Politecnico di Torino

TCP Connection Management for Stateful Container Migration at the Network Edge

Yenchia Yu, Antonio Calagna,
Paolo Giaccone, Carla Fabiana Chiasserini

MedComNet2023

cnit

IEEE ComSoc
IEEE Communications Society

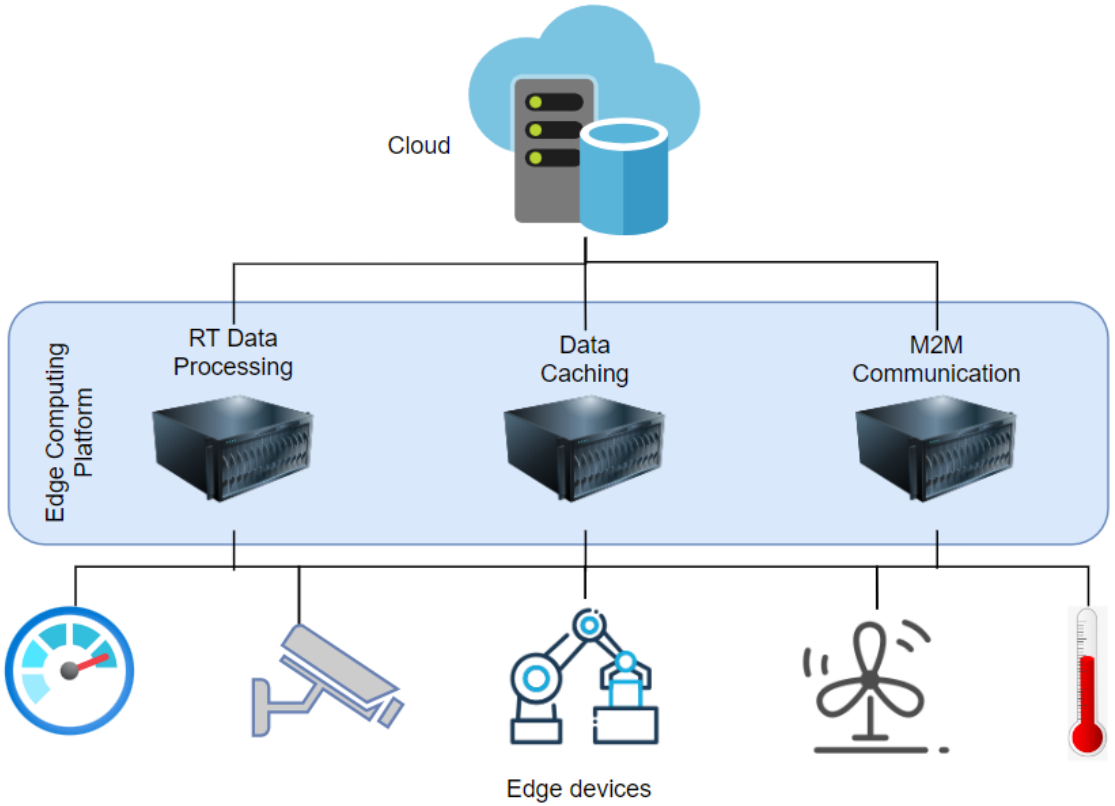
Introduction & Motivation

Edge computing:

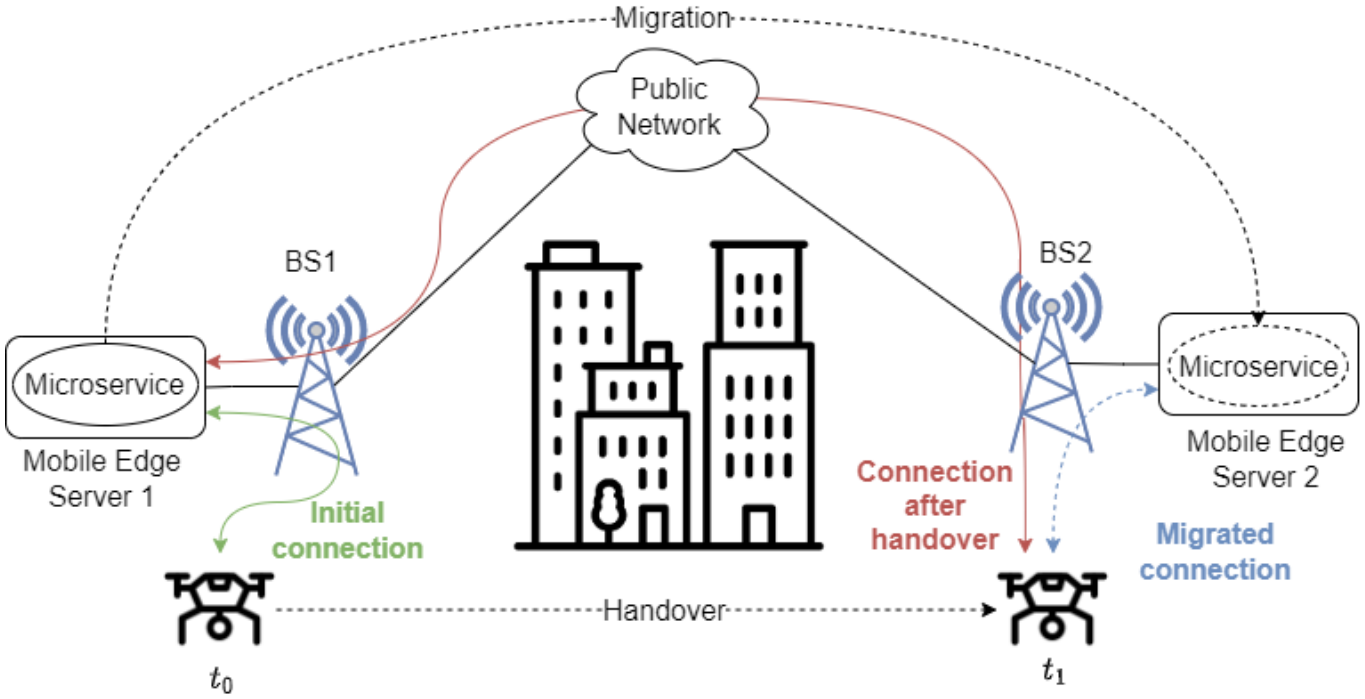
- Bring **computation and data storage** closer to the end user
- Enable **latency, bandwidth critical** services on infrastructure

Edge service consumer:

static devices - > **mobile devices**



Introduction & Motivation



Service Mobility requirement:

- Ensure proximity w.r.t the end user

Enable service mobility via:

- Stateful container migration
- Connection migration

Stateful Container Migration

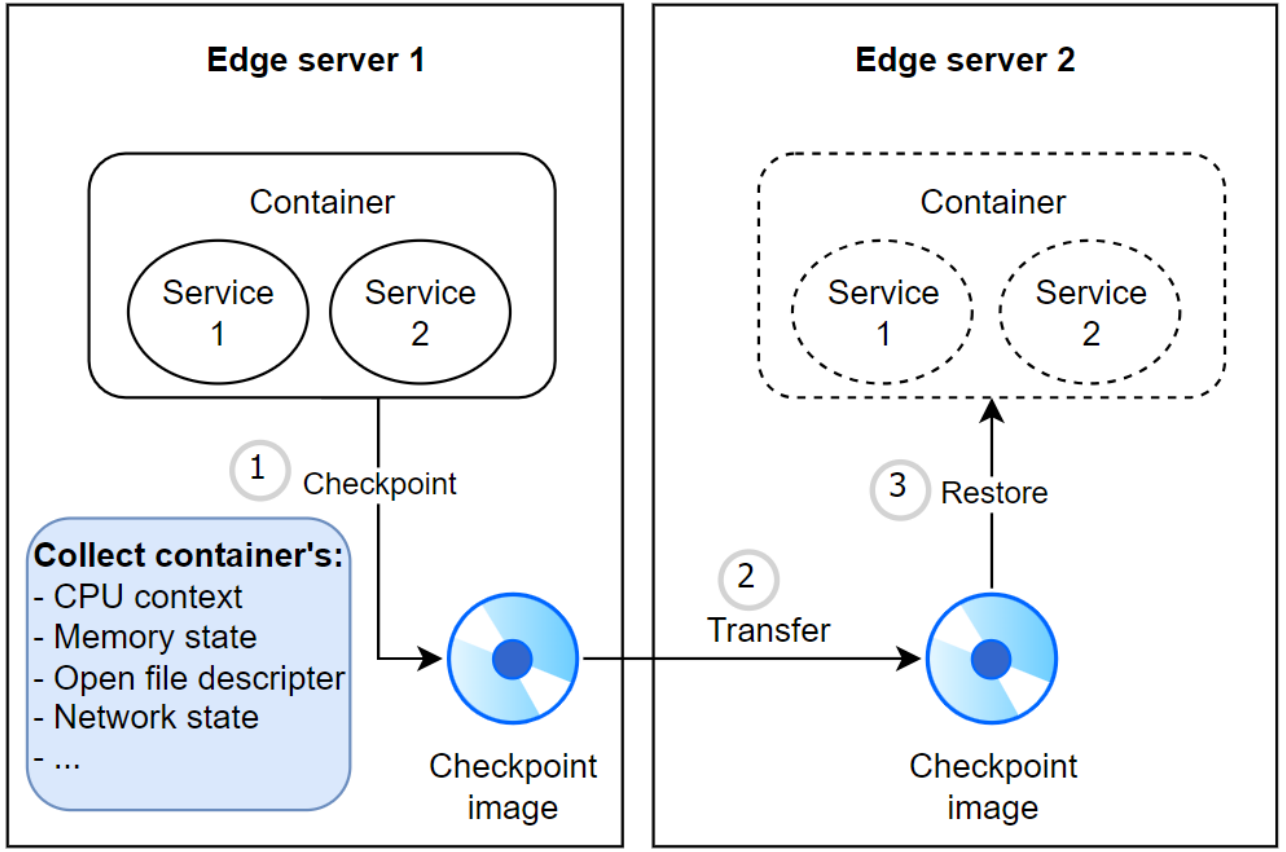
Container engines:

- Podman

Key tool for container migration:

- CRIU
  (Checkpoint/Restore In Userspace)

Connection migration
still needs to be addressed



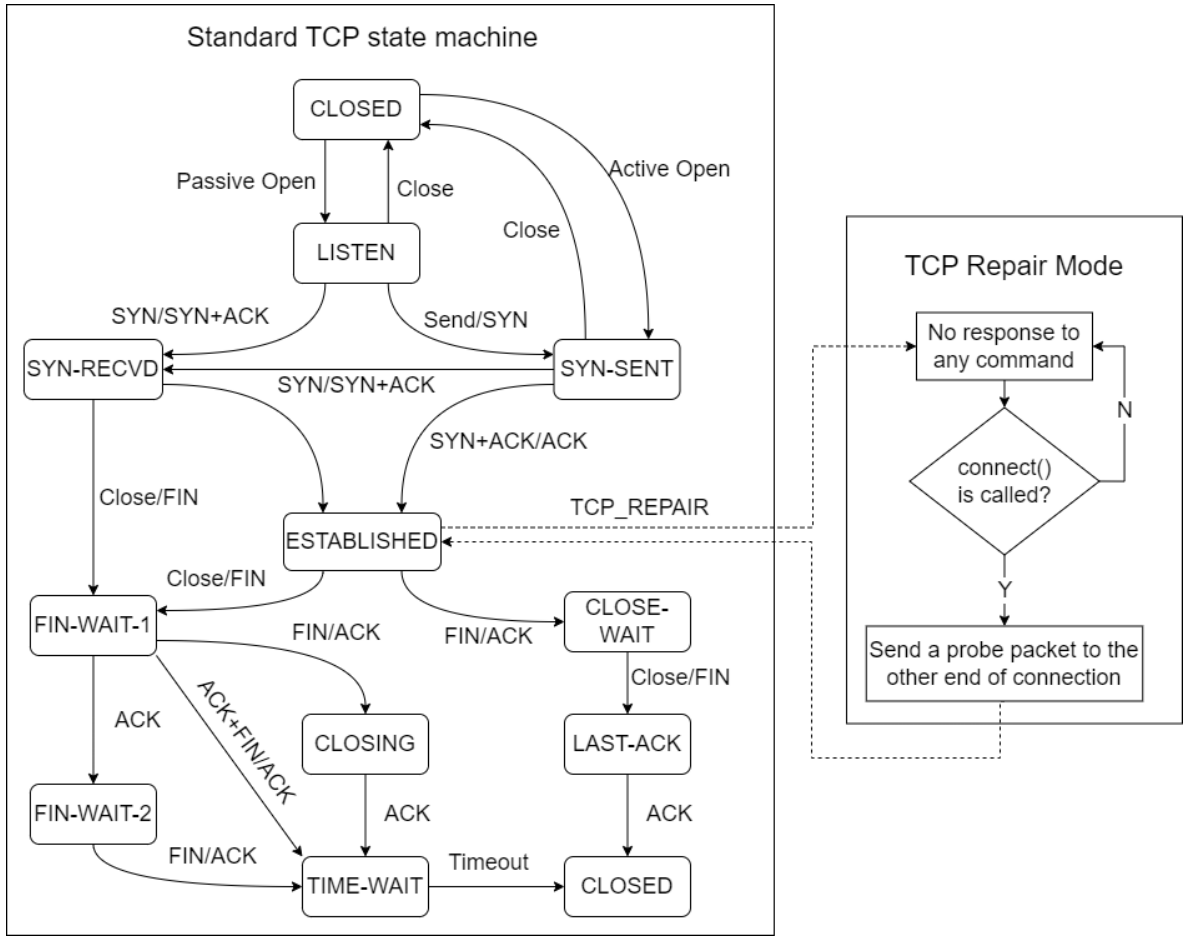
Connection Migration Challenge

- Connection mobility:
  - Protocols like UDP and QUIC support **client-side mobility**, but **not support service-side migration** unless customize the protocol
  - Protocol like LISP introduces **mobility function** in design, but such protocol is not widely adopted

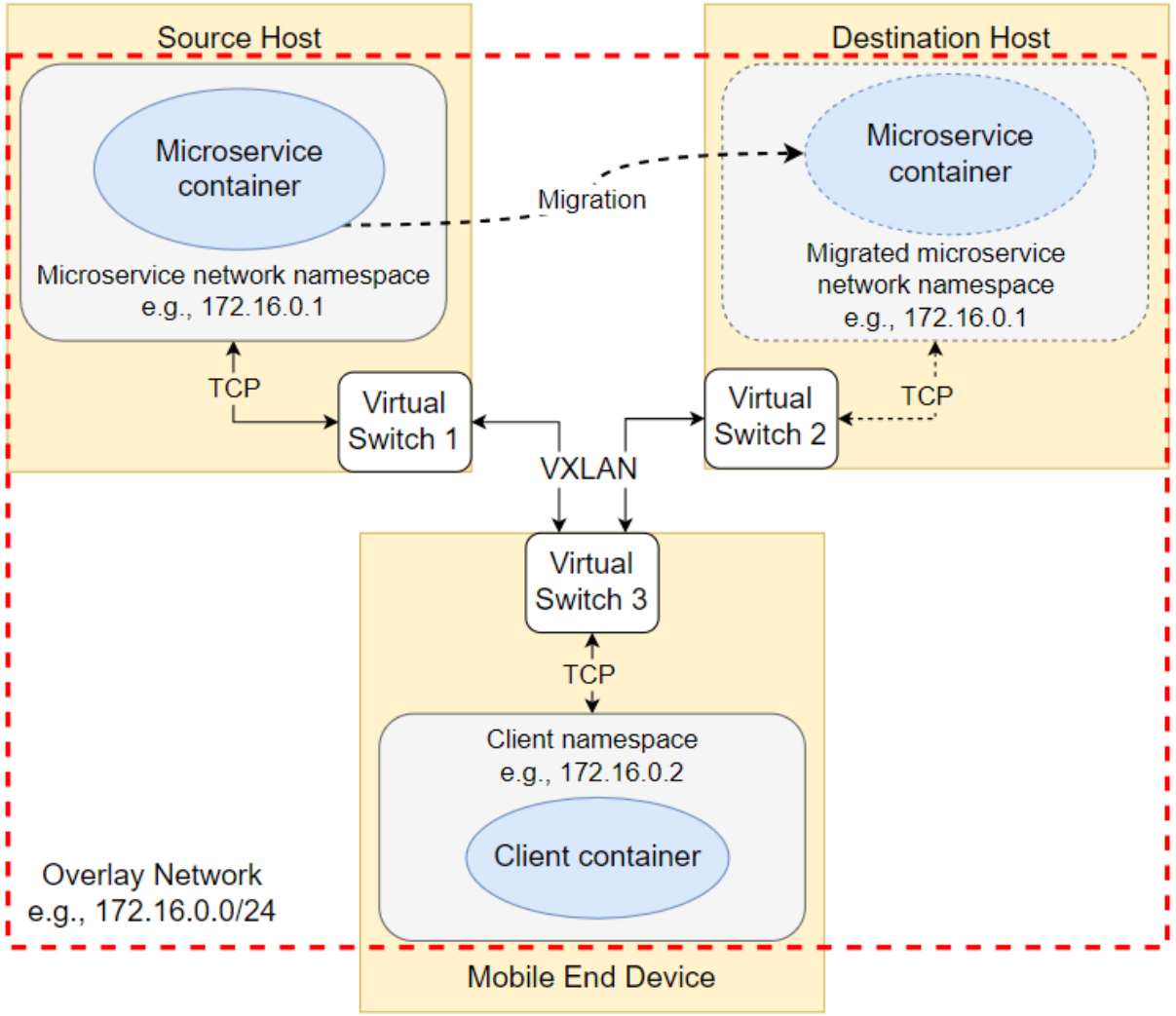
- We focus on **TCP Connection**:
  - Commonly used in legacy and modern edge application
  - Connection-oriented protocol
  - By default it does **not support mobility**

TCP_REPAIR option

- **TCP_REPAIR**: a special option for the TCP socket in Linux kernel from version 3.5 (in 2012)
- It can be leveraged to :
  - Collect TCP socket state information (checkpoint)
  - Inject connection state information to the socket and directly enter the “ESTABLISHED” connection state (restore)
- **Challenges:**
  - Address Consistency
  - Network Reachability



Our Solution

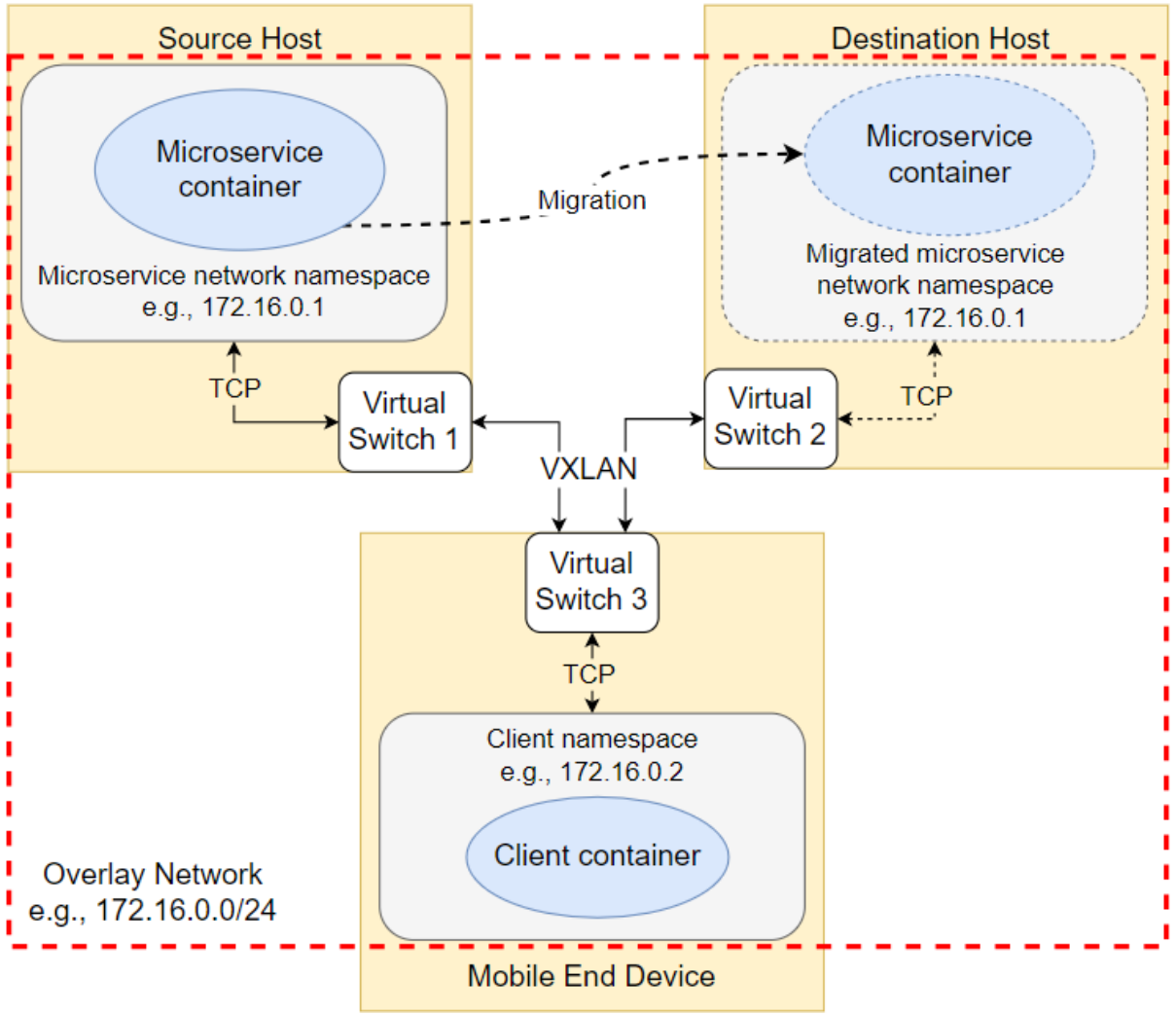


COAT:
Container OverlAy TCP architecture

Virtual Switch: **Open vSwitch**

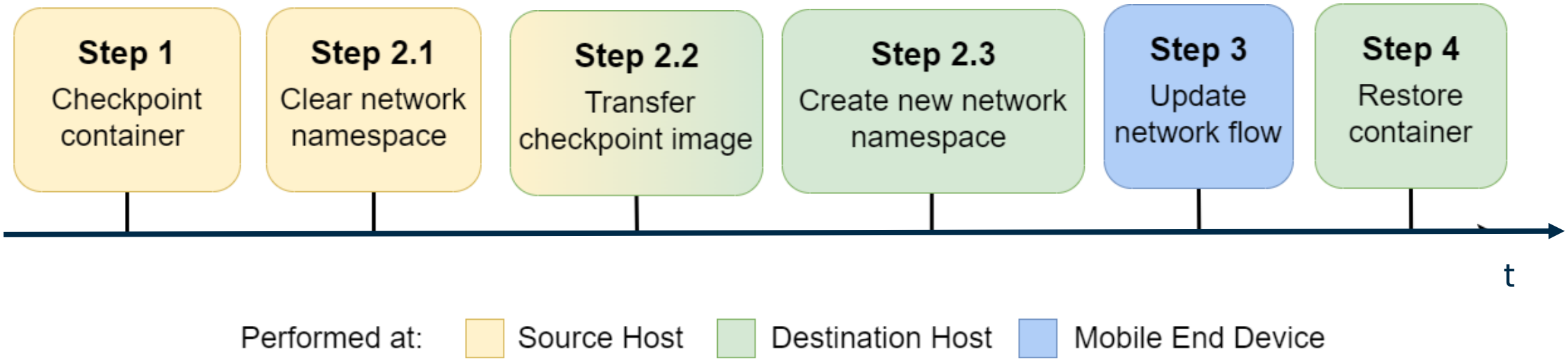
VXLAN protocol: Encapsulates OSI layer 2 Ethernet frames within layer 4 UDP datagrams

Our Solution

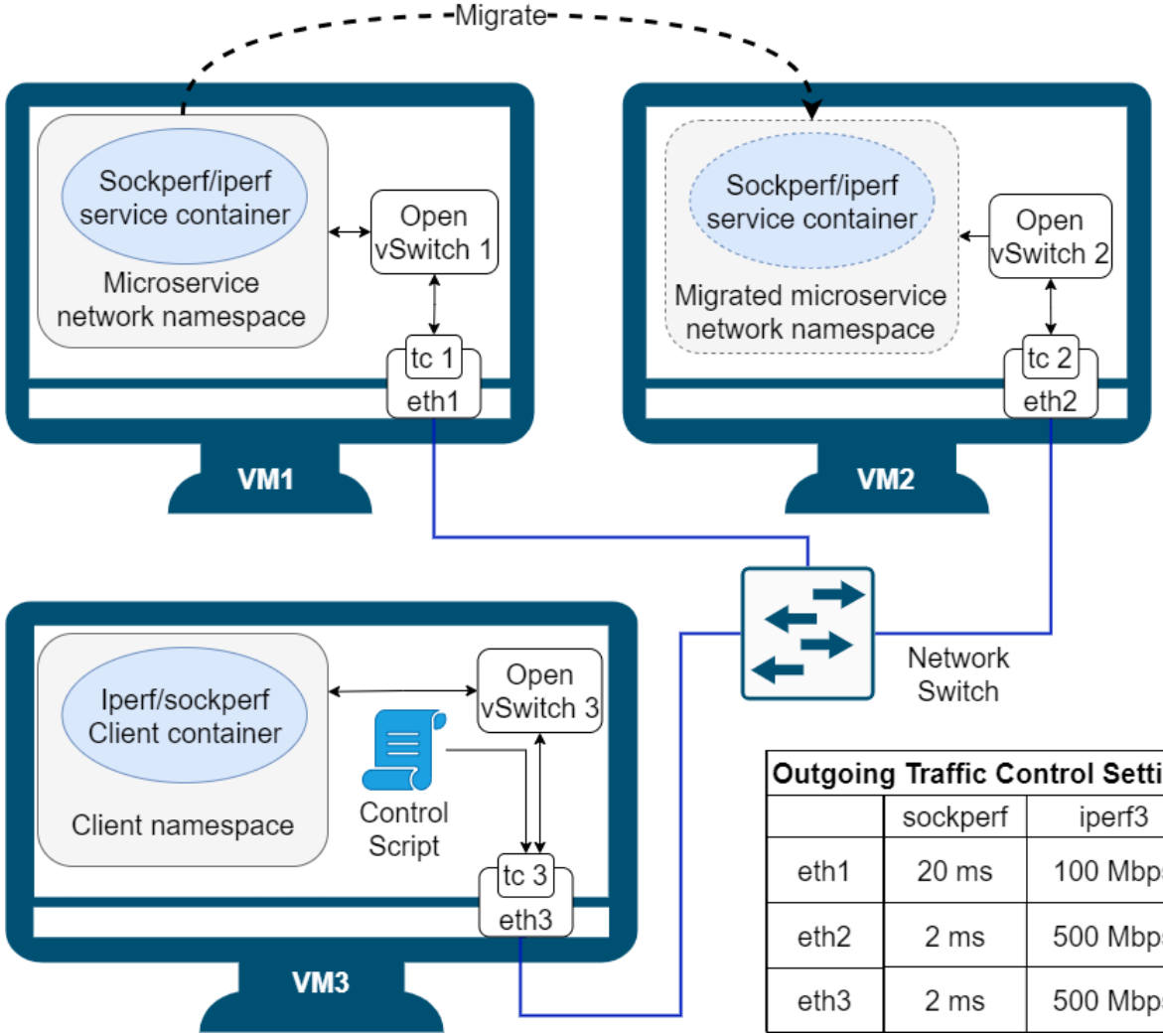


- The overlay network can be **dynamically updated**, yielding IP address consistency
- Containers binded to the same overlay network can **directly reach each other**

COAT Enhanced Microservice Migration



Testbed setup



Outgoing Traffic Control Setting		
	sockperf	iperf3
eth1	20 ms	100 Mbps
eth2	2 ms	500 Mbps
eth3	2 ms	500 Mbps

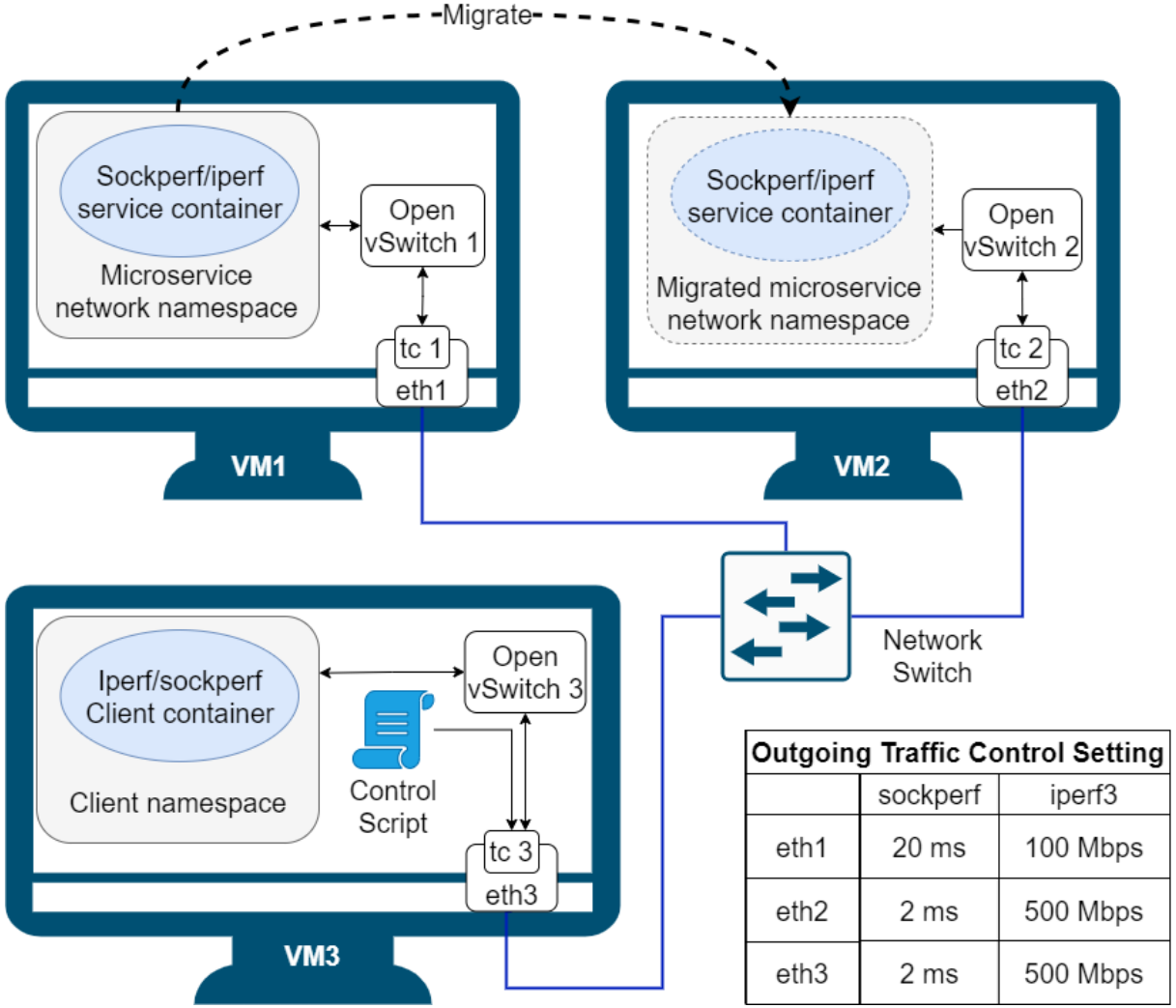
Outgoing traffic control :

- We assume VM2 is closer to VM3 w.r.t VM1
- We emulate a **realistic network** using “tc”

Control Script

- Sends the command of each COAT migration step to proper VM **via SSH tunnel**

Testbed setup



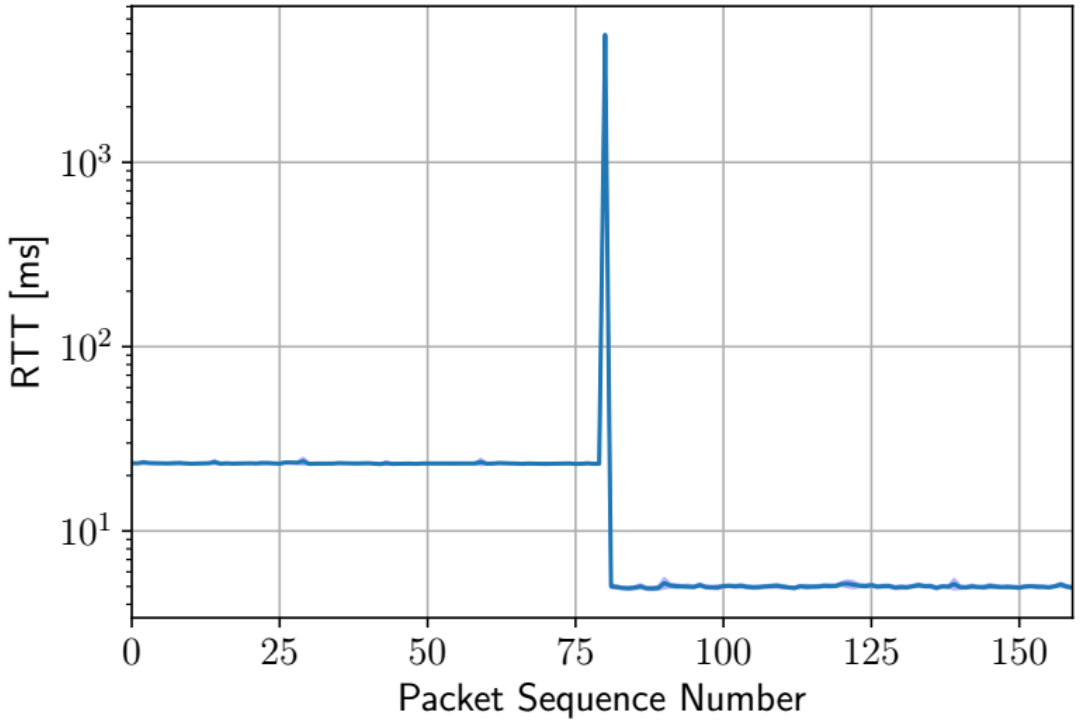
Outgoing Traffic Control Setting		
	sockperf	iperf3
eth1	20 ms	100 Mbps
eth2	2 ms	500 Mbps
eth3	2 ms	500 Mbps

Two independent service migration experiments:

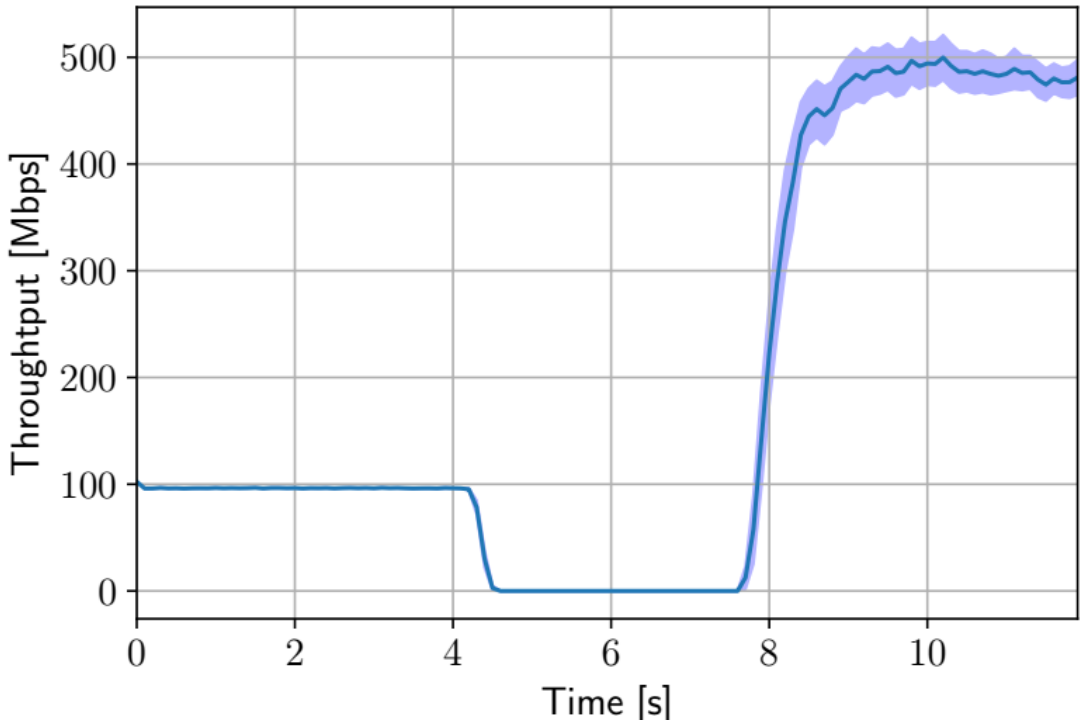
- **Sockperf** as microservice – measures the communication latency between service and client
- **Iperf3** as microservice – measures the communication throughput from client to service

Experimental Results

Sockperf measurements
(Measured on client side)

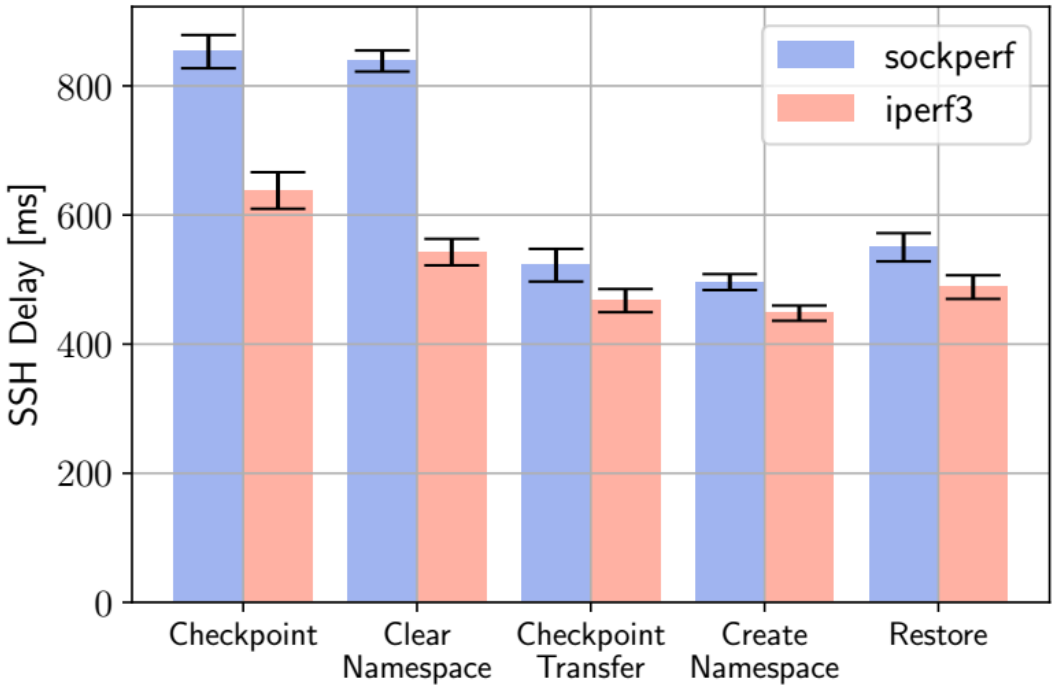


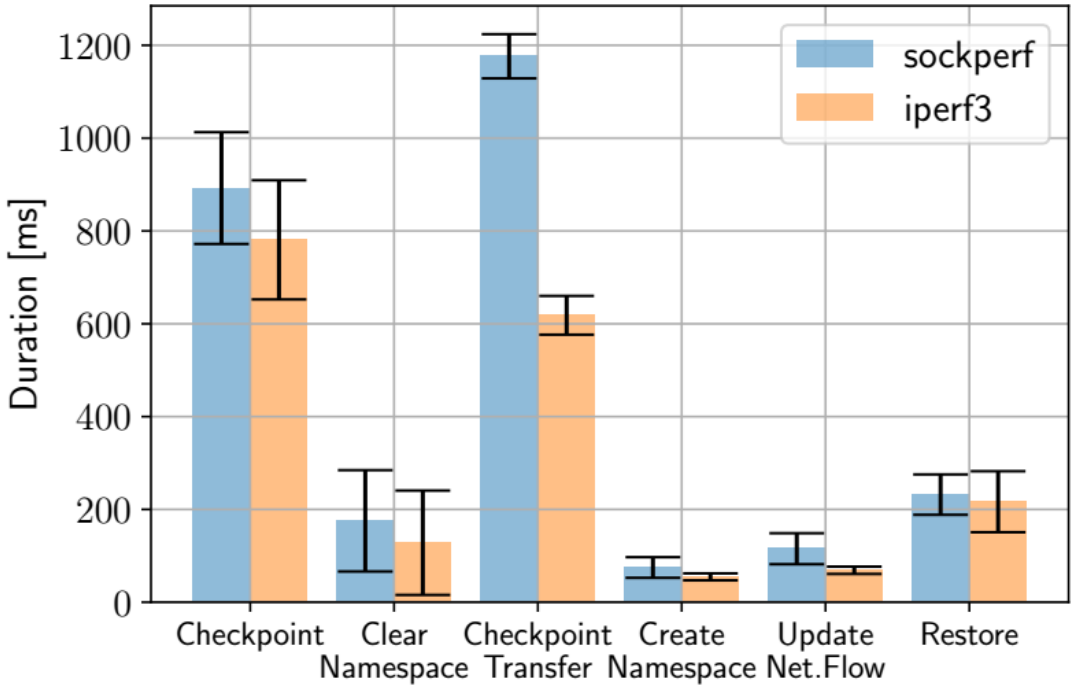
Iperf3 measurements
(Measured on client side)



1. TCP connection migration succeed.
2. No packet loss is experienced
3. Still service disruption is experienced

Migration duration breakdown





- SSH delay must be accounted for
- No additional overhead on checkpoint, transfer, and restore steps[1]

[1] A. Calagna, Y. Yu, P. Giaccone, and C. F. Chiasserini, “Processing-aware Migration Model for Stateful Edge Microservices,” IEEE ICC, 2023.

Conclusion

1. COAT allows TCP connection migration, thus **enabling service mobility** at the edge
2. COAT migration preserves the TCP socket state, hence **preventing packet loss** at the transport layer
3. COAT **does not require any modification** at either the server or the client application
4. COAT just introduces **a 14% maximum increase** on the migration duration

Future works:

1. Design a more efficient migration signaling system (alternative to SSH)
2. Integrate COAT with orchestration systems for complex scenarios

Thank you for your kind attention!

Politecnico
di Torino