Service Function Chaining Based on Segment Routing Using P4 and SR-IOV (P4-SFC)

Andreas Stockmayer, Stephan Hinselmann, Marco Häberle, Michael Menth
02.04.2020

http://kn.inf.uni-tuebingen.de
Outline

► Service Function Chaining

► P4-SFC
  ▪ Motivation & Idea
  ▪ Components
  ▪ Forwarding
  ▪ VNF Integration
  ▪ Traffic Classification
  ▪ Orchestration
Traffic of end-to-end services usually passes several network functions
- Firewall, NAT, application gateways, …

Traditional: Network functions are „hard-wired“
- Static path of network functions
- Problem: Not very flexible

Alternative: Service Function Chaining
- Traffic is classified
- Further processing depends on classification
Service Function Chaining (II)

Components
- Classifier
- Service Function Forwarder
- Service Functions

Forwarding
- Segment Routing (SRv6, MPLS, …)
- Network Service Header (NSH)

RFC 8595
P4-SFC: Motivation & Idea

► Motivation
  ▪ SFC classifier either limited performance and features or expensive
  ▪ SFC only practical for large operators

► Combination of SFC classification, network management und VNF orchestration
  ▪ SFC classification in P4
  ▪ Forwarding using MPLS label stacks
  ▪ VNFs using libvirt and lxc

► Combination of SDN and legacy
  ▪ Classification using P4
  ▪ Forwarding using legacy switches
  ▪ ⇒ Cost effective
P4-SFC: Components

Ingress Switch → Forwarder → Orchestrator

VNF 1 → VNF 2 → VNF 3 → ...
VNF 11 → VNF 12 → VNF 13 → ...
VNF 21 → VNF 22 → VNF 23 → ...

- MPLS label stacks
- Similar to SR-MPLS (draft-ietf-spring-sr-service-programming-01)

1 label per service function and per link between switches
- Changing SFC requires changes in classifier only
- Forwarding can be done by any switch supporting MPLS
Example: Packet goes through VNF 1, 11 and 23
Label stack: 100, 101, 111, 123, 100
Ingress Switch

- Classification by 5-Tuple (src & dst IP, src & dst Port, protocol)
- Prototype
  - Barefoot Tofino
  - Up to 10 MPLS labels
    - More possible, requires jumbo frames
  - Line Speed
  - No recirculation
  - IPv4 and 10 MPLS labels: up to ≈100,000 rules

```
| Push_Label_Stack |
|----------------------------------|
| Match keys                       |
| Ternary                          |
| p.srcIP & p.dstIP                |
| p.protocol & p.srcPort & p.dstPort |

| Action | Parameters |
|--------|------------|
| push_LS_1 | - L1 |
|         | - ... |
|         | - ... |
| push_LS_n | - Ln |
| miss    | Standard IPv4/MPLS forwarding |
```
VNF Integration

- VNF either VM or container
- Deployed by orchestrator
- Each VNF addressed by MPLS Label
- Dedicated virtual function using SR-IOV per VNF
- Forwarding from forwarder to virtual function using VLAN
- MPLS Router Module in Linux kernel used as SFC proxy
Central controller

SFC definitions
- Administrators/customers define service function chains
- Orchestrator allocates resources and MPLS labels

Network management
- Configuration of ingress switch (P4Runtime)
- Configuration of forwarders

Deployment of VNFs
- VNF either VM or container
- VNFs distributed fairly on all SF nodes
  - Redistribution if necessary
Orchestrator: Prototype

- Python
- Configuration of SFCs as JSON file
- Network Management
  - Southbound interface to Tofino (classifier)
- NF Deployment
  - API to libvirt und lxc
  - VNF definition as binary on NFS share
    - Executed when VNF is started
    - e.g. script that configures VM
Conclusion

► Service function chaining using MPLS segment routing
► Combination of SDN and legacy
  ▪ Classification using P4
  ▪ Forwarding using legacy switches
► Orchestrator as central controller
  ▪ SFC definition
  ▪ Network management
  ▪ VNF deployment

► Fully featured but minimal system
► Cost-effective
Marco Häberle, MSc.
haeberle@informatik.uni-tuebingen.de
University of Tuebingen, Dept. of Computer Science
Chair of Communication Networks
Sand 13, 72076 Tuebingen, Germany
http://kn.inf.uni-tuebingen.de/