SystemVerilog UVM-based Verification Environment for a SpaceFibre Router

Lorenzo Gigli, Pietro Nannipieri, Luca Zulberti,
Simone Vagaggini and Luca Fanucci

9th International SpaceWire and SpaceFibre Conference

Outline

1. Introduction
   - SpaceFibre Standard and Network Layer
   - SotA Verification Methodologies
2. UVM-based Verification IP for SpaceFibre Routers
   - UVM Environment, DUT Interface, and SpFi Virtual Sequences
   - Testing Capabilities
3. Results on *IngeniArs SpFi Routing Switch IP Core*
4. Conclusions

Outline

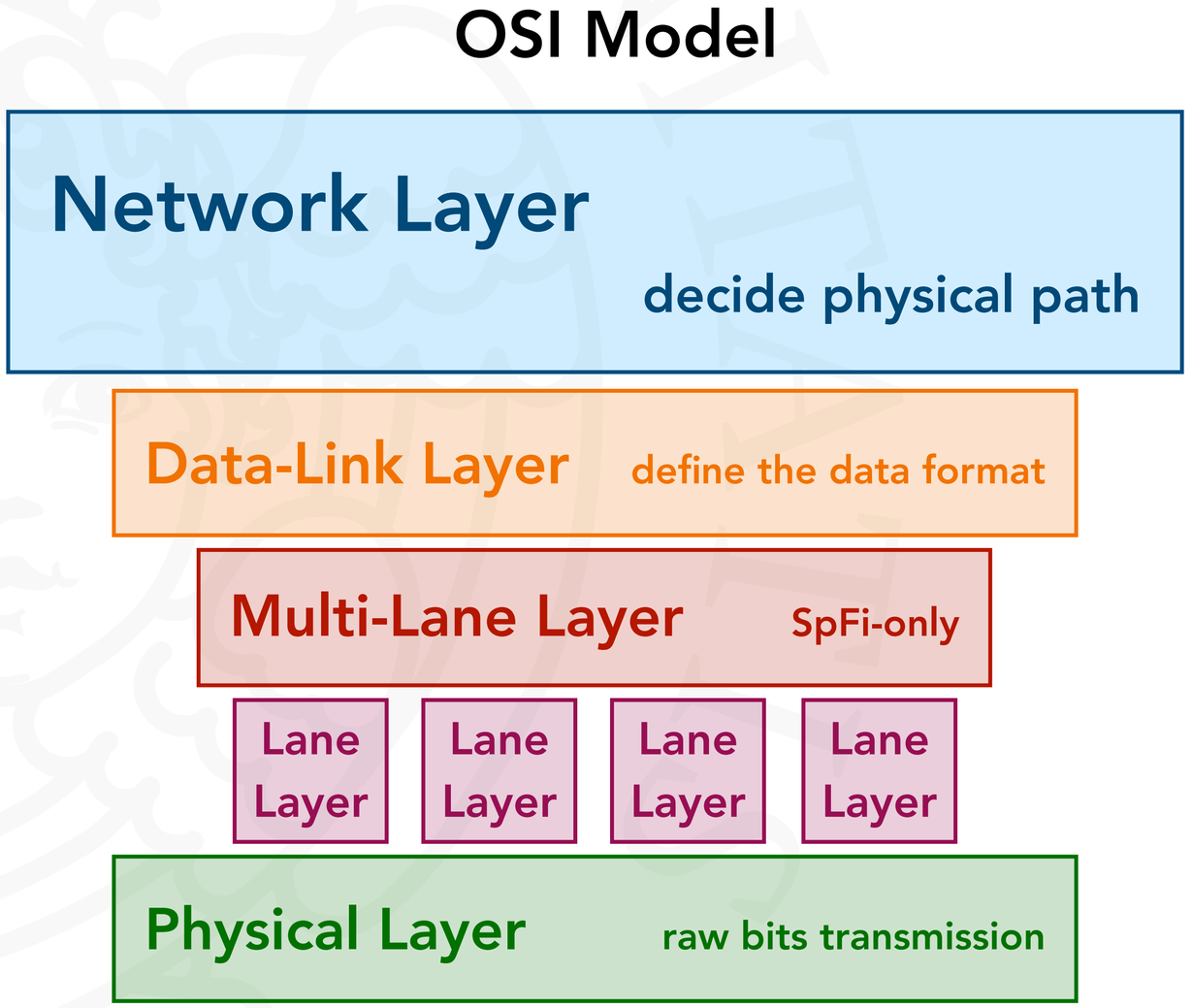
1. Introduction
   - SpaceFibre Standard and Network Layer
   - SotA Verification Methodologies
2. UVM-based Verification IP for SpaceFibre Routers
   - UVM Environment, DUT Interface, and SpFi Virtual Sequences
   - Testing Capabilities
3. Results on *IngeniArs SpFi Routing Switch IP Core*
4. Conclusions

SpaceFibre Standard

1. Very High-Speed Serial Link and **Network** Technology
2. Improvements over *SpaceWire*
   a. Data Rate up to **6.25 Gbps** (16 parallel lanes)
   b. Innovative **QoS Mechanism**
      i. Hardware-separated *Virtual Channels*
   c. Integrate **FDIR**
   d. **Optical Fibre** communication link

SpaceFibre Network Layer

1. Packet Transfer Service (PTS)
   a. Up to 32 Virtual Channels
   b. Transmit *N-Chars* and *FILLs*
2. Broadcast Message Service (BMS)
   a. Transmit short *BC* messages over all the SpFi network
3. **Quality of Service**
   a. Bandwidth Reservation
   b. Priority
   c. Scheduling (Timeslots)



UVM Verification Environment for SpaceFibre Router

SpaceFibre Packet Format

1. Addressing Schemes
   a. Path Addressing
   b. Logical Addressing
   c. Regional Logical Addressing
   d. Group Adaptive Routing
   e. Multicast

2. Cargo types
   a. Data
   b. RMAP

UVM Verification Environment for SpaceFibre Router

Verification and Validation Methodologies for SpaceFibre Network Layer

1. **SpaceART** (SpaceWire/SpaceFibre Analyser Real-Time)
   - *Link-Analyzer*: not an exhaustive RTL validation
2. Network Simulator (e.g. **SHINe)**
   - Hardware-in-the-Loop, but not RTL verification/validation

An exhaustive solution for SpaceFibre Network Level Verification is missing!

UVM Verification Environment for SpaceFibre Router

RTL Verification for SpaceFibre Network Layer

Objective

Build an *UVM-based V*erification IP (VIP) to reach the full functional coverage of the SpaceFibre Network Layer (NL) specification for **any Routing Switch**.

Motivations

1. Provide the first reference in the literature of a VIP for SpFi NL
2. Use UVM which is the IEEE standard for verification
   a. Modular by design
   b. Scalable to any level of complexity (re-usable components)

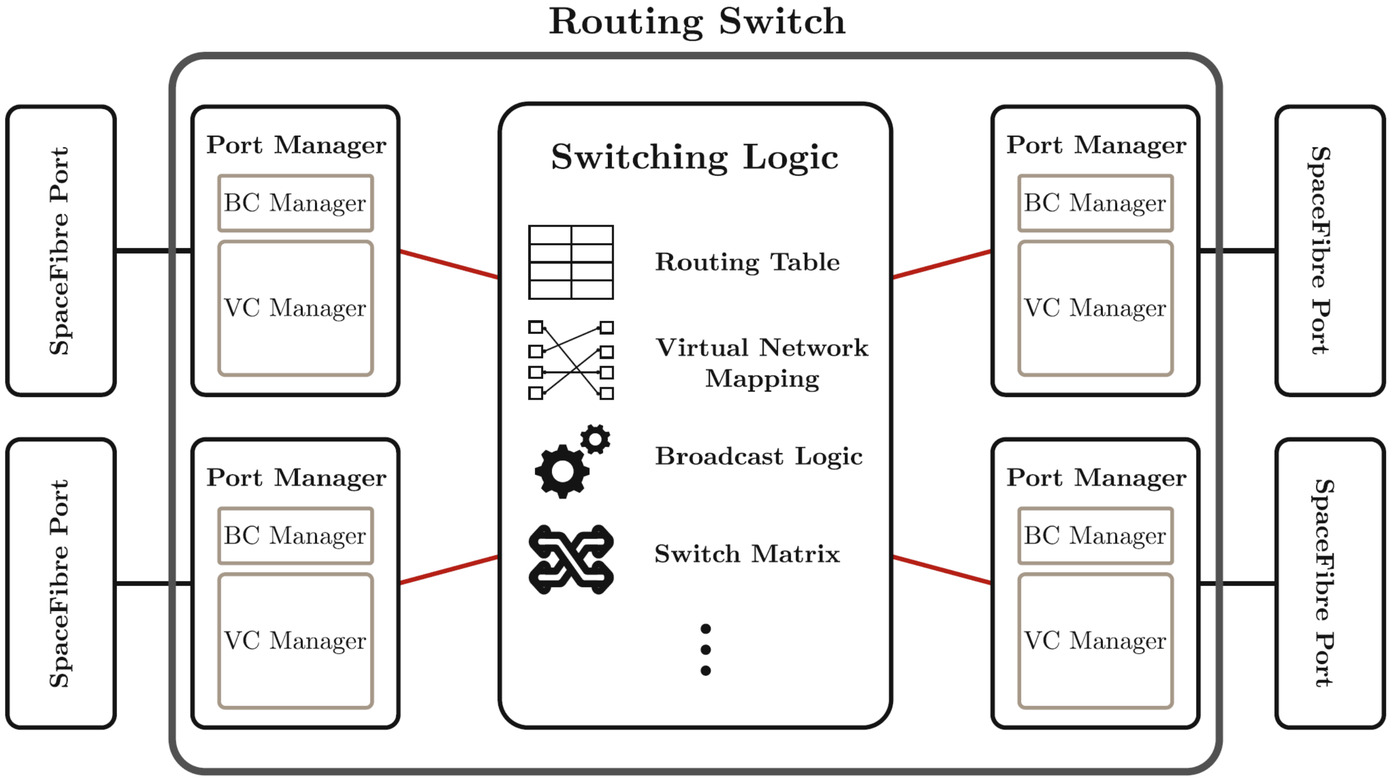
Outline

1. Introduction
   - SpaceFibre Standard and Network Layer
   - SotA Verification Methodologies
2. UVM-based Verification IP for SpaceFibre Routers
   - UVM Environment, DUT Interface, and SpFi Virtual Sequences
   - Testing Capabilities
3. Results on *IngeniArs SpFi Routing Switch IP Core*
4. Conclusions

SpaceFibre Routing Switch

Elements to Test:

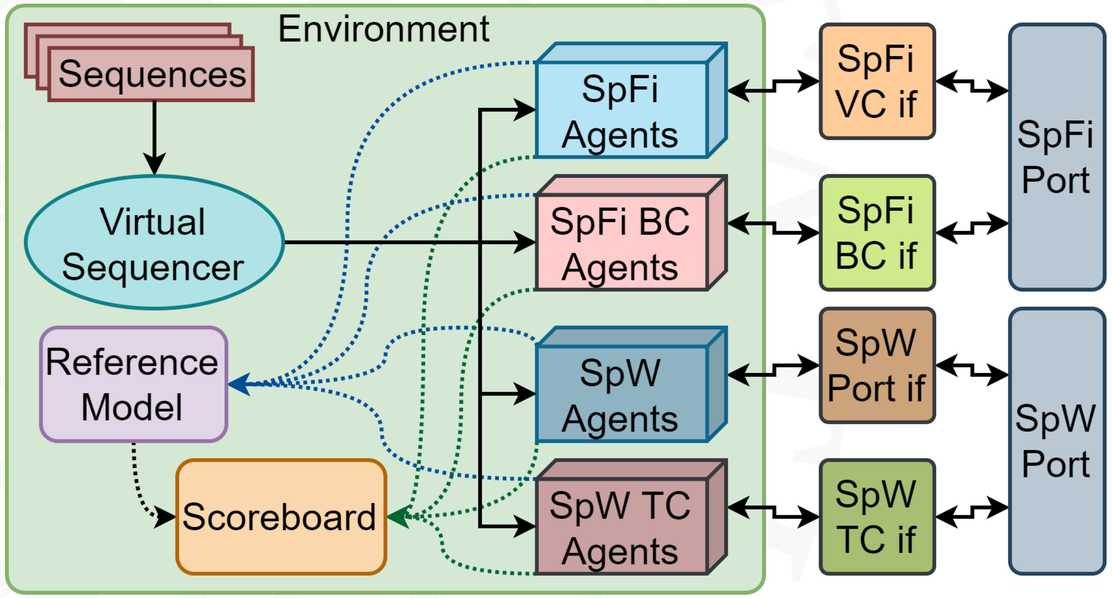
- PTS and BMS
- Routing Table (RT)
- Virtual Network Tables (VNT)
- *RMAP* Target
- Timeout Mechanisms



SpaceFibre UVM Verification Environment

Environment Architecture

1. Testers create *Virtual Sequences*
2. The ENV delivers them to the correct Agent
3. From TX *monitors* to the *Reference Model*
4. From RX *monitors* to the *Scoreboard*
5. Correctness check within the Scoreboard

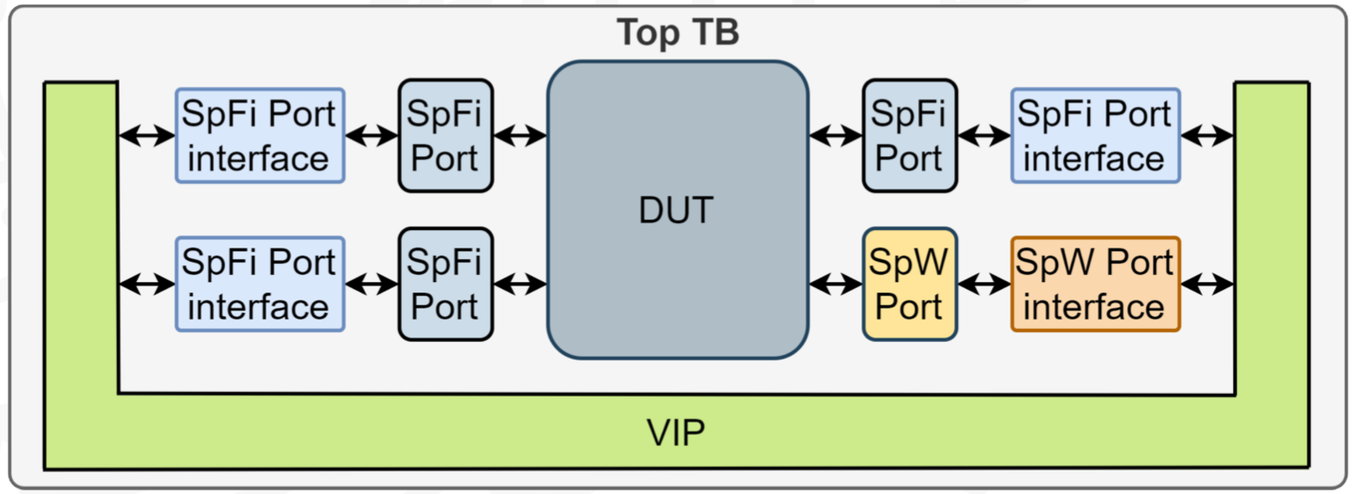


SpaceFibre UVM Verification IP Interoperability

Compliant with *any SpFi Router:* using **proven SpFi and SpW Ports**.

Support for connection in any point of the stack between **DataLink** and **Physical** layers:

- Serial loopback
- Parallel loopback
  - Pre 8b10b
  - Post 8b10b

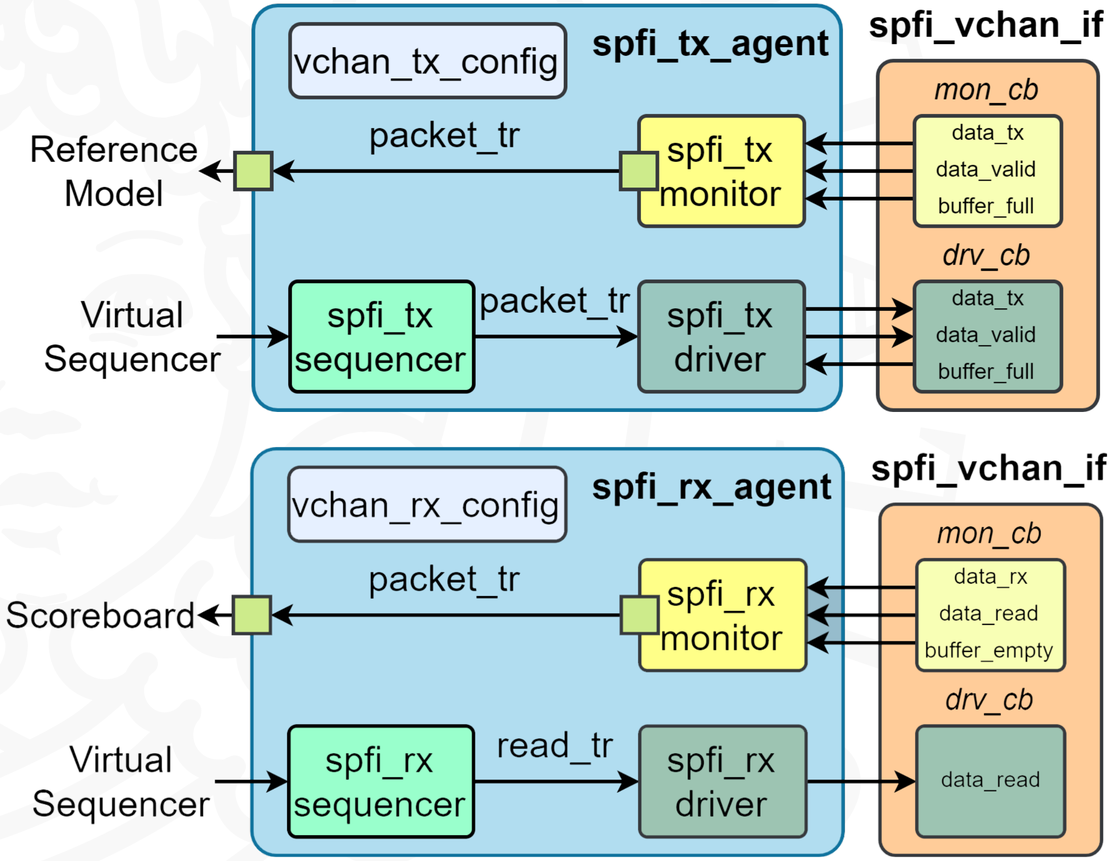


The SpFi/SpW UVM Agents and DUT Connection

- TX Agent
  - Provides *Data* and asserts *Valid*
- RX Agent
  - Asserts *Ready*

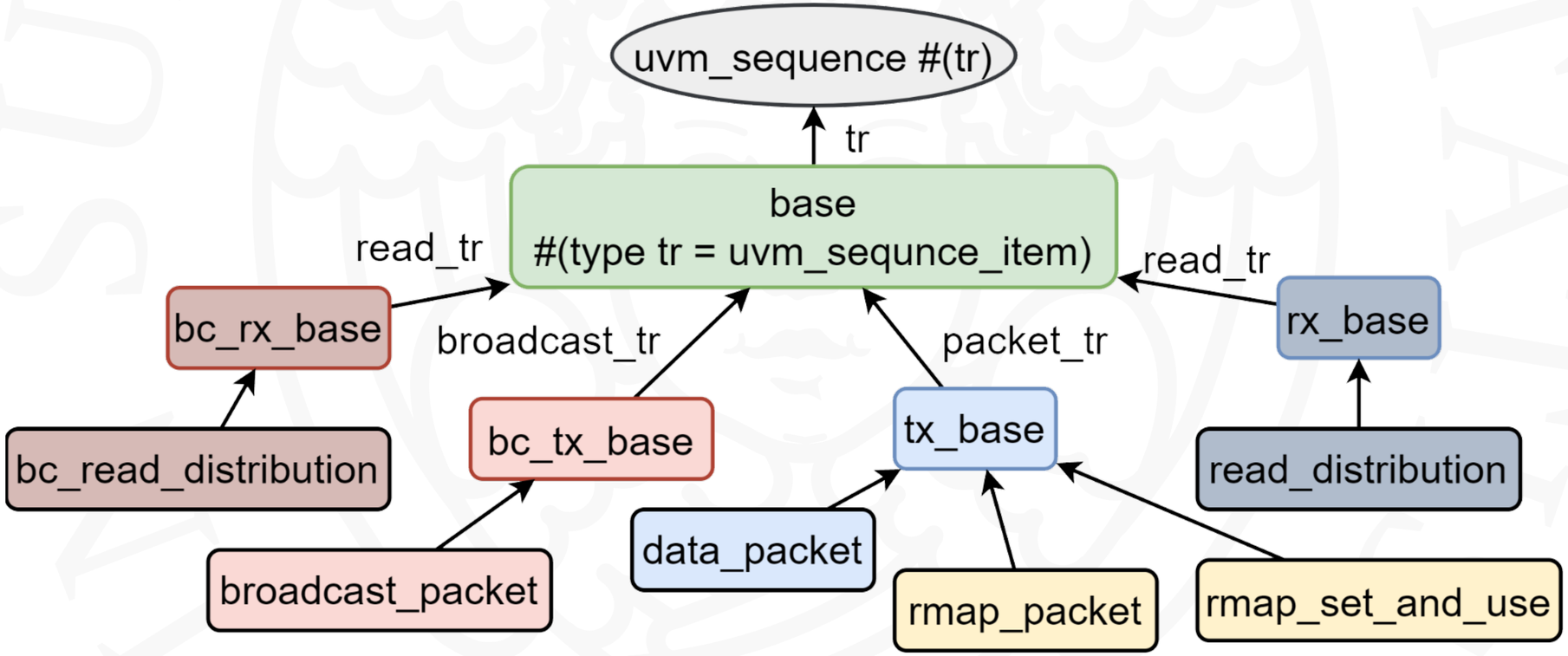
These knobs permit to test:

- Internal FIFOs and *Congestion*
- Virtual Channel Timeouts



SpFi UVM Virtual Sequences

What can the tester do?



SpFi UVM Virtual Sequences - Transactions Randomization

Data Packet

- Number of FILL in head
- Addressing Scheme
- Cargo content

Broadcast Packet

- Type: broadcast or timecode (SpW)
- BC Channel
- Message content

RMAP Packet

- Type: Read, Write, R-Modify-W
- Destination Register
  - Generates two RMAP for those having “use” reg
- Value
- Reply Address

All Sequences

- Invalid Format for testing purposes
- Valid & Ready assertion probability
- Source <Port, VC> tuples

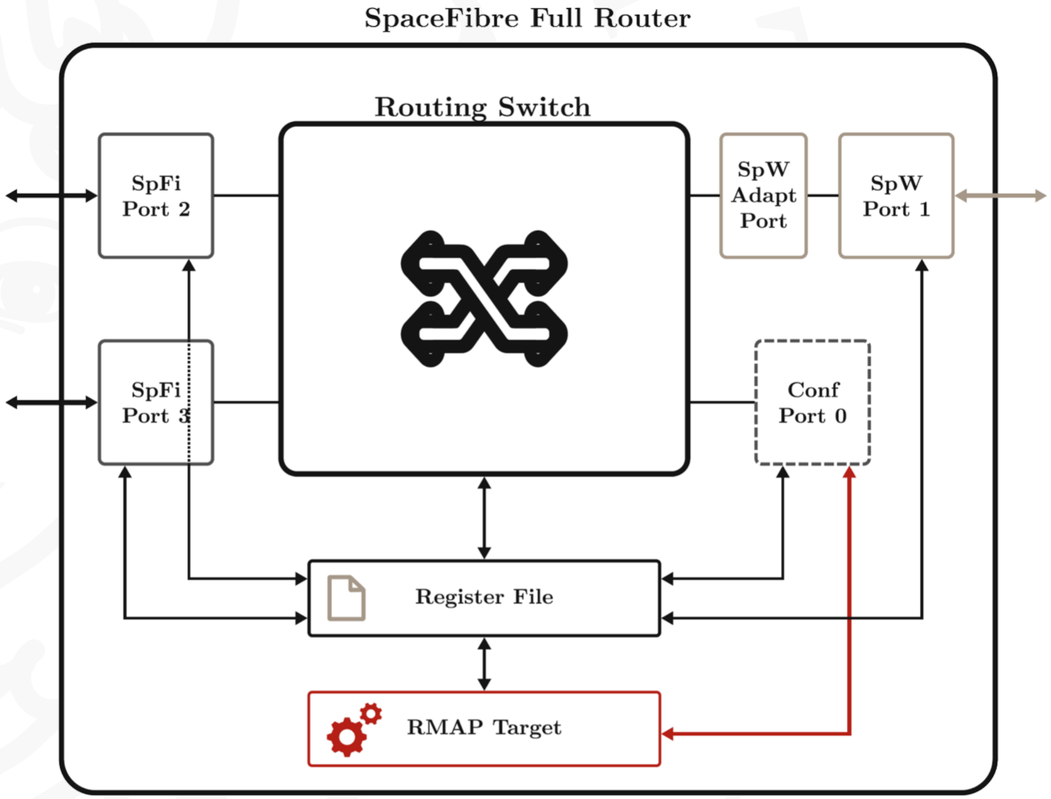
Outline

1. Introduction
   - SpaceFibre Standard and Network Layer
   - SotA Verification Methodologies
2. UVM-based Verification IP for SpaceFibre Routers
   - UVM Environment, DUT Interface, and SpFi Virtual Sequences
   - Testing Capabilities
3. Results on *IngeniArs SpFi Routing Switch IP Core*
4. Conclusions

UVM Verification Environment for SpaceFibre Router

IngeniArs SpaceFibre Router IP Core

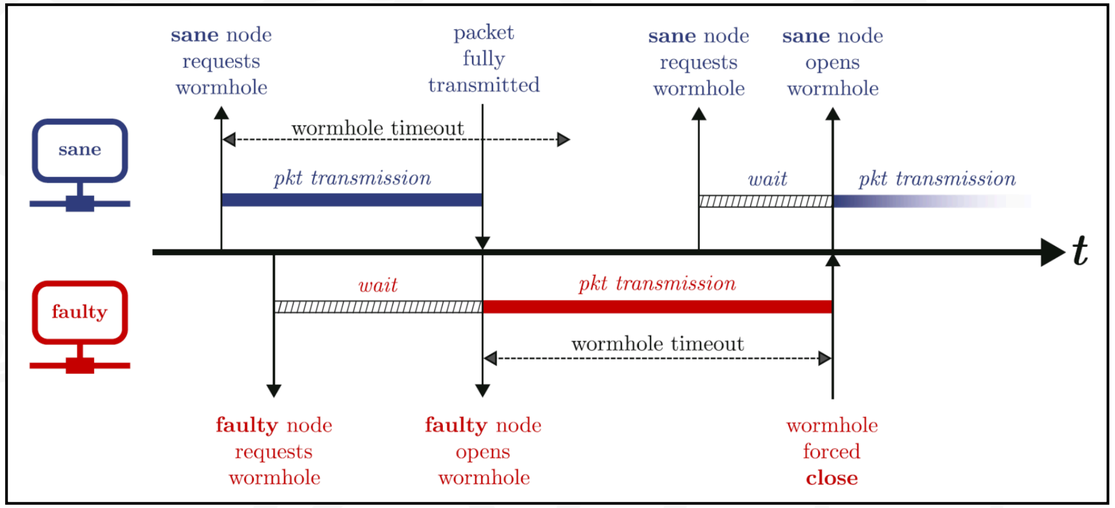
- Fully compliant with the SpaceFibre standard **ECSS-E-ST-50-11C**
- Supports all Network-Layer features
  - Multicast
  - Group Adaptive Routing
  - VC Timeout



IngeniArs SpaceFibre Router IP Core - Specific Features

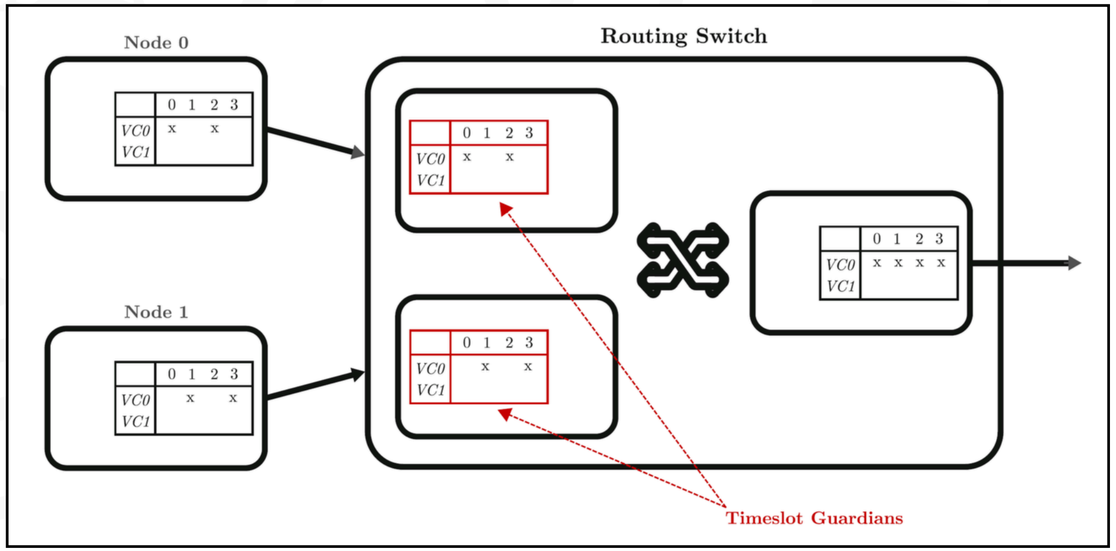
- Wormhole Timeout

  The maximum time a wormhole for a specific <Port, VC> tuple can last.



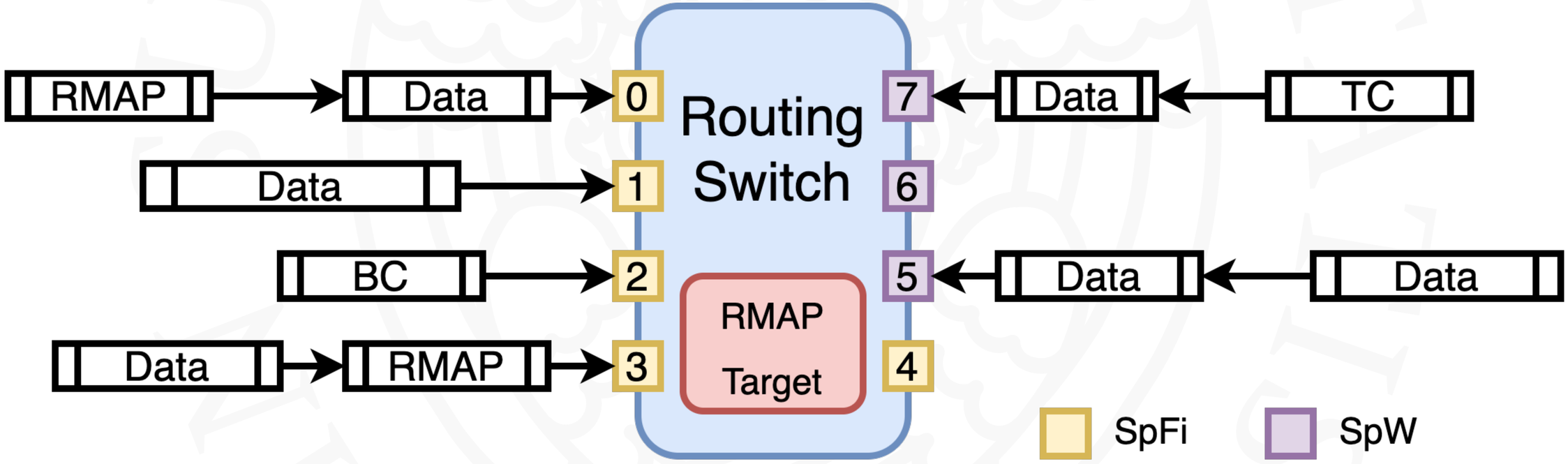
- Timeslot Guardian

  A network-level FDIR mechanism checks that packets finish within the assigned timeslot and closes the related wormholes otherwise.



Any Scenario is Configurable

It lays the foundations for a full functional coverage test plan!



Outline

1. Introduction
   - SpaceFibre Standard and Network Layer
   - SotA Verification Methodologies
2. UVM-based Verification IP for SpaceFibre Routers
   - UVM Environment, DUT Interface, and SpFi Virtual Sequences
   - Testing Capabilities
3. Results on *IngeniArs SpFi Routing Switch IP Core*
4. Conclusions

Conclusions

We presented the first fully compliant **SpFi Router Verification IP** for *Network-Layer* available in the literature.

Main features:

- **High-Level Sequences:** any test scenarios can be easily setup.
- **Re-usable** *SpFi UVM Components* in system-level designs with multiple SpFi routers and SpFi/SpW codecs.

Thank you for your attention

Luca Zulberti
luca.zulberti@phd.unipi.it
PhD Student
Department of Information Engineering
University of Pisa