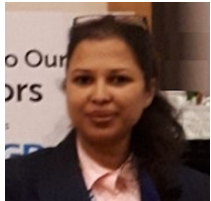


FabTest - NVMe-oF Compliance Test Suite



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



Agenda

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



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

- ✓ nvme compliance (tnvme + dnvme)
- ✓ nvmetools
- ✓ pynvme

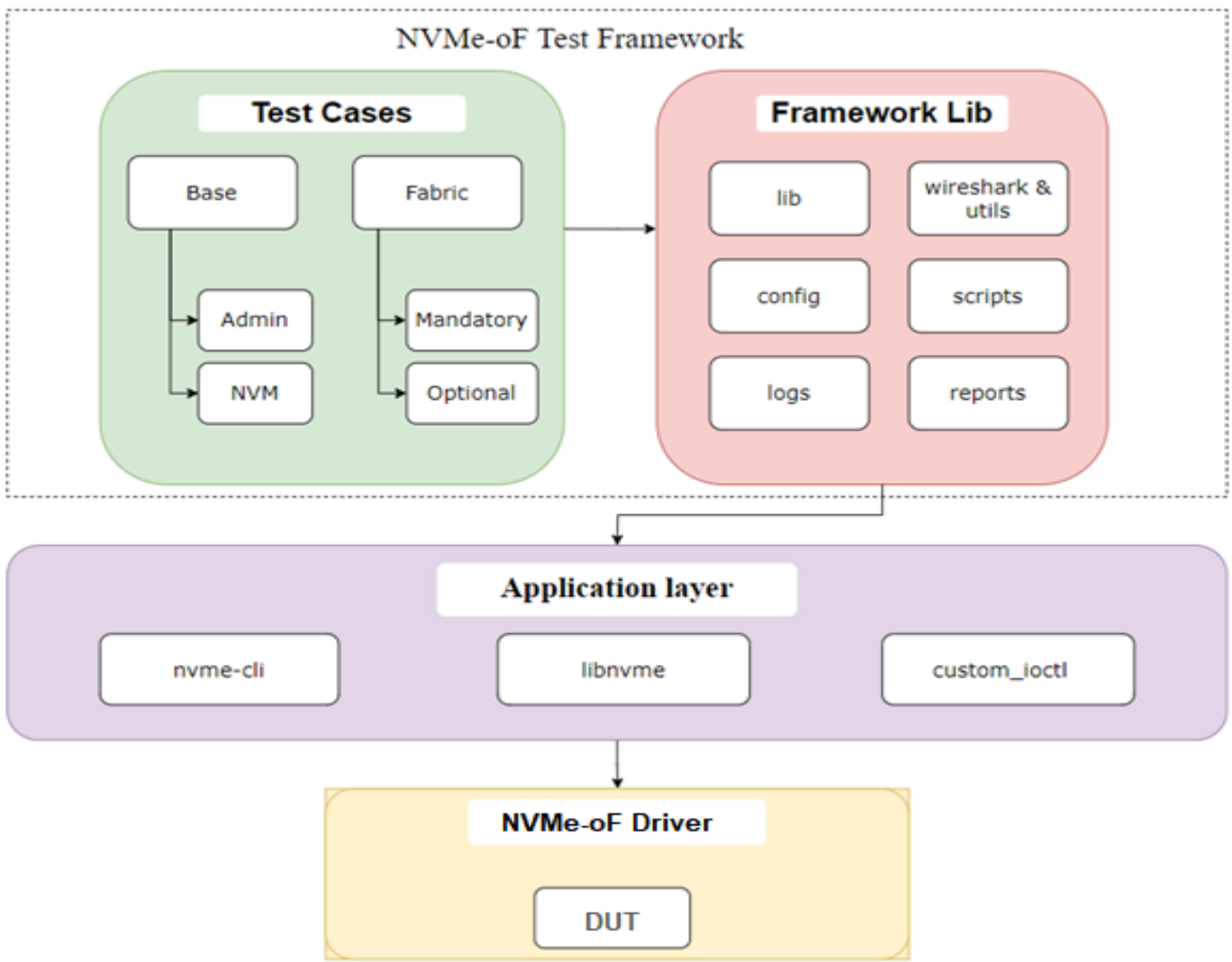


Our Goal

- ❑ 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



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

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



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



Wireshark packet capture and tracing

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

```
5862 27.748279 10.0.0.223 10.0.0.220 NVMe/TCP 140 NVMeOF Property Set Controller Configuration
▶ Frame 5862: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits)
▶ Linux cooked capture v1
▶ Internet Protocol Version 4, Src: 10.0.0.223, Dst: 10.0.0.220
▶ Transmission Control Protocol, Src Port: 44376, Dst Port: 4420, Seq: 433, Ack: 145, Len: 72
▶ NVMe Express Fabrics TCP, Fabrics Type: Property Set (0x00) Cmd ID: 0x000a
  [Cmd Qid: 0 (AQ)]
  Pdu Type: CapsuleCommand (4)
  ▶ Pdu Specific Flags: 0x00
  Pdu Header Length: 72
  Pdu Data Offset: 0
  Packet Length: 72
  Cmd
    Opcode: 0x7f (Fabric Command)
    [Fabric Cqe in: 5863]
    Reserved: 0x40
    Command Identifier: 0x000a
    Fabric Command Type: Property Set (0x00)
    Reserved: 00000000000000000000000000000000000000000000000000000000000000005a
    ▶ ....000 = Attributes: 0x0
    Reserved: 000000
    Offset: Controller Configuration (0x00000014)
    Property Data: 0114046000000000
      ▶ Controller Configuration: 0x00464001
        .... = Enable: 0x1
        ....000. = Reserved: 0x0
        ....000.... = IO Command Set Selected: NVM IO Command Set (0x0)
        ....0000.0... = Memory Page Size: 0x0 (4096 bytes)
        ....0000.... = Arbitration Mechanism Selected: Round Robin (0x0)
        ....01.... = Shutdown Notification: Normal Shutdown (0x1) ←
        ....0110.... = IO Submission Queue Entry Size: 0x6 (64 bytes)
        ....0100.... = IO Completion Queue Entry Size: 0x4 (16 bytes)
        0000 0000.... = Reserved: 0x00
      Reserved: 0x00000000
    Reserved: 0000000000000000
```

**Property Set:
Shutdown
Notification to
Normal Shutdown**



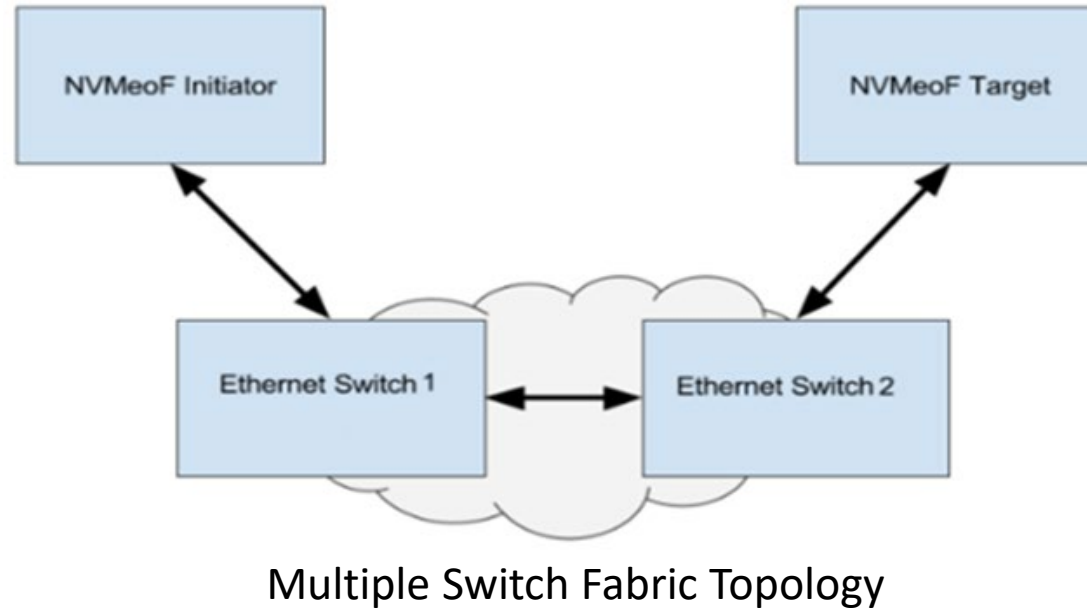
```
781 1.927608 10.0.0.223 10.0.0.220 NVMe/TCP 1164 NVMeOF Connect
▶ Frame 781: 1164 bytes on wire (9312 bits), 1164 bytes captured (9312 bits)
▶ Linux cooked capture v1
▶ Internet Protocol Version 4, Src: 10.0.0.223, Dst: 10.0.0.220
▶ Transmission Control Protocol, Src Port: 40350, Dst Port: 4420, Seq: 129, Ack: 129, Len: 1096
▶ NVMe Express Fabrics TCP, Fabrics Type: Connect (0x01) Cmd ID: 0x0000
  [Cmd Qid: 0 (AQ)]
  Pdu Type: CapsuleCommand (4)
  ▶ Pdu Specific Flags: 0x00
  Pdu Header Length: 72
  Pdu Data Offset: 72
  Packet Length: 1096
  Cmd
    Data
      Host Identifier: 02aa0bec7e3846f9be444dda7d7b02e1
      Controller ID: 0xffff
      Reserved [truncated]: 0000000000000000000000000000000000000000000000000000000000000000
      Subsystem NQN: nqn.2014-08.org.nvmeexpress:uuid:*THIS_IS_CLEARLY_INVALID* ←
      Host NQN: nqn.2014-08.org.nvmeexpress:uuid:11630db7-fa2c-4620-b385-2c315dcd88fb
      Reserved [truncated]: 0000000000000000000000000000000000000000000000000000000000000000
```

**Connect: Incorrect
Subsystem NQN**

```
948 6.310313 10.0.0.223 10.0.0.220 NVMe/TCP 1164 NVMeOF Connect
▶ Frame 948: 1164 bytes on wire (9312 bits), 1164 bytes captured (9312 bits)
▶ Linux cooked capture v1
▶ Internet Protocol Version 4, Src: 10.0.0.223, Dst: 10.0.0.220
▶ Transmission Control Protocol, Src Port: 53344, Dst Port: 4420, Seq: 129, Ack: 129, Len: 1096
▶ NVMe Express Fabrics TCP, Fabrics Type: Connect (0x01) Cmd ID: 0x0000
  [Cmd Qid: 0 (AQ)]
  Pdu Type: CapsuleCommand (4)
  ▶ Pdu Specific Flags: 0x00
  Pdu Header Length: 72
  Pdu Data Offset: 72
  Packet Length: 1096
  Cmd
    Opcode: 0x7f (Fabric Command)
    [Fabric Cqe in: 949]
    Reserved: 0x40
    Command Identifier: 0x0000
    Fabric Command Type: Connect (0x01)
    Reserved: 0000000000000000000000000000000000000000000000000000000000000000
    ▶ SGL1
      Record Format: 0
      Queue ID: 0 (AQ)
      Submission Queue Size: 32
      Connect Attributes: 0x00
      Reserved: 00
      Keep Alive Timeout: 367000ms ←
      Reserved: 00000000000000000000000000000000
    ▶ Data
```

**Connect: Custom Keep Alive
Timeout**

Error Injection & Performance Tests



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



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.buf
    get_property_value = ctypes.c_uint64()
    nvme_cmd.buf = 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



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



Thank you!

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

