	
saithrift client & server integration with DASH bmv2 DASH-docker-p4c-bmv2-build-image #20: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 1m 33s	...	
saithrift client & server integration with DASH bmv2 DASH-docker-bmv2-bldr-build-image #23: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 1m 44s	...	
saithrift client & server integration with DASH bmv2 DASH-docker-grpc1.43.2-build-image #23: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 15m 12s	...	
saithrift client & server integration with DASH bmv2 DASH-BMV2-CI #33: Pull request #164 opened by chrissommers	chrissommers:chris-saithri...	20 minutes ago 6m 41s	...	
Add support for IPV4 optional headers DASH-BMV2-CI #32: Pull request #160 reopened by saad-mzhr	saad-mzhr:ipv4-options	13 hours ago 2m 0s	...	

The main build-test workflow (others are just docker image builds)

# CI Pipeline – Drill down

## Build and Test Sirius Pipeline

succeeded 23 hours ago in 8m 30s

- Set up job
- Run actions/checkout@v3
- Pull docker p4c image
- Build P4 software switch (bmv2) and P4Info
- Install SAI submodule
- Pull docker saithrift-bldr image
- Generate SAI API
- Generate saithrift-server
- Generate saithrift-client local docker
- Pull docker bmv2-bldr image
- Build libsaï c++ tests
- Prepare network
- Run P4 software switch (bmv2) with P4Runtime
- Test SAI library over P4RT to switch
- Run saithrift server
- Run PTF Tests
- Deploy ixia-c Traffic Generator
- Run Pytests
- Post Run actions/checkout@v3
- Complete job

## Build and Test Sirius Pipeline

succeeded 23 hours ago in 8m 30s

Search logs

- Run saithrift server 0s
- Run PTF Tests 1s
  - Run DOCKER\_FLAGS=\$docker\_fg\_flags make run-saithrift-ptftests
  - docker run -u root --network=host --rm -w /tests/ -u root --privileged \
  - w /tests/ \
  - local/dash-saithrift-client:latest \
  - ./saithrift/run-saithrift-ptftests.sh
  - /usr/local/lib/python3.8/dist-packages/ptf-0.9.1-py3.8.egg/EGG-INFO/scripts/ptf:19: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  - import imp
  - test\_thrift\_session.TestSaiThriftSession ... ok
  - 
  - Ran 1 test in 0.002s
  - OK
  - test\_thrift\_session.TestSaiThriftSaiHelper ... ok
  - 
  - Ran 1 test in 0.001s
  - OK
  - test\_saithrift\_vnet.TestSaiThrift\_create\_outbound\_eni\_to\_vni\_entry ... ok
  - 
  - Ran 1 test in 0.217s
  - OK
  - TestSaiThriftSession OK
  - TestSaiThriftSaiHelper OK
  - 
  - Sending packet... <Ether dst=00:01:02:03:04:05 src=00:06:07:08:09:0a type=IPv4 |<IP ihl=None tos=0x0 id=1 flags= frag=0 ttl=64 proto=udp src=192.168.0.1 dst=192.168.0.2 |<UDP sport=1234 dport=80 |<Raw load='\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&'()\*+,-./0123456789' |>>>>
  - Verifying packet... <Ether dst=00:01:02:03:04:05 src=00:06:07:08:09:0a type=IPv4 |<IP ihl=None tos=0x0 id=1 flags= frag=0 ttl=64 proto=udp src=192.168.0.1 dst=192.168.0.2 |<UDP sport=1234 dport=80 |<Raw load='\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&'()\*+,-./0123456789' |>>>>
  - test\_sai\_thrift\_create\_outbound\_eni\_to\_vni\_entry OK
- Deploy ixia-c Traffic Generator 15s
- Run Pytests 8s
- Post Run actions/checkout@v3 0s

# Ready for more test-cases!

```
dash-pipeline
├── bmv2
├── dockerfiles
├── images
├── SAI
├── tests
│   ├── init_switch
│   ├── libsai
│   └── saithrift
│       ├── ptf
│       │   ├── thrift
│       │   │   └── test_thrift_session.py
│       │   └── vnet
│       │       ├── test_saithrift_vnet.py
│       │       ├── README.md
│       └── pytest
│           ├── echo
│           │   └── test_echo_port.py
│           ├── switch
│           │   └── test_saithrift_switch.py
│           ├── thrift
│           │   └── test_saithrift_session.py
│           └── vnet
│               ├── test_saithrift_vnet.py
│               ├── conftest.py
│               ├── pytest.ini
│               ├── saithrift_rpc_client.py
│               ├── README.md
│               ├── run-saithrift-ptftests.sh
│               ├── run-saithrift-pytests.sh
│           └── vnet_out
│               └── vnet_out
```

```
import pytest
import snappi
import scapy

from sai_thrift.sai_headers import *
from sai_base_test import *
# TODO - when switch APIs implemented:
# class TestSaiThrift_create_outbound_eni_to_vni_entry(SaiHelper):

class TestSaiThrift_create_outbound_eni_to_vni_entry(ThriftInterfaceDataPlane):
    """ Test saithrift vnet outbound """
    def setUp(self):
        super(TestSaiThrift_create_outbound_eni_to_vni_entry, self).setUp()
        self.switch_id = 0
        self.eth_addr = '\xaa\xcc\xcc\xcc\xcc\xcc'
        self.vni = 60
        self.eni = 7
        self.dle = sai_thrift_direction_lookup_entry_t(switch_id=self.switch_id, vni=self.vni)
        self.eam = sai_thrift_eni_ether_address_map_entry_t(switch_id=self.switch_id, address = self.eth_addr)
        self.e2v = sai_thrift_outbound_eni_to_vni_entry_t(switch_id=self.switch_id, eni_id=self.eni)

    try:

        status = sai_thrift_create_direction_lookup_entry(self.client, self.dle,
                                                           action=SAI_DIRECTION_LOOKUP_ENTRY_ACTION_SET_OUTBOUND_DIRECTION)
        assert(status == SAI_STATUS_SUCCESS)

        status = sai_thrift_create_eni_ether_address_map_entry(self.client,
                                                                eni_ether_address_map_entry=self.eam,
                                                                eni_id=self.eni)
        assert(status == SAI_STATUS_SUCCESS)

        status = sai_thrift_create_outbound_eni_to_vni_entry(self.client,
                                                             outbound_eni_to_vni_entry=self.e2v,
                                                             vni=self.vni)
        assert(status == SAI_STATUS_SUCCESS)

    except AssertionError as ae:
        # Delete entries which might be lingering from previous failures etc.; ignore failures here
        print ("Cleaning up after failure...")
        sai_thrift_remove_outbound_eni_to_vni_entry(self.client, self.e2v)
        sai_thrift_remove_eni_ether_address_map_entry(self.client, self.eam)
        sai_thrift_remove_direction_lookup_entry(self.client, self.dle)
        raise ae

    def runTest(self):
```

# TODO...

Azure / Projects / 

DASH Test WG

Search or jump to...

DASH Test WG

Summary View

+ New view

Title	Assignees	Status	Notes	ETA - Check Back	Updates	Assignee2
1 SAI Thrift Server Integration w/bmv2 libsai		Done	PR will be prepared	7.27.22	Prelim demo given 7.20.22	Chris
2 libsai for bmv2 - get, set operations		Backlog	jinja2 templates for sai-to-P4RT backend			
3 libsai for bmv2 - sai_api_query		Backlog	jinja2 templates for per-API structures		hardcoded, need template-based	
4 libsai for bmv2 - implement get_switch_attributes()		Backlog	Need to support PTF SaiHelper()		PTF base class setup fails w/o at least	
5 Define Overlay tests (VNET Overlay case)		Backlog	vnet config via SAI + packet test vectors	7.27.22		Chris, Marian, Intel
6 Implement VNET testcase - PTF and/or Pytest		Backlog	Python coding of test case			
7 Volunteer to write more test cases w/Keysight		Backlog		7.27.22	7.25.22 Expect PR this week for test in	Community
8 SAI Thrift Demo		In Progress		7.27.22	Prelim demo given 7.20.22	Chris
9 Community: volunteers to try the PTF test cases using Intel's SAI dev branch		Backlog	Community to try SAI dev branch	8.3.22		Community
10 Merge DASH PTF branch (Intel) with SAI PRs from Chris		Backlog				Intel/Volodymyr
11 Import merged DASH PTF branch back into DASH		Backlog				Intel, Chris
12 Verify and document remote saithrift-client usage		Backlog				Chris
13 Add fixtures/usage for remote Pytest usage		Backlog				Chris
14 Adapt saithrift server and client dockerization to generic SAI		Backlog				
15 Migrate docker images to ACR		Backlog				Keysight/MSFT
16 Migrate CI runners to Azure		Backlog				Keysight/MSFT
17 Research PTF test failure due to unexpected packets (Saad found)		Backlog				Chris

<https://github.com/orgs/Azure/projects/258/views/1> - ask for access