FabTest - NVMe-oF Compliance Test Suite

Swati Chawdhary

Senior Manager

Samsung Semiconductor Inc

Nehal Kumar Ram

Staff Engineer

Samsung Semiconductor Inc

Sathish KM

Associate Technical Director,

Samsung Semiconductor Inc

Raj Kumar Dani

Associate Director,

Samsung Semiconductor Inc

©2024 Conference Concepts, Inc.
All Rights Reserved

Agenda

- NVMe-oF Goals and Trend
- Existing Tools
- FabTest Framework Design
- Key Features
- NVMe Base/Fabrics Command support
- Contributing to FabTest
- Future Work
- Summary

©2024 Conference Concepts, Inc.
All Rights Reserved

NVMe-oF Goals and Trend

- NVMe-oF – Non-volatile memory express over fabrics
  - ✓ Propulsion of NVMe-oF in the market and replacement of iSCSI
  - ✓ Diversity of NVMe/NVMe-oF products deployed
  - ✓ Huge complexity in ensuring compatibility and performance across different implementations.
  - ✓ Robust testing especially with compliance is must.

- Existing Open Source Tools
  - ✓ nvmecompliance (tnvme + dnvme)
  - ✓ nvmetools
  - ✓ pynvme

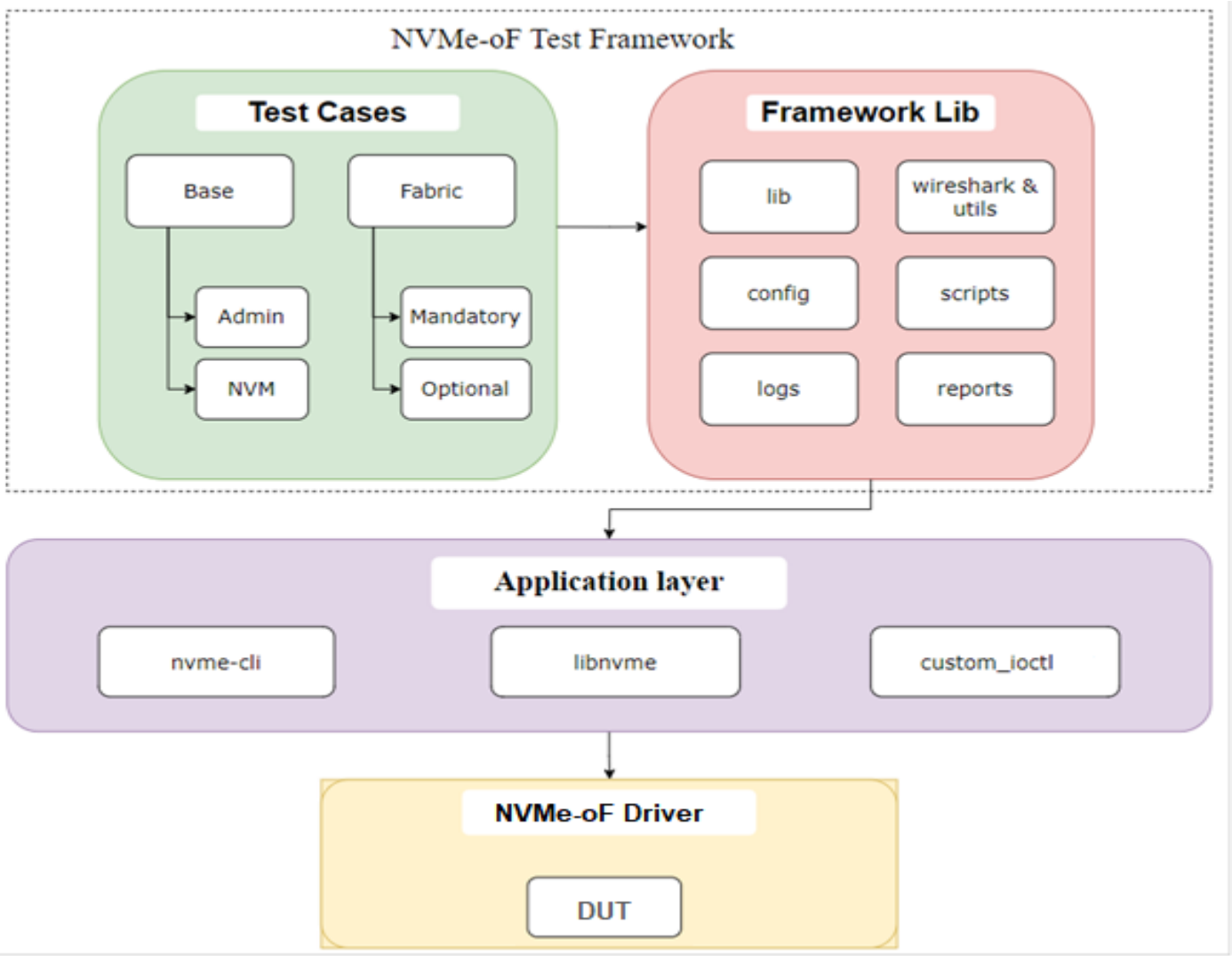
©2024 Conference Concepts, Inc.
All Rights Reserved

Our Goal

- ❑ NVMe-oF Compliance suites available in market are proprietary
  - ✓ Aim – To develop an open source compliance framework
- ❑ Our Solution: FabTest – NVMe-oF compliance Test framework
  - ✓ Easily pluggable to any other standard framework

©2024 Conference Concepts, Inc.
All Rights Reserved

FabTest Framework



- ✓Python pytest based test framework
- ✓Simple CLI interface
- ✓ Application Lib layer - abstracted from TCs
- ✓Supports interfaces to use C based libnvme library
- ✓ Uses pass-through interface of nvme-cli/libnvme

©2024 Conference Concepts, Inc.
All Rights Reserved

FabTest - Key Features

- ✓Support for NVMe Base(1.4), NVMe-oF(1.1a) Protocol
- ✓ Support for RDMA/TCP as the transport protocol
- ✓ Wireshark - Test analysis by packet capture and tracing
- ✓ Error Injection
- ✓ Performance tests

©2024 Conference Concepts, Inc.
All Rights Reserved

NVMe Base/Fabrics Command Support

Commands	Fabtest Support
Connect (M)	✓
Disconnect (O)	✓
Property Get (M)	✓
Property Set (M)	✓
Authentication Send/Receive (O)	✓
Discovery Service (M)	✓

NVMe Fabrics

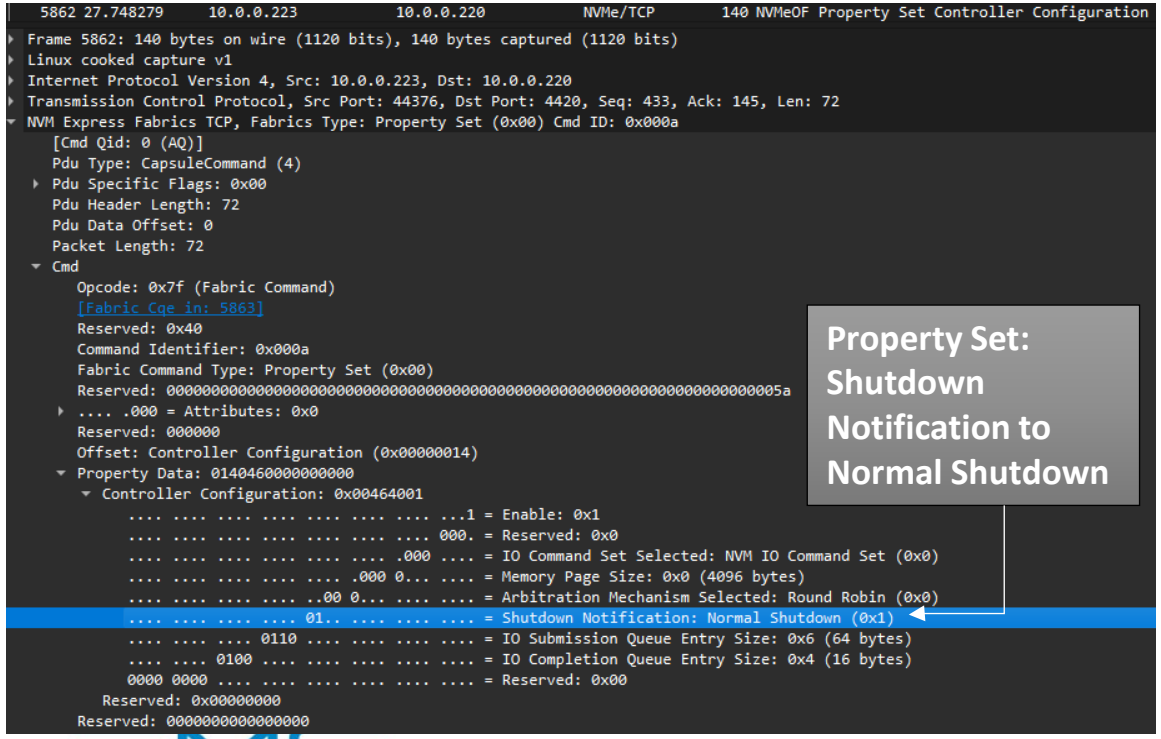
Commands	Fabtest Support
Identify (M)	✓
Get Log page (M)	✓
AER (M)	✓
Abort (M)	✓
Read (M)	✓
Write (M)	✓
Flush (M)	✓
Set Features (M)	✓
Get Features (M)	✓

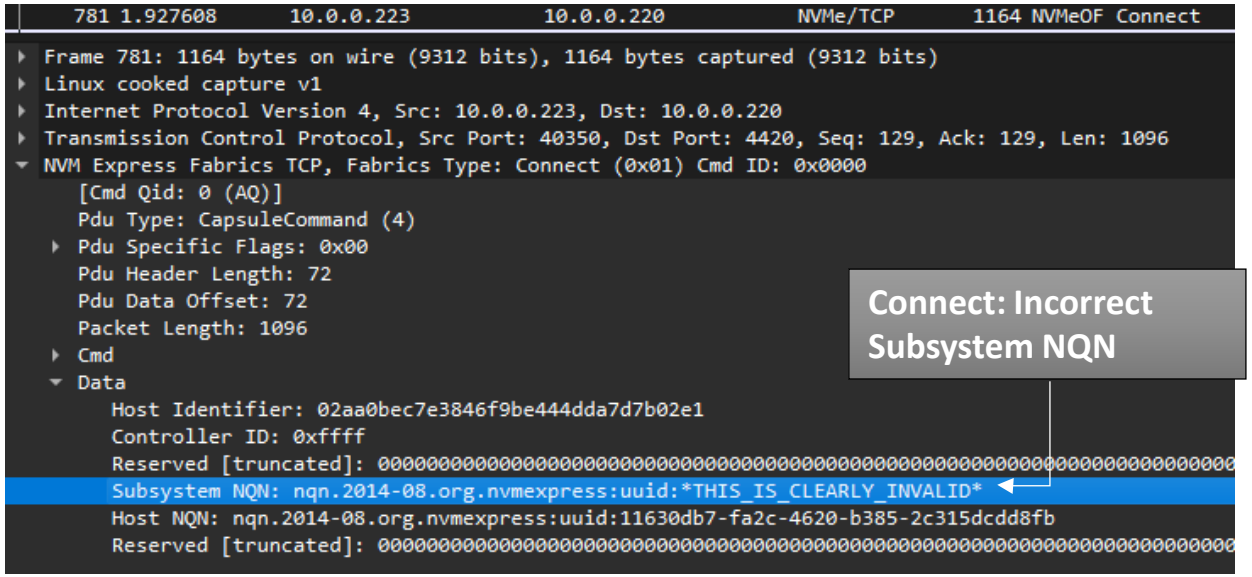
NVMe Base

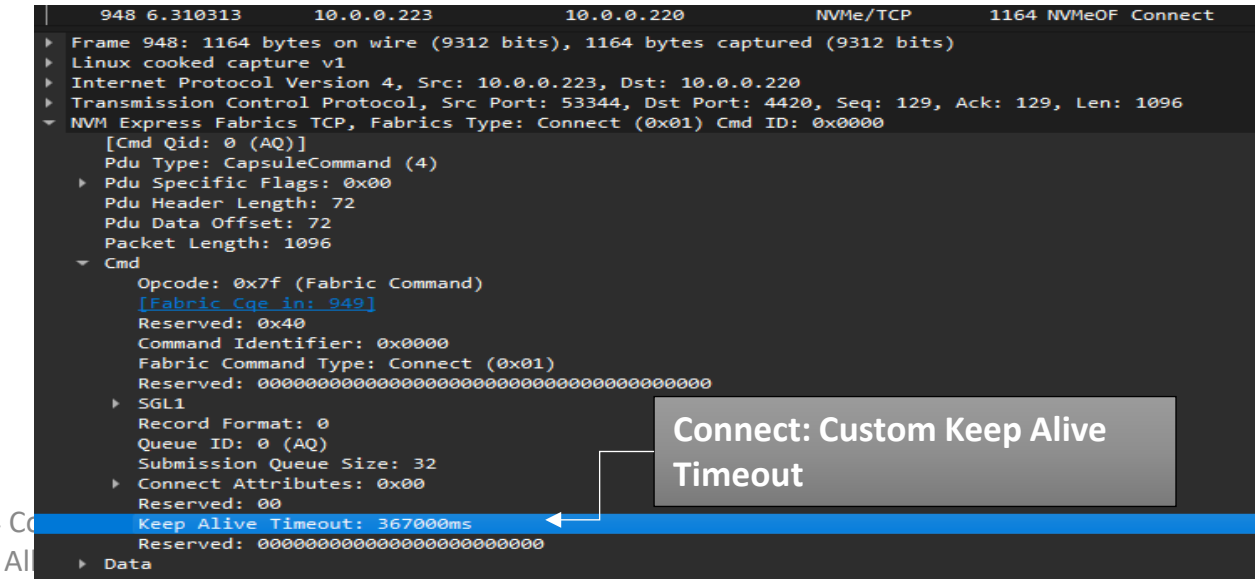
©2024 Conference Concepts, Inc.
All Rights Reserved

Wireshark packet capture and tracing

- Fabtest supports API's to start a packet capture using tcpdump tool
- Check test parameters sent on the wire

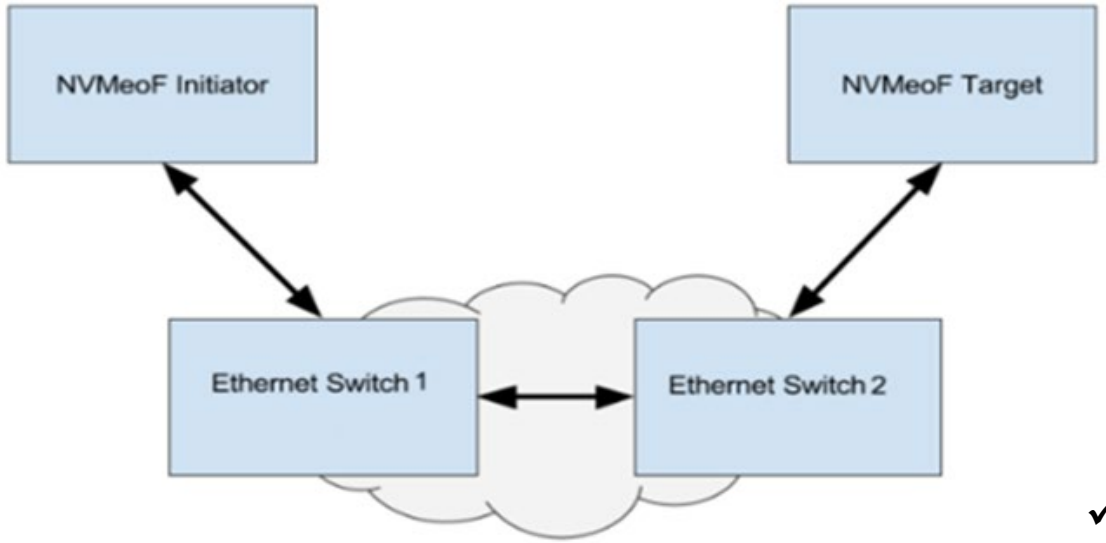






©2024 Co
Al

Error Injection & Performance Tests



Multiple Switch Fabric Topology

Error Injection

- ✓ Network Link/Port Down
- ✓ NVMe command/packet puncture
- ✓ CPU/Memory constraints
- ✓ NVMe hot plug and fault injection testing

Performance Tests

- ✓ Clean/Sustained State IO
- ✓ IO with known data patterns
- ✓ IO with varying data size
- ✓ IO workload generation for high performance and low latency

©2024 Conference Concepts, Inc.
All Rights Reserved

FabTest Github

- Fabtest is Open source @ https://github.com/SamsungDS/nvmFabTest
- Getting Started - https://github.com/SamsungDS/nvmFabTest/blob/main/README.md#code-examples

Code examples

Sample Test Case:

```
class TestNVMeConnect:

    @pytest.fixture(scope='function', autouse=True)
    def setup_method(self, fabConfig):
        ''' Setup Test Case by initialization of controller object '''

        self.fabConfig = fabConfig
        device = self.fabConfig.device
        application = self.fabConfig.application
        self.controller = Controller(device, application)
```

```
    def teardown_method(self):
        '''Teardown test case'''

        # Any memory freeing or disconnections to be made
        # after test case completion
```

```
def test_sample(self):
    ''' Perform test '''

    # Get the required structure from self.controller.cmdlib
    nvme_cmd = self.controller.cmdlib.get_property_get_cmd()

    # Make required modifications as per the test scenario
    offset = 0x14
    nvme_cmd.cmd.generic_command.cdw11.raw = offset

    # Allocate memory for the expected response
    # according to the command and store in nvme_cmd.buff
    get_property_value = ctypes.c_uint64()
        nvme_cmd.buff = ctypes.addressof(get_property_value)

    # Send the command
    res_status = self.controller.app.submit_passthru(nvme_cmd, verify_rsp=True, async_run=False)

    # Make Testing assertions according to scenario
    # using the response data and response status
    if res_status!=0:
        assert False
    if get_property_value.value == 0:
            assert False, f"No value obtained"
        assert True
```

Future Work & Contributing to FabTest

- NVMe Optional commands
- NVMe Upcoming Commands/Features – FDP etc.
- Future NVMe protocol revisions
- NVMe Binding Spec Features
- Contributions can be made towards -
  - New Test cases
  - Add features to the framework
  - Bug fixes/Enhancements

©2024 Conference Concepts, Inc.
All Rights Reserved

Summary

- ❑ Fabtest provides an end to end opensource ecosystem for compliance testing
- ❑ Could be used as a Whitebox test framework for NVMe-oF projects
- ❑ QA/Developers – can build own custom scripts with framework libraries

©2024 Conference Concepts, Inc.
All Rights Reserved

Thank you!

s.chawdhary@samsung.com, sathish.km@samsung.com

©2024 Conference Concepts, Inc.
All Rights Reserved