5GtangO

A Prototyping Platform to Validate and Verify Network Service Header-based Service Chains

Manuel Peuster, **Stefan Schneider**, Frédéric Tobias Christ
and Holger Karl
Paderborn University
stefan.schneider@upb.de

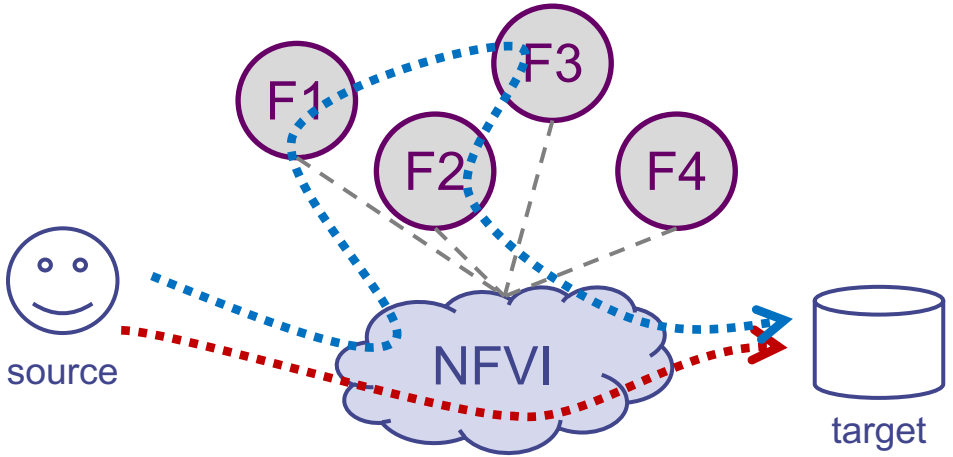
IEEE NFV-SDN 2018, Verona, Italy

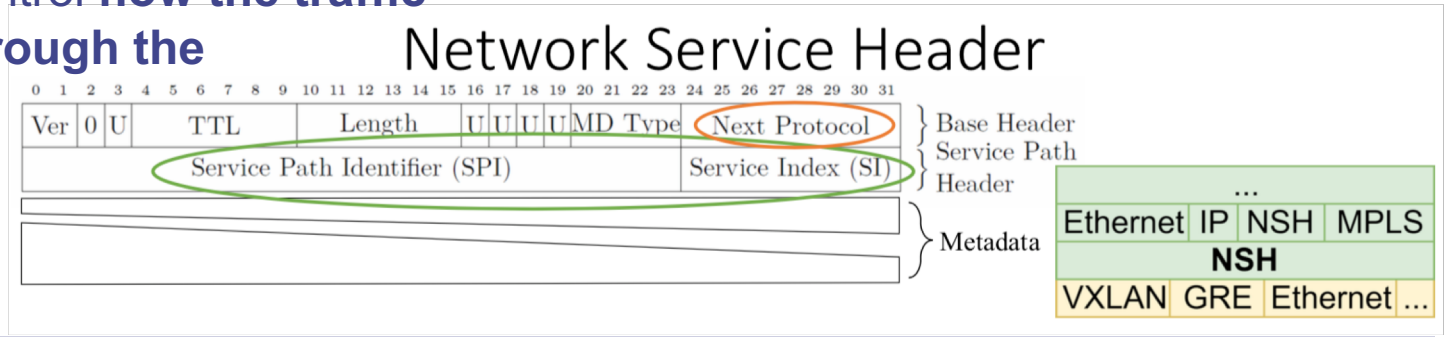
PADERBORN UNIVERSITY
The University for the Information Society

SFB901
ON-THE-FLY COMPUTING

Scenario: Chain functions to build a service

- NFV to **virtualize** network **functions between user and target** service
- All **traffic** passes **through** those **functions**
- But how to control **how the traffic is steered through the functions?**

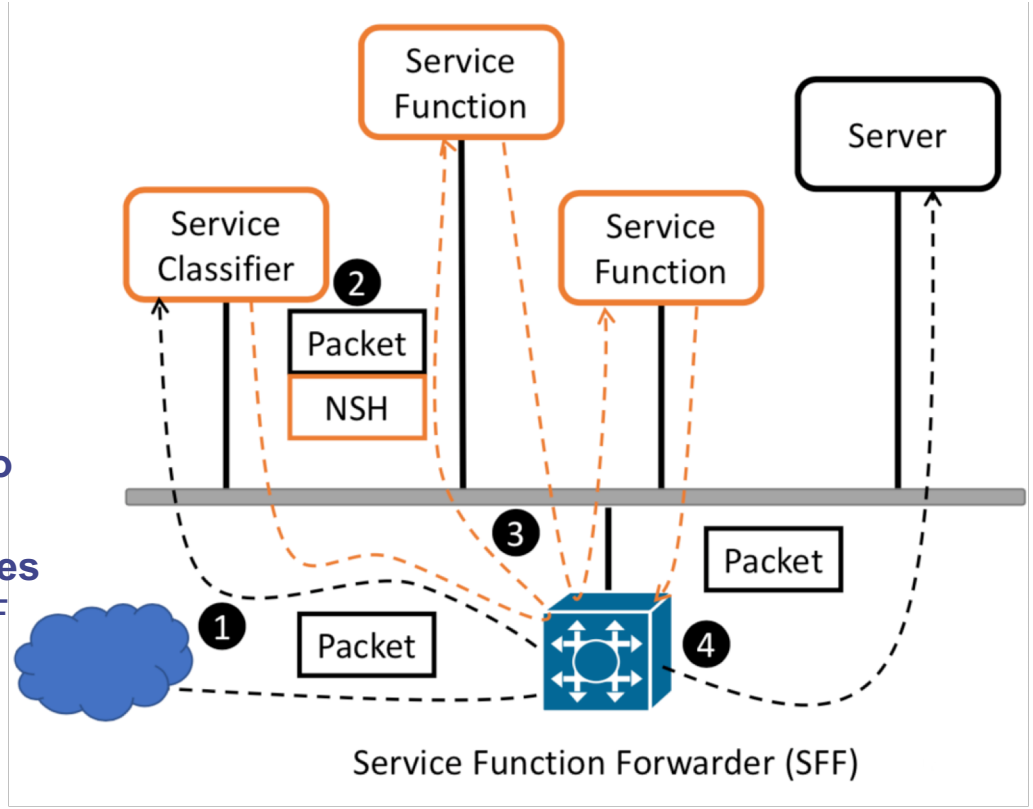




IETF SFC Architecture and NSH

- Components
  - Service function forwarder (SFF)
    - e.g., SDN switch
  - Service function (SF)
  - Service classifier

- Flow
  1. SFF sends (all) **incoming traffic to classifier**
  2. classifier selects SFC, **encapsulates** packet using **NSH**, forwards to SFF
  3. SFF **forwards according to NSH**
  4. Last SFF **removes NSH**

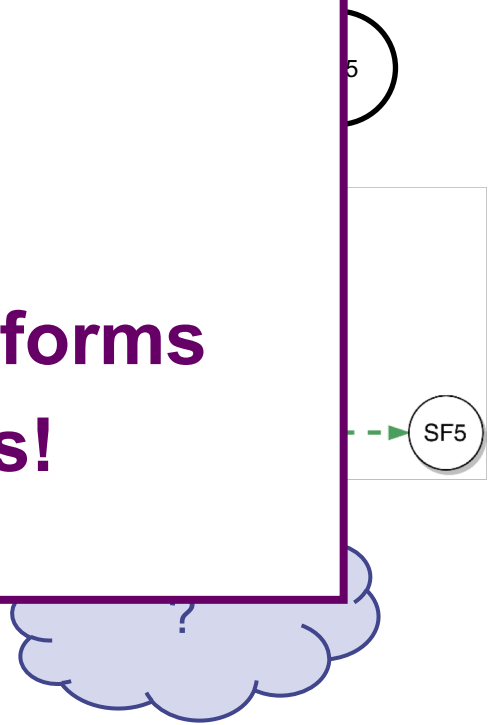


Problem: Prototyping support

- I am an NS developer
- I have a s
- I want to r
- I want to c
- I want mul
- I want to (
  e.g., corre
  correctly r

What tools to use?

Lack of prototyping platforms for NSH-based SFCs!



What would we need from a prototyping platform for NSH?

1. **Quick deployment** and execution of **arbitrary SFs**

2. Support developer-defined, **complex topologies**

3. Realistic traffic transport and **correct implementation of the forwarding paths**

4. **Seamless integration** with NFV landscape, e.g., **MANO solutions**

Related Work

- Most research work is based on simulations
  → does not help for prototyping of real SFCs

- Full-featured, real-world testbeds
  → prototyping is often time intensive

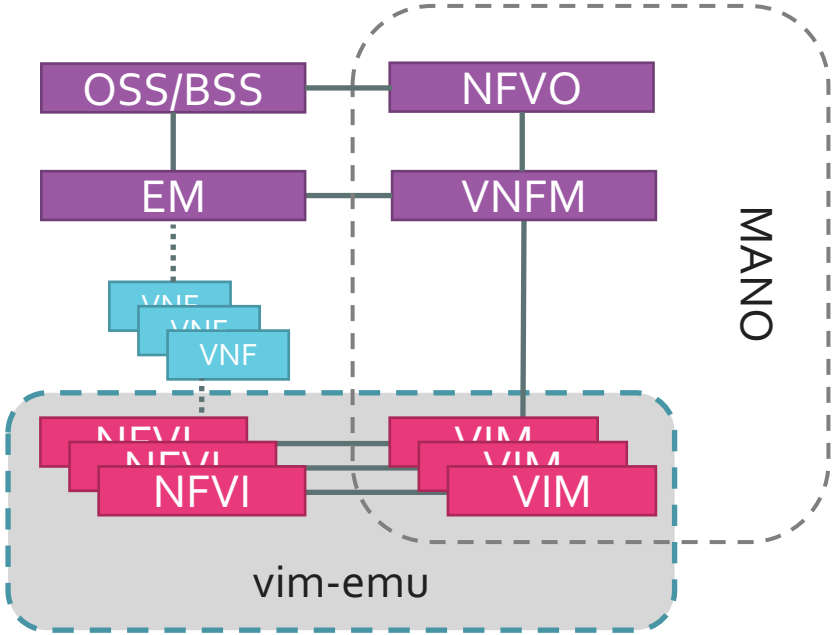
Proposed approach:

Extend vim-emu to support NSH-based chaining

- Emulation
  → looks promising: Lightweight, fast, less resource needs, can execute real SFs
  → but existing emulators, e.g., Mininet, VLSP, or vim-emu do not support NSH

vim-emu?

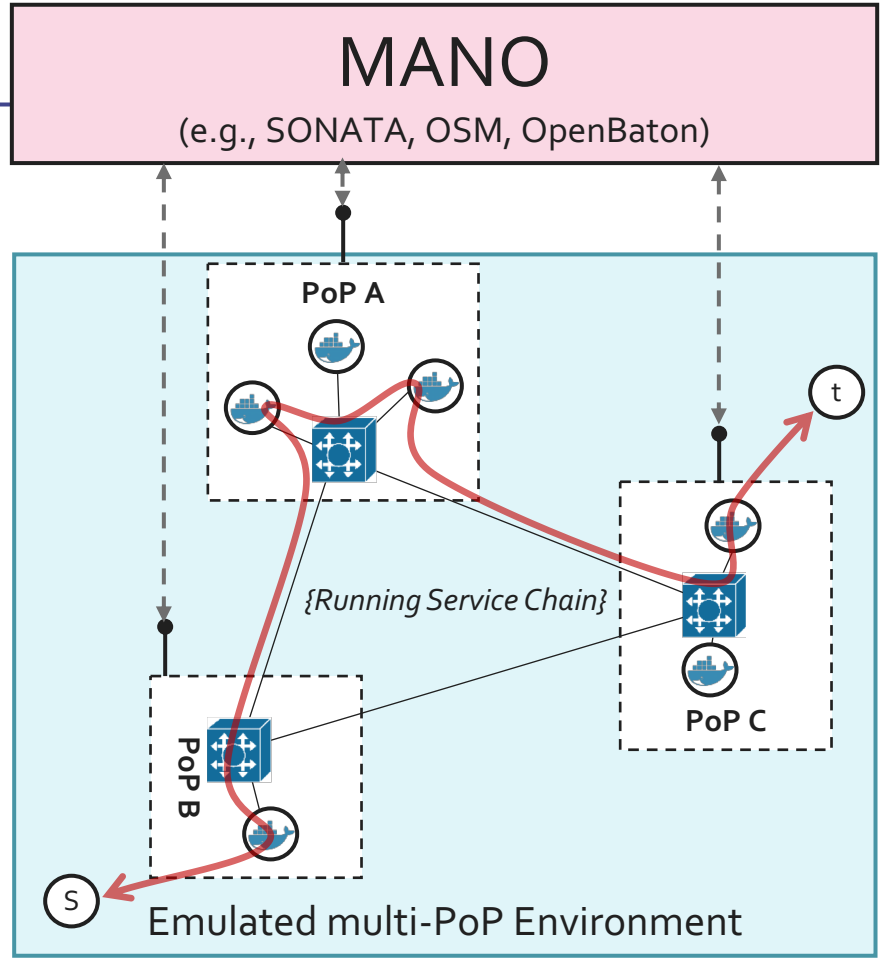
- Vision: Create an easy-to-use and easy-to-deploy **NFV test platform**
- Focus: Test VNFs and service chains **locally** on developer's machine
- Main idea: Emulate **multi-PoP NFVI** Infrastructure
- But: Interface with **real MANO** systems



Scope of the Emulation Platform in a simplified ETSI framework

How does vim-emu work?

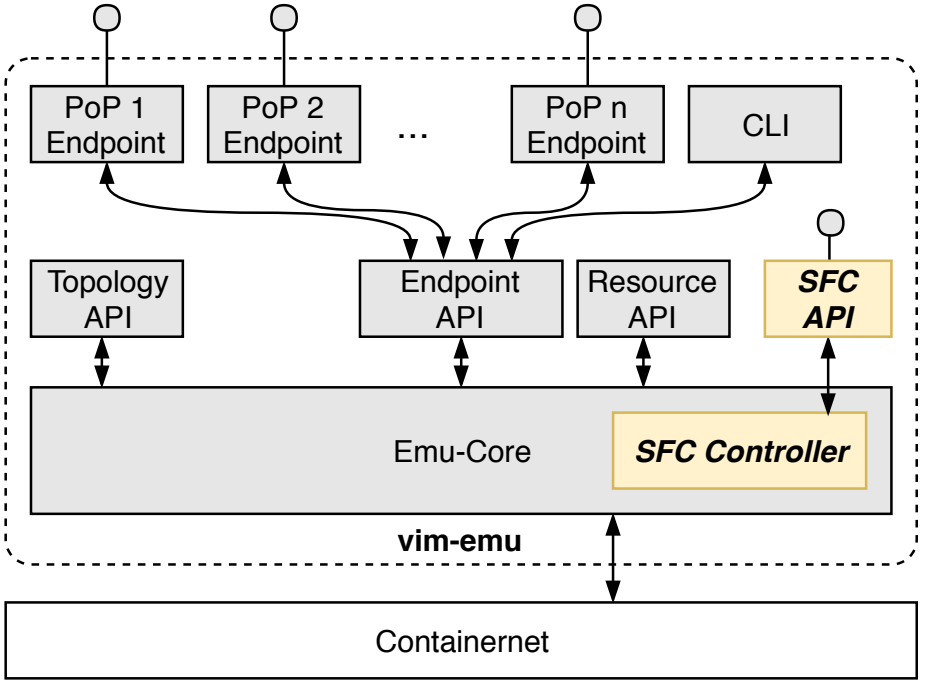
- What is emulated?
  - NFVI PoPs
  - MANO systems can interact with each individual PoP, e.g., start a SF
  - SFs are Docker containers (not VMs)
  - Containers can contain any SF software
- Environment:
  - Mininet- / Containernet-based
  - User-defined topologies
  - Each PoP offers its own VIM-like interface to deploy/manage SFs
  - Executed on single physical or virtual machine



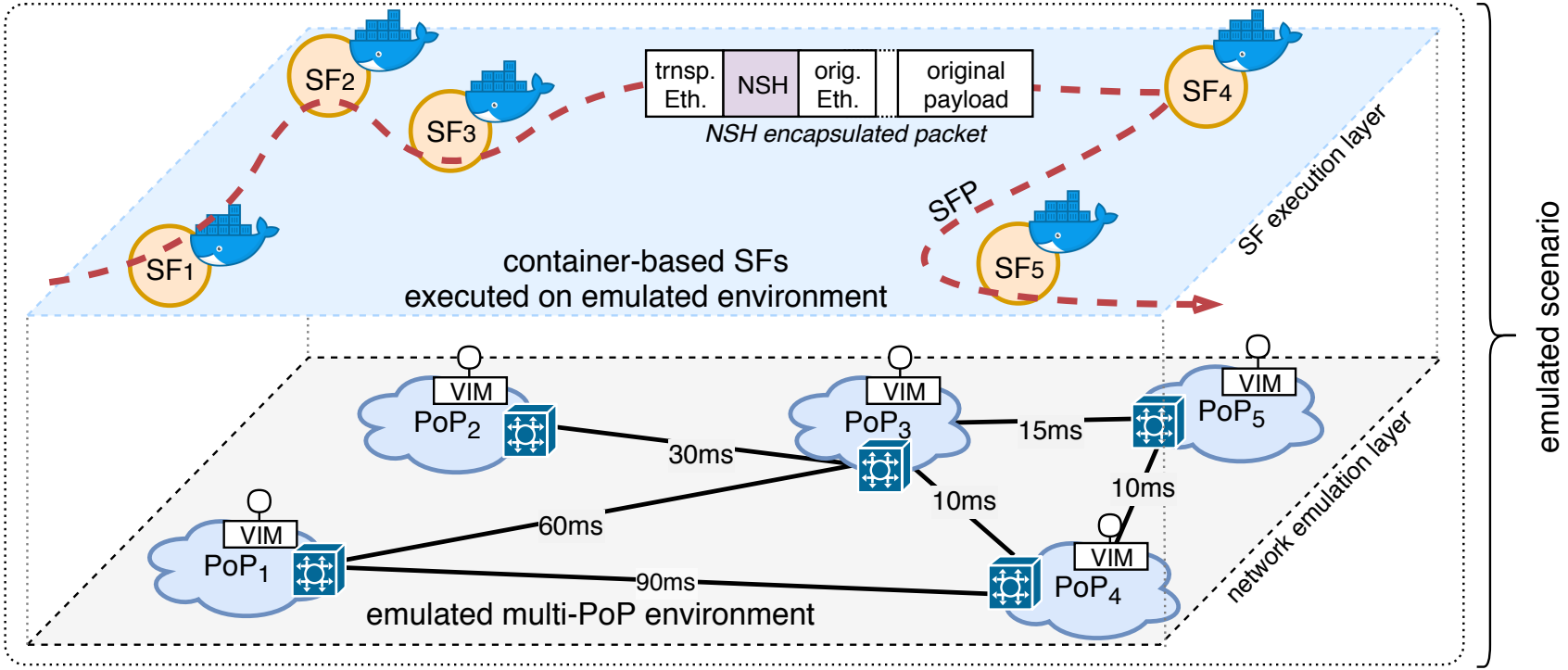
Example: Three emulated NFVI PoPs

Adding NSH support to vim-emu

- SFC API
  - Create, update, delete forwarding paths
  - Compatible with OpenStack Neutron SFC API
- SFC Controller
  - Compiles SFC configuration to OpenFlow table entries
  - Deploys table entries on involved OVS
  - OVS supports NSH (from v2.9)
  - Prototype build on top of Ryu



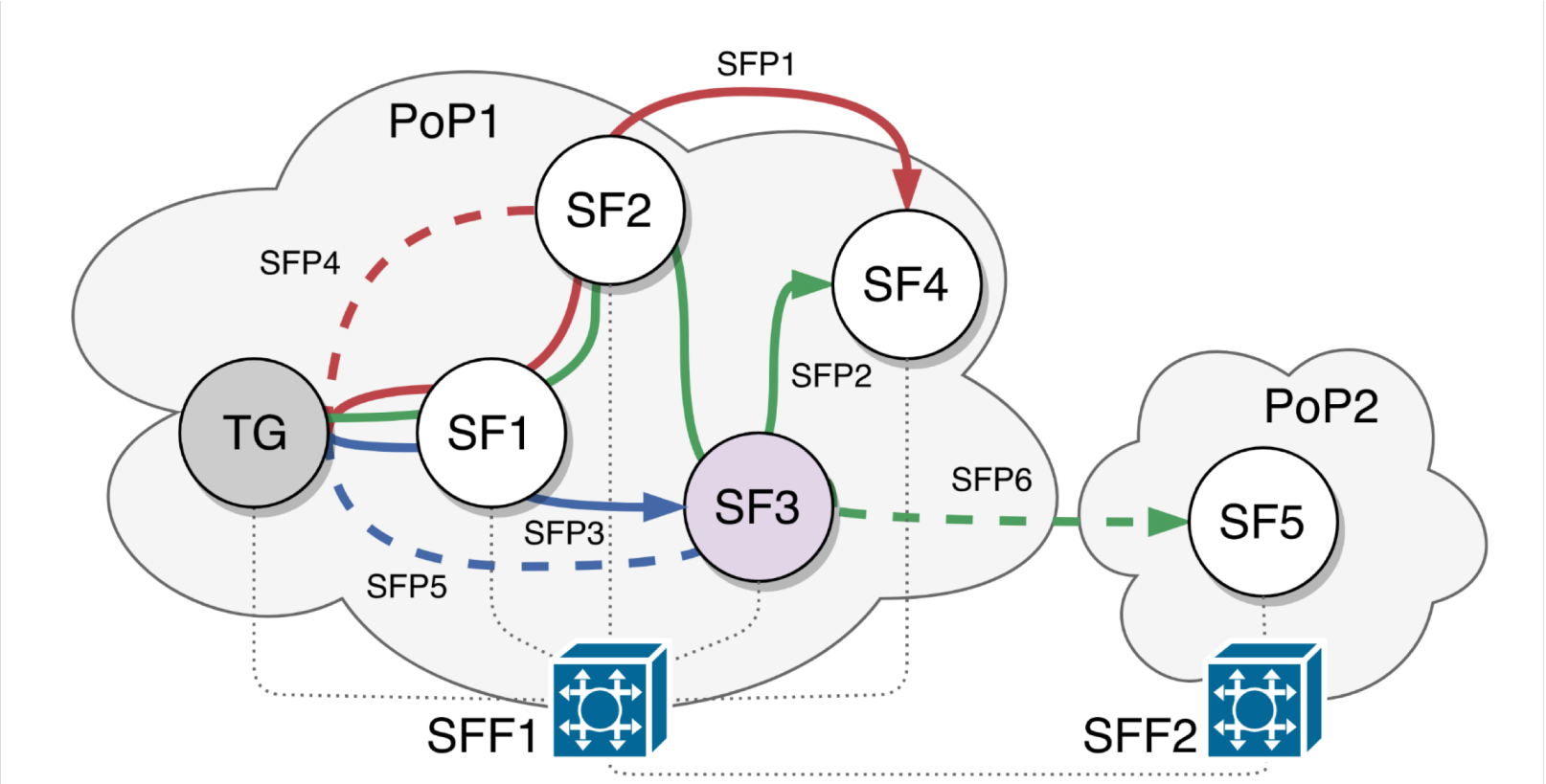
Running vim-emu emulation with NSH forwarding



Does it work?

Evaluation (qualitative)

Evaluation: Scenario



Evaluation: Generated traffic

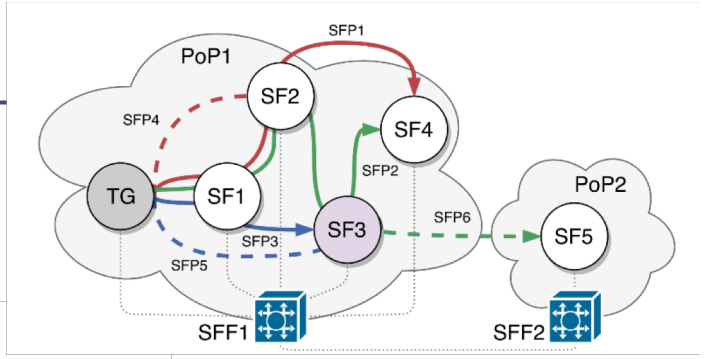
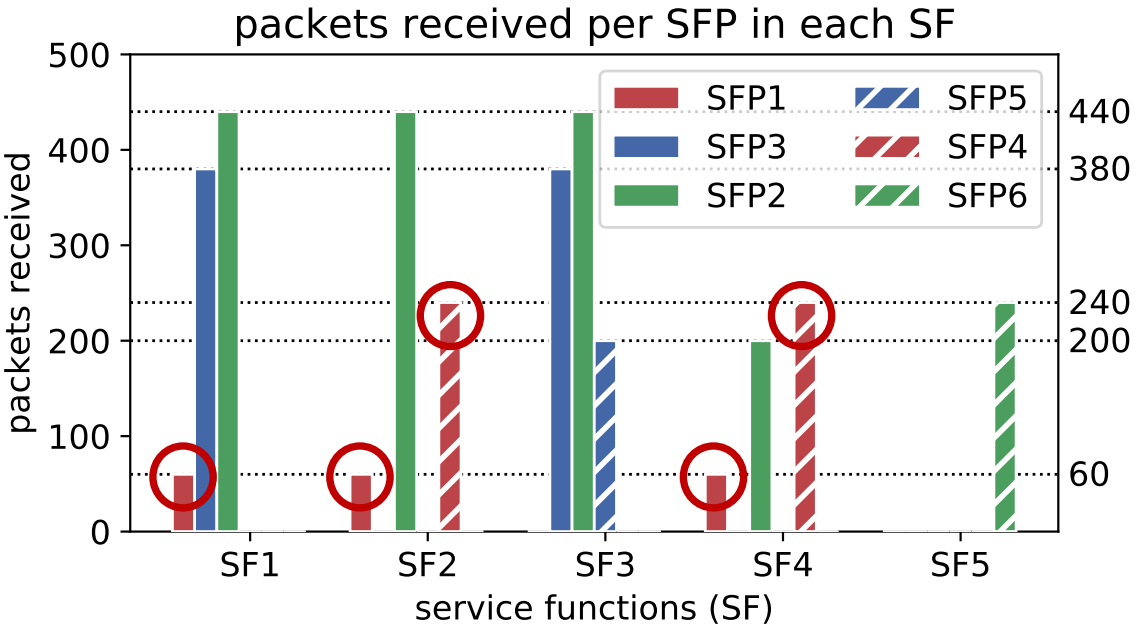
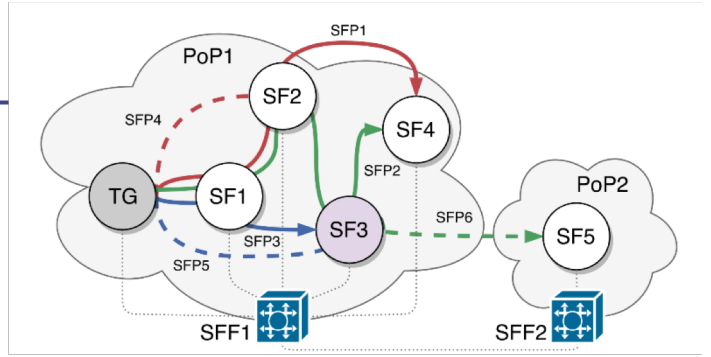


TABLE I: Generated traffic per SFP

color	path id.	rate	packets sent	$\sum$ packets sent
red	SFP1	8 pkt/s	60	300
	SFP4	8 pkt/s	240	
blue	SFP3	24 pkt/s	380	580
	SFP5	24 pkt/s	200	
green	SFP2	16 pkt/s	440	440
	SFP6	16 pkt/s	0	

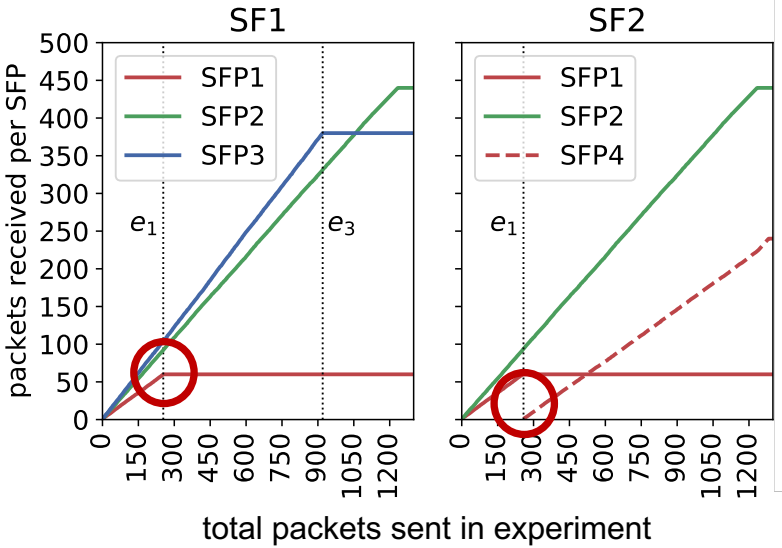
Results: Received traffic

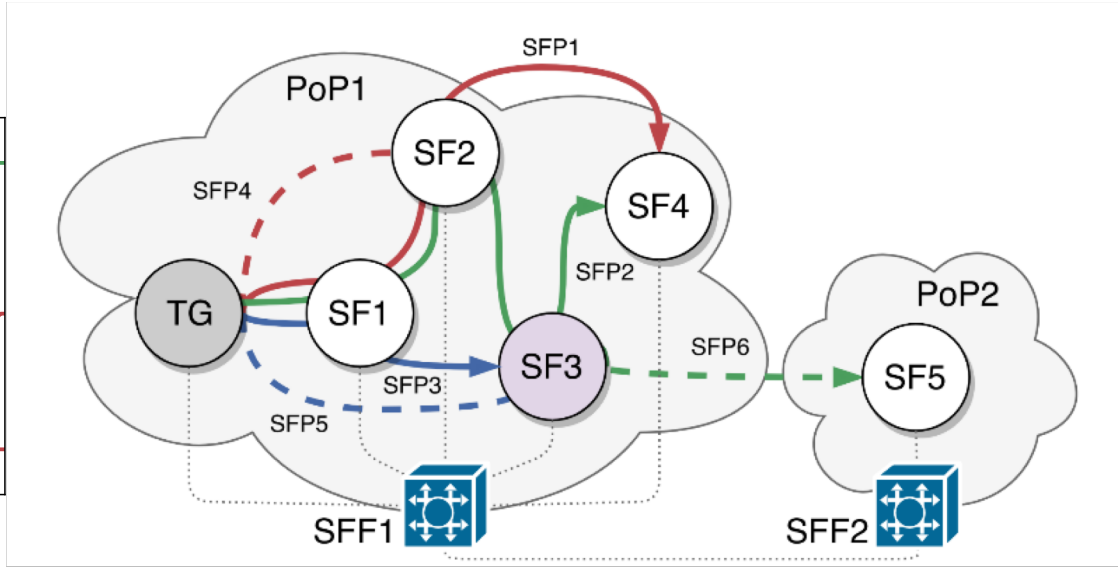




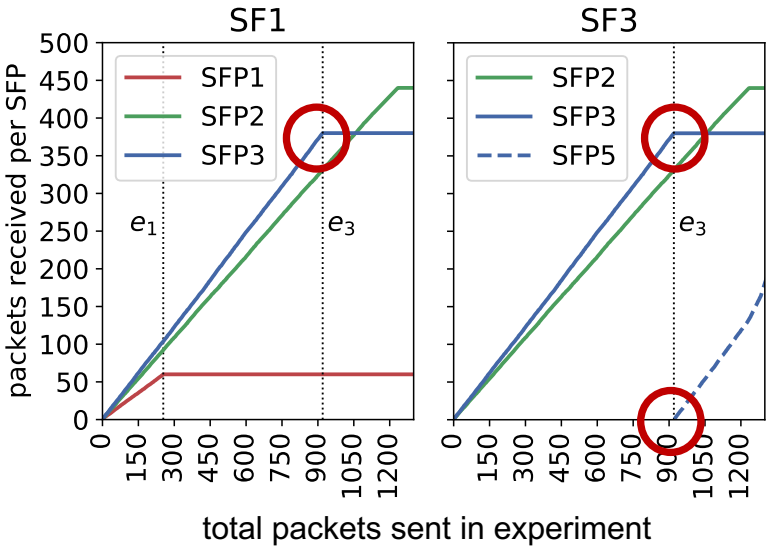
color	path id.	rate	packets sent
red	SFP1	8 pkt/s	60
	SFP4	8 pkt/s	240
blue	SFP3	24 pkt/s	380
	SFP5	24 pkt/s	200
green	SFP2	16 pkt/s	440
	SFP6	16 pkt/s	0

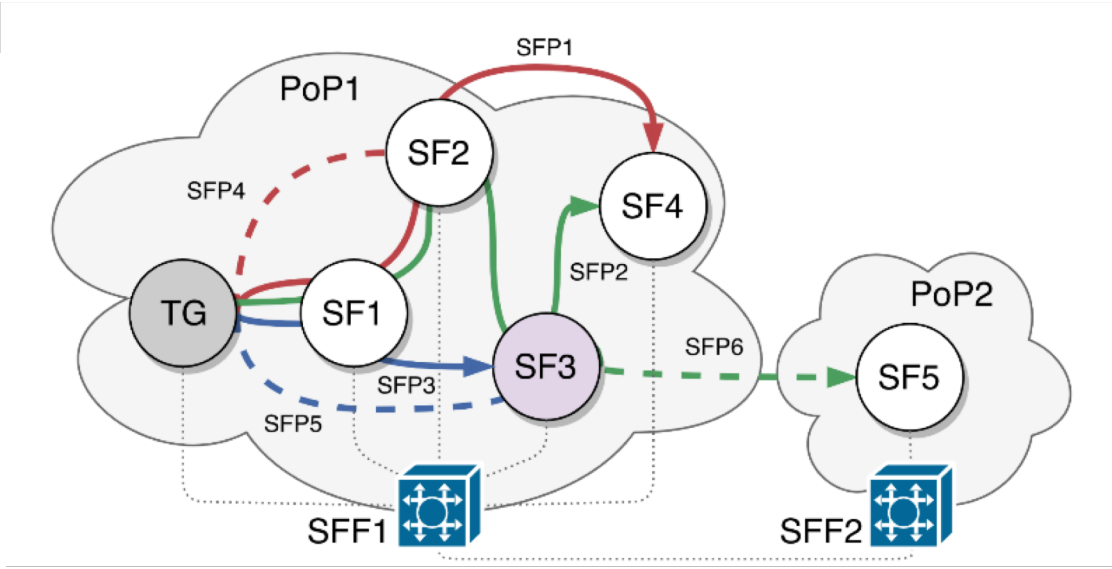
Results: Dynamic traffic steering



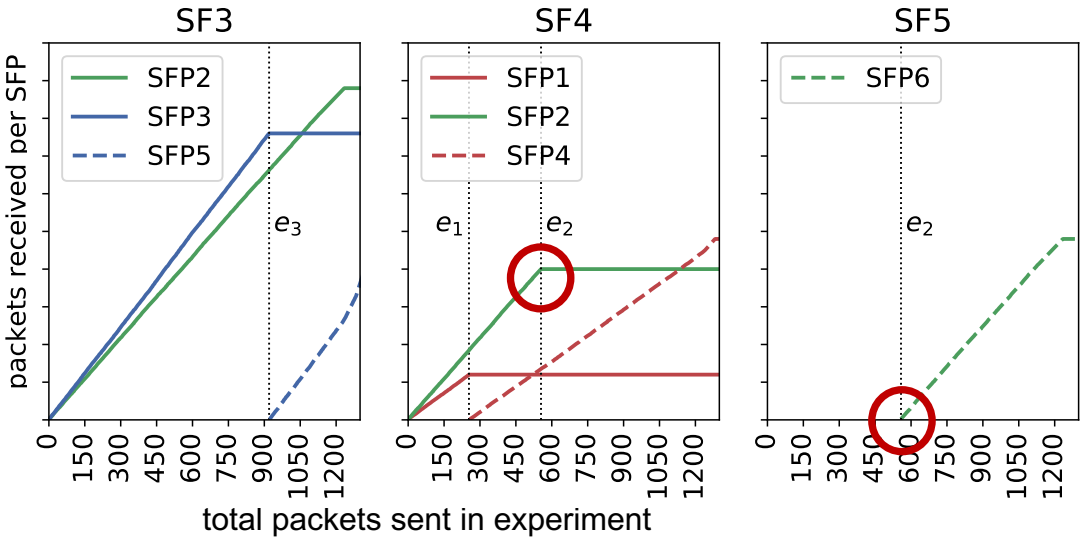


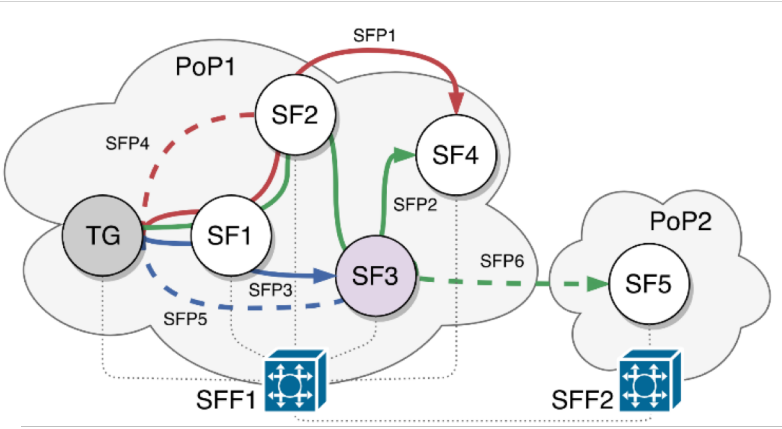
Results: Dynamic traffic steering





Results: Dynamic traffic steering





Conclusion and outlook

- NSH is one of the key enablers for wide adoption of SFC
- Our work enables researchers and developer to quickly prototype NSH-based SFCs in a local environment
- The presented solution is lightweight and can run on a developer's laptop
- Future work
  - There are other solutions, e.g., *segment routing*, which offer similar functionality as NSH
    - http://www.segment-routing.net/ and RFC8402
  - Extend vim-emu to support those alternative solutions

Thank you!

5GtangO

5G DEVELOPMENT AND VALIDATION PLATFORM FOR GLOBAL INDUSTRY-SPECIFIC NETWORK SERVICES AND APPS

www.5gtango.eu
@5gtango
http://lnked.in/5gtango

Prototype: **vim-emu with NSH support**

- Source (Apache 2.0): https://git.io/vim-emu-nsh

SFB901
ON - THE - FLY COMPUTING