L²5GC+: An Improved, 3GPP-compliant 5G Core for Low-latency Control Plane Operations

Yu-Sheng Liu† **Shixiong Qi*** Po-Yi Lin† Han-Sing Tsai†
K. K. Ramakrishnan* Jyh-Cheng Chen†

†*National Yang Ming Chiao Tung University* **University of California, Riverside*

Visit us at:
https://kknetsyslab.cs.ucr.edu/
https://cslab.cs.nycu.edu.tw/

Software-based 5GC Control Plane

5GC Control Plane NFs built as microservices

- 5GC Control Plane NFs coupled together
  - Account for task dependencies
  - NFs are ***networked*** together
    - Microservice "chain"
- Example
  - **UE authentication**
    - Part of UE registration procedure
  - Frequent interaction between **AMF**, **AUSF**, **UDM**

➢ Need tight coupling to provide low-latency to complete complex interactions

But:

- Control plane NFs communicate over 3GPP SBI
  - Use kernel-based HTTP/REST API
- ***Increased control plane latency!***

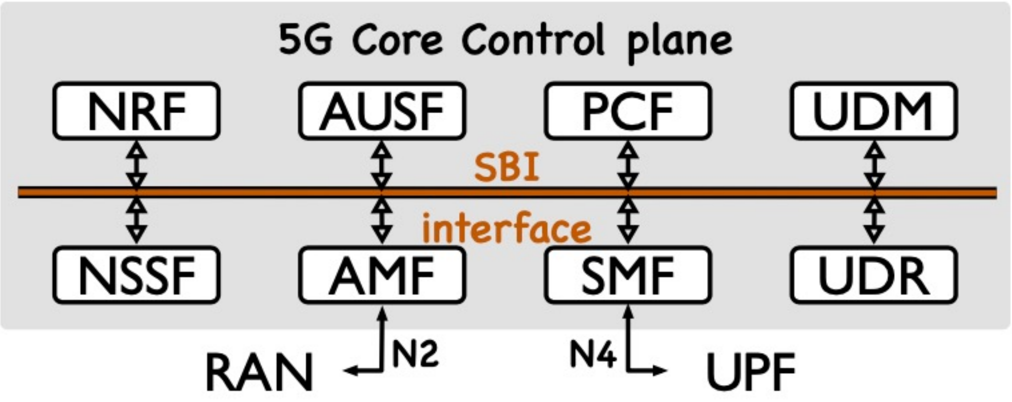
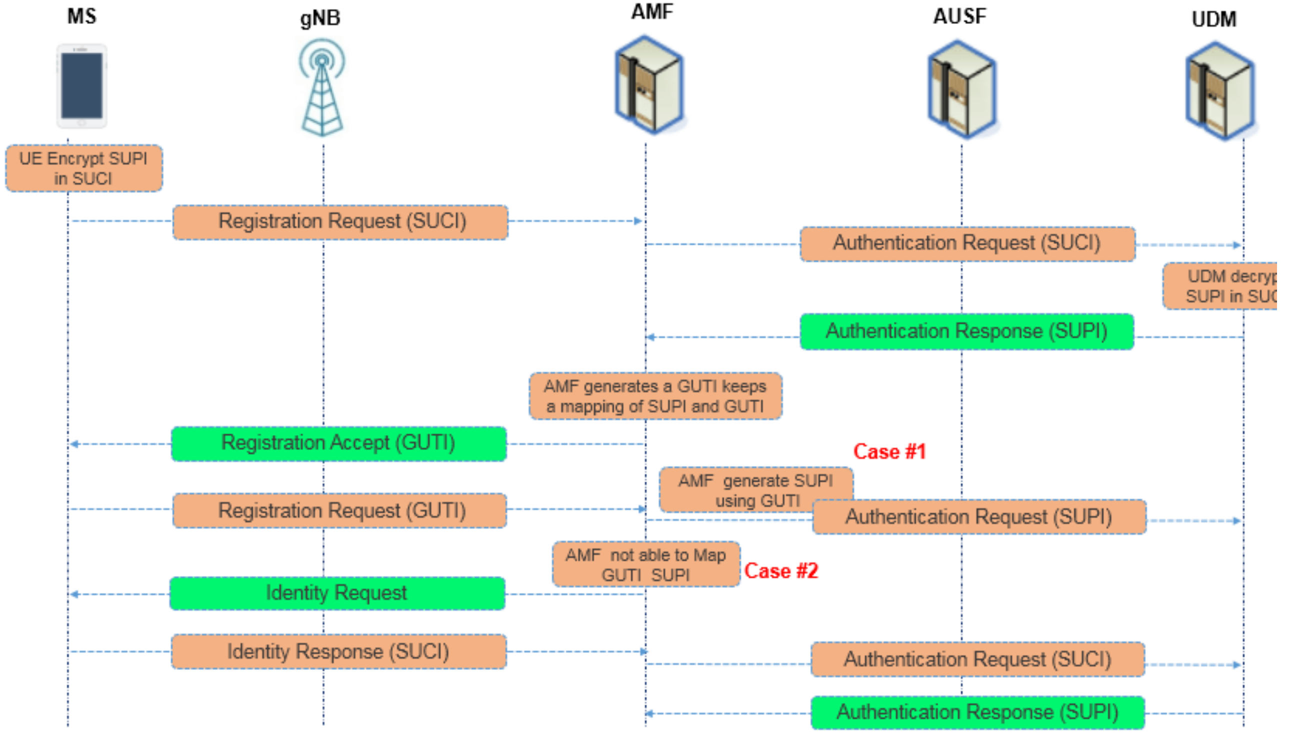


Fig. 1: The architecture of 5G core control plane



L²5GC: the state-of-the-art 5GC control plane

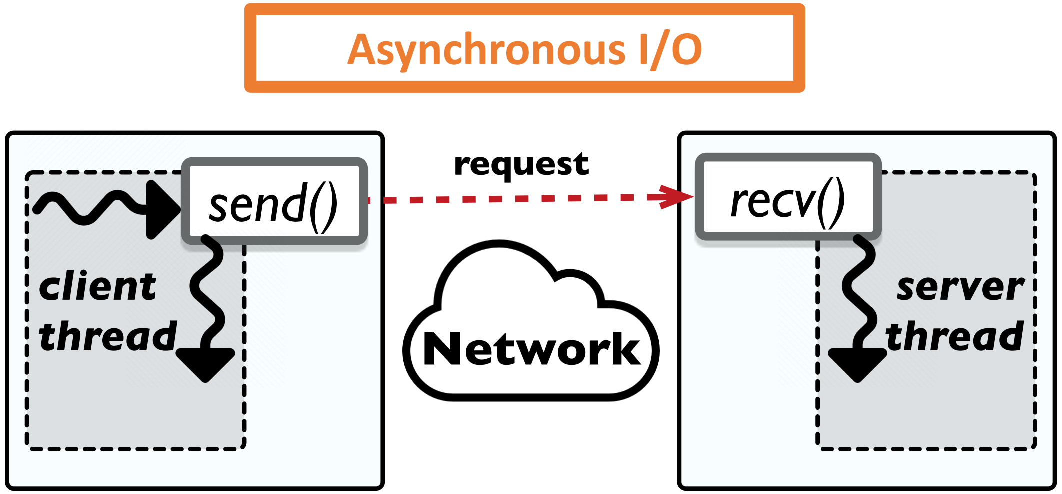
Using Shared Memory Processing to reduce control plane latency

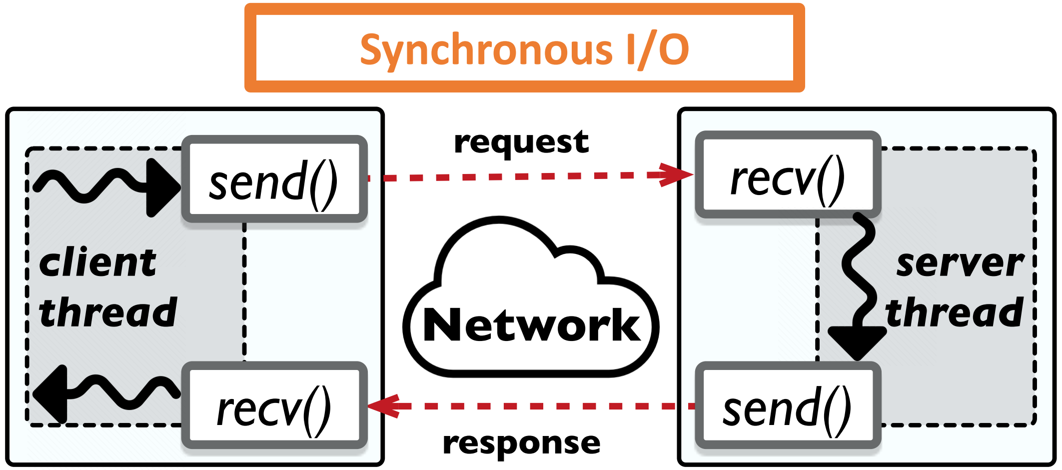
Limitations of existing **Shared Memory Processing:**
OpenNetVM[1]

- **Stateless, Asynchronous I/O** for shared memory communication
  - Caller **does not** wait for a response from callee
    - The caller is **not blocked** waiting for the request
  - ***Unidirectional*** data exchange: a send/recv pair
  - Was suitable for NFV platform with L2 NFs

- 3GPP SBI requires **Synchronous I/O**
  - Caller **expects** a response from callee
    - Caller **waits** (blocked) until response is returned
  - ***Bidirectional*** data exchange: two send/recv pairs

L²5GC needs enhancement!





[1] Zhang, Wei, et al. "OpenNetVM: A platform for high performance network service chains." Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization. 2016.

$L^2$5GC+

An Improved, 3GPP-compliant 5G Core for Low-latency Control Plane Operations

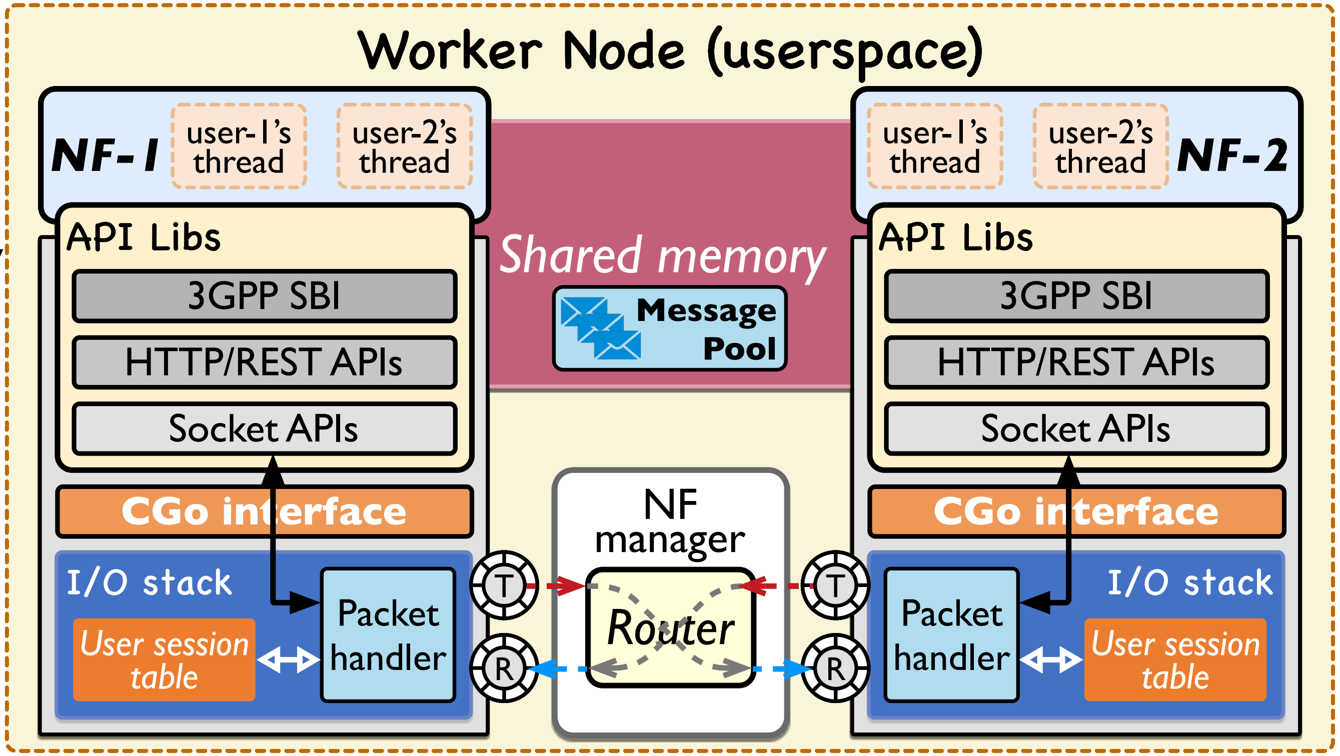
Networked Systems Group
UC RIVERSIDE

$L^2$5GC (directly based on OpenNetVM)		$L^2$5GC+
Asynchronous I/O over shared memory	→	**Synchronous** I/O over shared memory
Unable to scale up **concurrent** user sessions	→	**Concurrent connection** support
Programming language **incompatibility**	→	**Cross-language** support

$L^2$5GC+: An Improved, 3GPP-compliant 5G Core

Overview: $L^2$5GC+ Unified Sync/Async communication; multiple user sessions

- ***Shared memory I/O stack:***
  - ***Shared memory processing with lock-free rings***
    - ***Like $L^2$5GC, built with OpenNetVM***
  - ***Asynchronous I/O only***
  - ➢ Retaining a high-performance data plane
- ***API Libs:***
  - **Synchronous I/O support w/ shared memory**
    - ***Combining our efforts on X-IO[1]***
  - ➢ Being 3GPP-compliant
- ***Concurrent connection management:***
  - Using "User session table"
  - ➢ Scale up to a number of concurrent user sessions
- ***Cross-language support:***
  - **CGo interface** in Golang
  - ➢ Reducing porting effort (from free5GC)



[1] X-IO: A High-performance Unified I/O Interface using Lock-free Shared Memory Processing. 2023 IEEE 9th International Conference on Network Softwarization (NetSoft). IEEE, 2023.

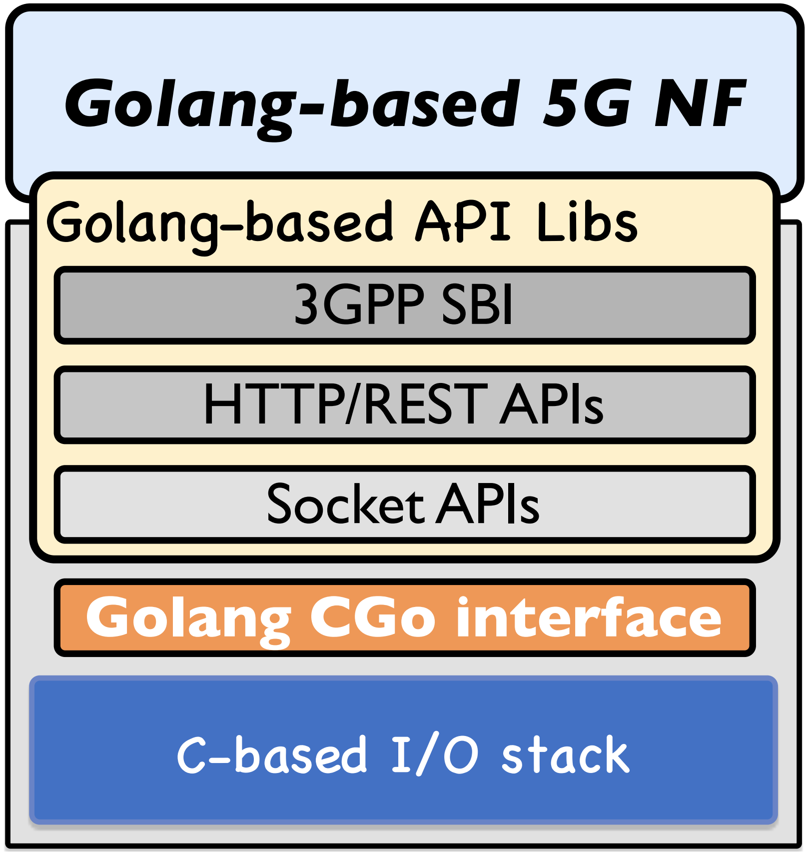
$L^2$5GC+: An Improved, 3GPP-compliant 5G Core

API libs

- *A layered design:*
  - *Bottom Layer: POSIX-like socket APIs*
    - *Equivalent Golang-style socket interfaces*
    - *Read(), Write(), Listen(), Accept(), Dial()*
  - *Middle Layer: HTTP/REST APIs*
  - *Top Layer: 3GPP SBI*

➢ **Facilitate ease of implementation and avoids re-implementing the entire stack**

- *Simply replacing the lower-layer socket APIs*



$L^2$5GC+: An Improved, 3GPP-compliant 5G Core

Networked Systems Group
UC RIVERSIDE

Supporting seamless porting of applications that depend on the POSIX socket API

- We keep the upper layer HTTP/REST APIs and 3GPP SBI **unchanged**

```
import "net"

/* Golang-style socket server */
listener, _ := net.Listen(server_address)
conn, _ := listener.Accept()

receive_buffer := make([]byte, RECV_MSG_SIZE)
n, err := conn.Read(receive_buffer)

conn.Close()

/* Golang-style socket client */
conn, err := net.Dial(server_address)

send_buffer := make([]byte, SEND_MSG_SIZE)
n, err := conn.Write(send_buffer)

conn.Close()
```

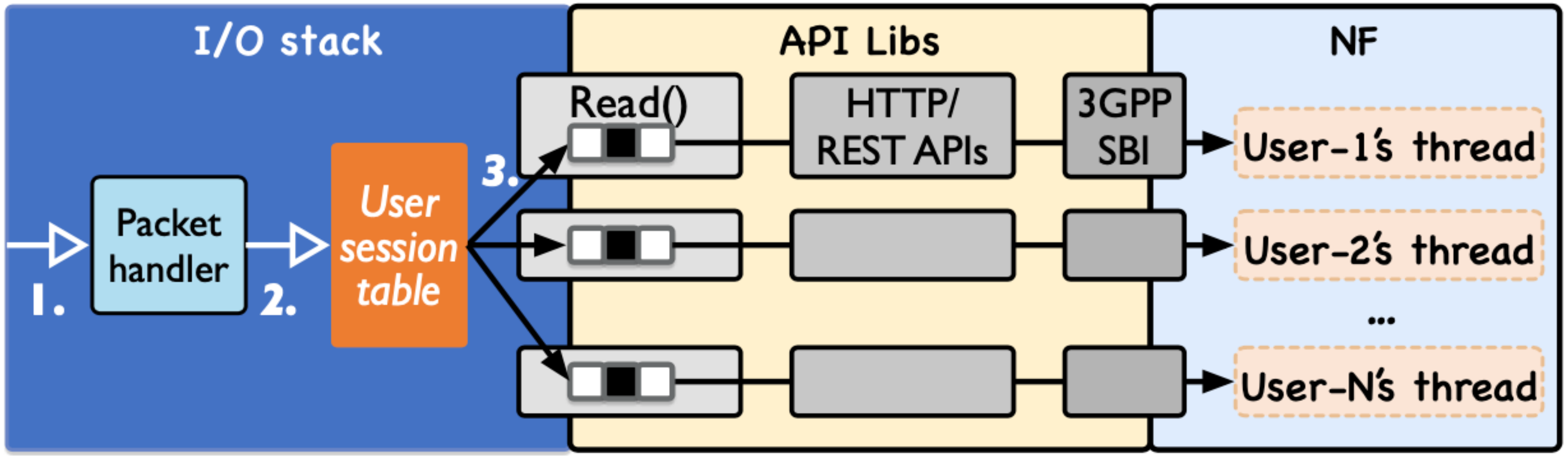
→

```
import "xio"

/* X-IO-based socket server */
listener, _ := xio.Listen(server_address)
conn, _ := listener.Accept()

receive_buffer := make([]byte, RECV_MSG_SIZE)
n, err := conn.Read(receive_buffer)

conn.Close()

/* X-IO-based socket client */
conn, err := xio.Dial(server_address)

send_buffer := make([]byte, SEND_MSG_SIZE)
n, err := conn.Write(send_buffer)

conn.Close()
```

L²5GC+: An Improved, 3GPP-compliant 5G Core

Concurrent user session support

- Turning "stateless" to "stateful"
  - **user session table** in I/O stack
  - Dispatch requests to different user sessions via ***IP 4-tuples lookup***



$L^2$5GC+: An Improved, 3GPP-compliant 5G Core

Networked Systems Group

UC RIVERSIDE

Additional Features in $L^2$5GC+: *Leveraging our earlier effort*

- ***HTTP/REST API optimization [2]***
- ***Connection management [2]***
  - Establishment; Teardown
- ***5GC data plane (UPF) optimization [1]***
- ***5GC deployment strategy [1]***
- ***5GC resiliency [1]***

Please refer to our previous work: $L^2$5GC [1] and X-IO [2]

[1] $L^2$5GC: A Low Latency 5G Core Network based on High-performance NFV Platforms. Proceedings of the ACM SIGCOMM 2022 Conference. 2022.
[2] X-IO: A High-performance Unified I/O Interface using Lock-free Shared Memory Processing. 2023 IEEE 9th International Conference on Network Softwarization (NetSoft). IEEE, 2023.

Evaluation

Experiment Setup

- ***3GPP-compliant commercial testbed***
  - UEs: laptops with 5G dongles (from Apal)
  - RU: from Alpha Networks Inc.
  - CU/DU: from AEWIN Technologies
  - UE Registration & PDU Session Establishment

- ***Simulated UE/RAN:***
  - UE & RAN simulator from $L^2$5GC
  - Scale up more UEs
  - Additionally look at Paging event

- We compare ***$L^2$5GC+*** with ***free5GC***

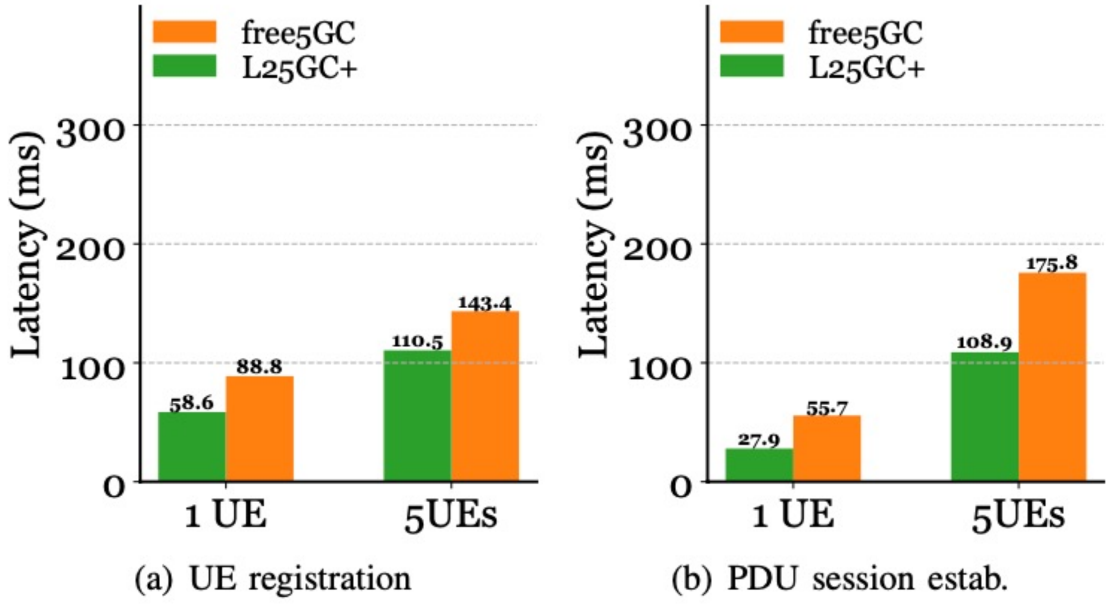
Evaluation

Commercial testbed results

- Verify the improvement of L²5GC+ with commercial testbed
  - "**CN**": The contributed latency by the 5GC

- ***UE registration***
  - L25GC+ has **1.5× lower** "**CN**" latency (Single UE) and **1.3× lower** "**CN**" latency (5 UEs)

- ***PDU session establishment***
  - L25GC+ has **2× lower** "**CN**" latency (Single UE) and **1.6× lower** "**CN**" latency (5 UEs)



Evaluation

Results with simulated UE/RAN

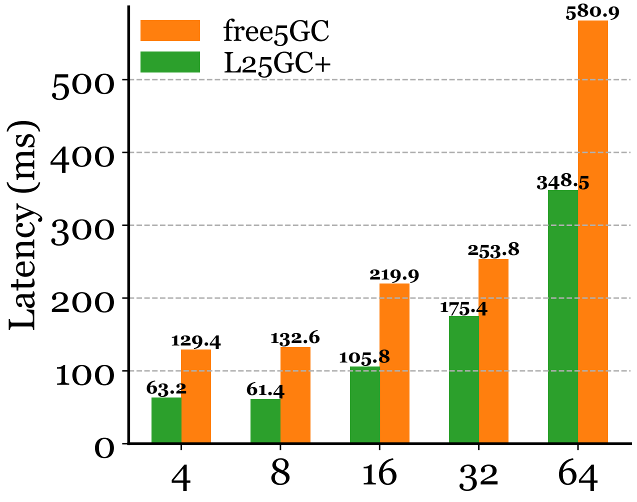
- Verify the improvement of L^{2}5GC+ with more UEs
  - We show the "**Total**" latency
  - X-axis is number of concurrent UEs

- ***UE registration***
  - L25GC+ has **1.9× latency reduction**

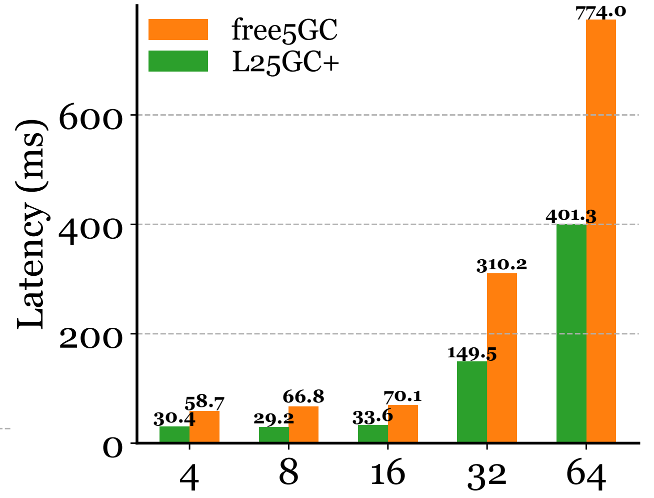
- ***PDU session establishment***
  - L25GC+ has **2× latency reduction**

- ***Paging event (idle-to-active transition)***
  - L25GC+ has **1.6× latency reduction**

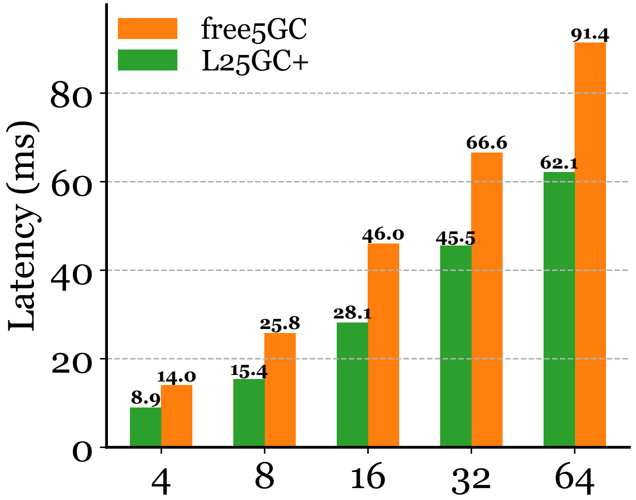
UE Registration



PDU session estab.



Paging



Conclusion

L25GC+ is an improved, 3GPP-compliant 5GC designed for low-latency control plane operations

- **Synchronous I/O interface**
  - **3GPP-compliant**
- **Concurrent user session support**
  - **More scalable compared to $L^2$5GC**
- **=> A shared memory based network stack**
- **=> Up to 2X control plane latency reduction compared to free5GC**

- **Inter-node communication**
  - **Current using kernel protocol stack; RDMA in the future**
- ***$L^2$5GC+ is Available***
  - **☞ Find $L^2$5GC+ at: https://github.com/nycu-ucr/L25GC-plus.git**
  - **If you have any questions or comments, please feel free to email us (l25gc@googlegroups.com)**

Backup Slides

Networked
Systems Group
UC RIVERSIDE

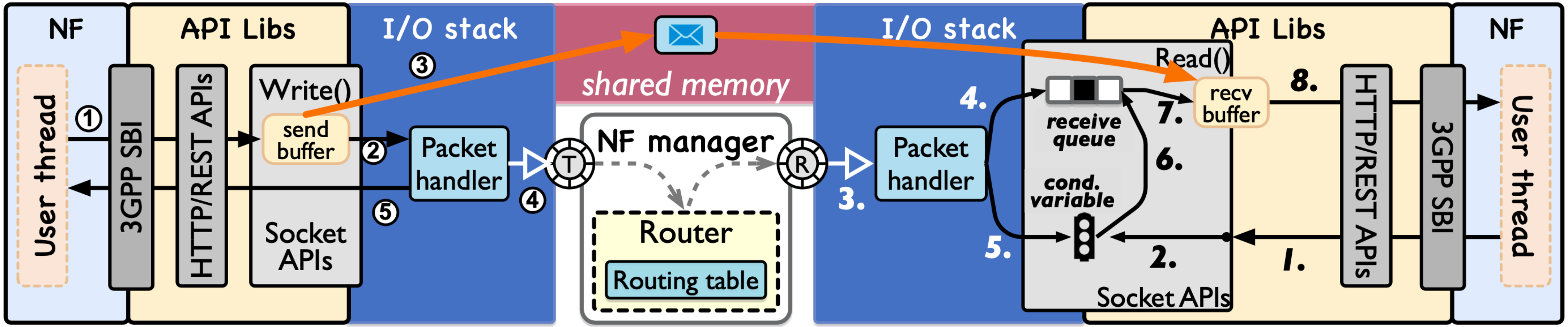
L^25GC+: An Improved, 3GPP-compliant 5G Core

Adding Synchronous I/O

- **Adding *blocking primitives* to the asynchronous shared memory network stack**
  - The caller of Read() is **blocked** until it **receives** the request from the I/O stack
  - The caller of Write() is **blocked** until the data in **send buffer** is moved to the **shm buffer**
  - ➢ **Conditional Variable** in OS
    - **Caller waits** until it is **signaled to wake up**
  - ➢ **Batch wake-up mechanism**
    - A **receive queue** to buffer the requests (*descriptors*)
    - ***Reduce wake-up overhead***

Distinguish blocking call with synchronous I/O

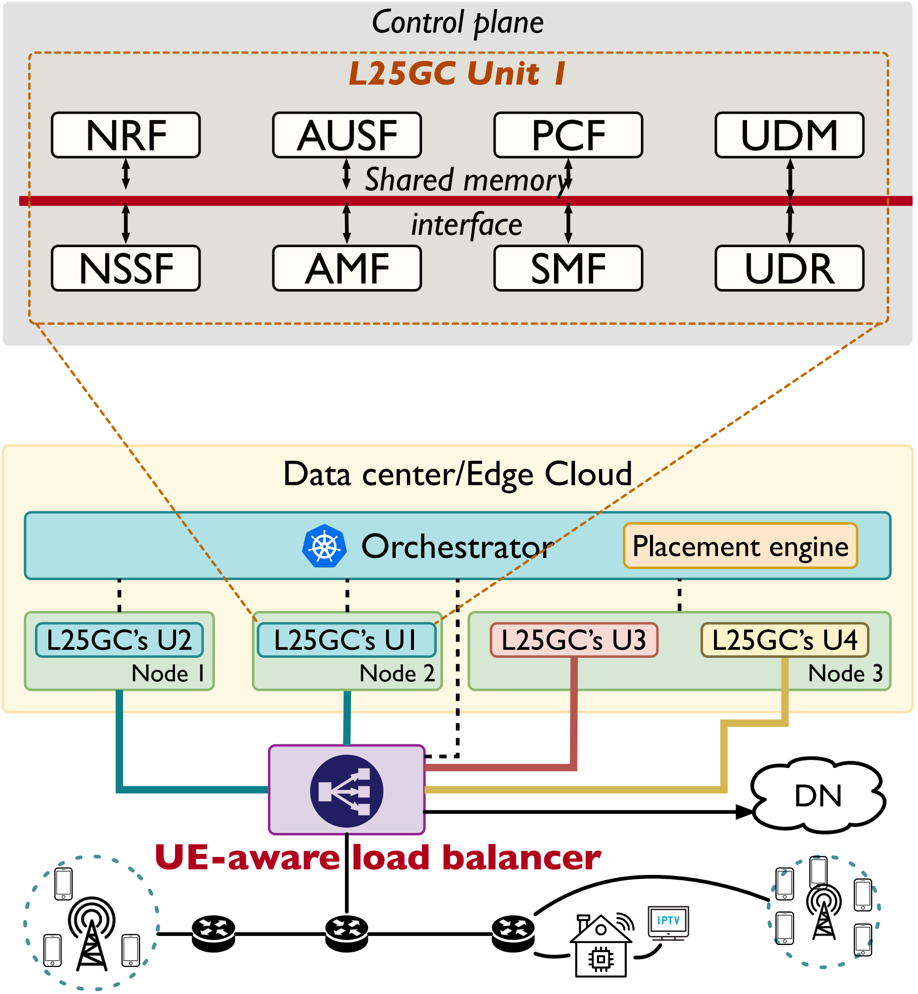
- **Example: How does our Read() & Write() synchronize?**



L25GC: the state-of-the-art 5GC control plane

Optimizing the Service Based Interface

- ***Shared memory for communication between NFs in same node***
  - Built on OpenNetVM[1]
    - A high performance NFV platform based on DPDK
  - Information exchanged directly in userspace: no kernel overheads or protocol processing
  - Zero-copy packet delivery between NFs: no data movement, serialization/de-serialization cost



[1] Wei Zhang et al. 2016. OpenNetVM: A Platform for High Performance Network Service Chains. HotMIddlebox '16

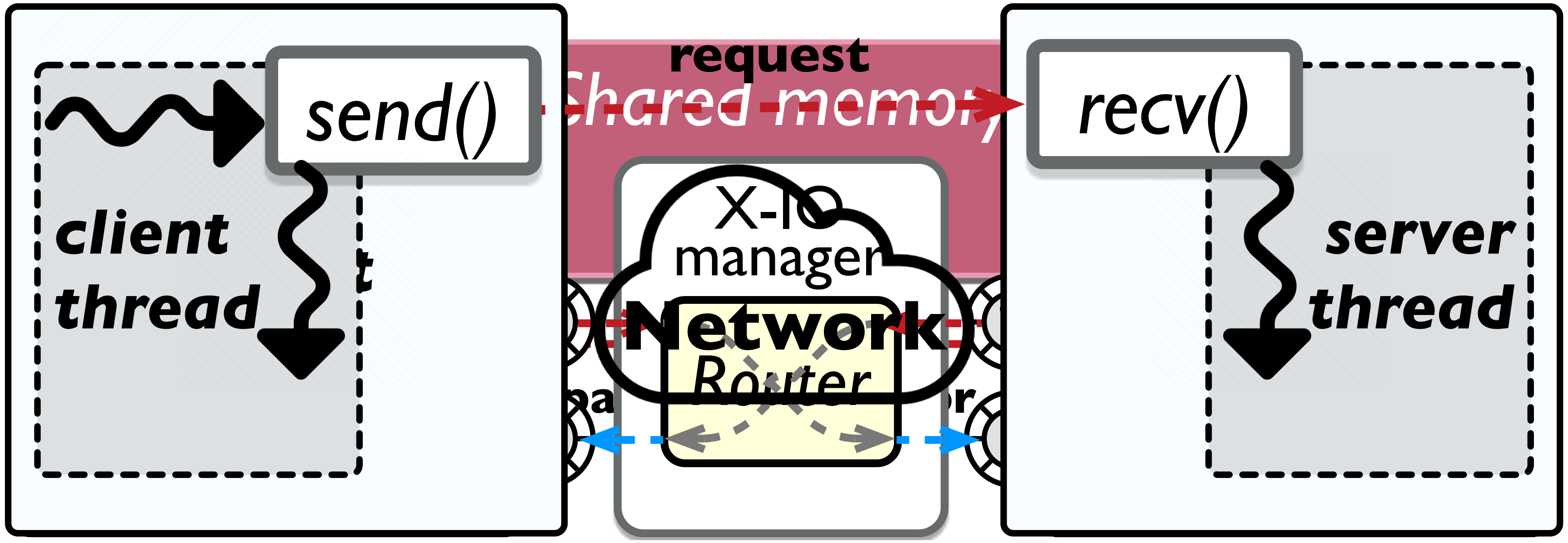
L²5GC: the state-of-the-art 5GC control plane

Networked Systems Group
UC RIVERSIDE

A Quick Primer on *ZERO-COPY* Shared Memory Processing from L²5GC

- ***The shared memory network stack in L²5GC***
  - Shared Memory Pool
  - Lock-free Producer/Consumer Rings with Busy-polling

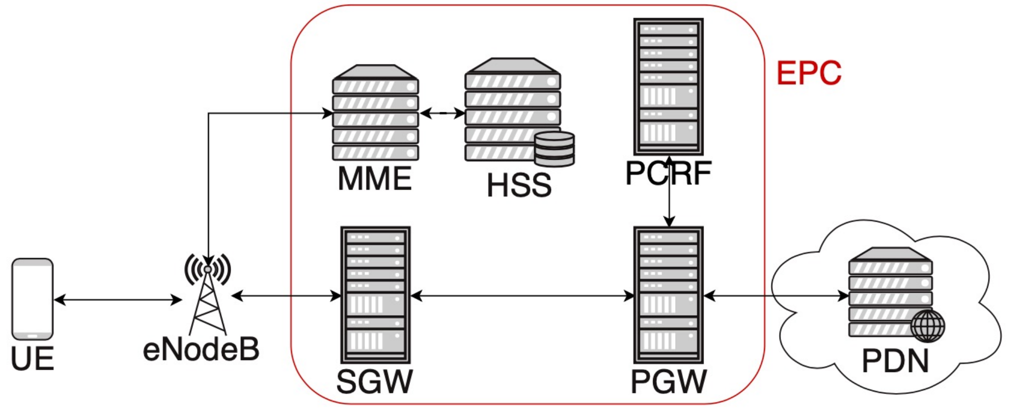
- Zero-copy I/O primitives from the I/O stack
  - ***io_malloc***(), ***io_tx***(), and ***io_rx***()
  - ***Non-blocking, Asynchronous*** operations only
    - Not compatible with 3GPP SBI
  - ***No state keeping***
    - Unable to track connection state
  - ***Developed in C***
    - Massive refactoring of free5GC (in Golang)



The evolution of *softwarized* cellular core

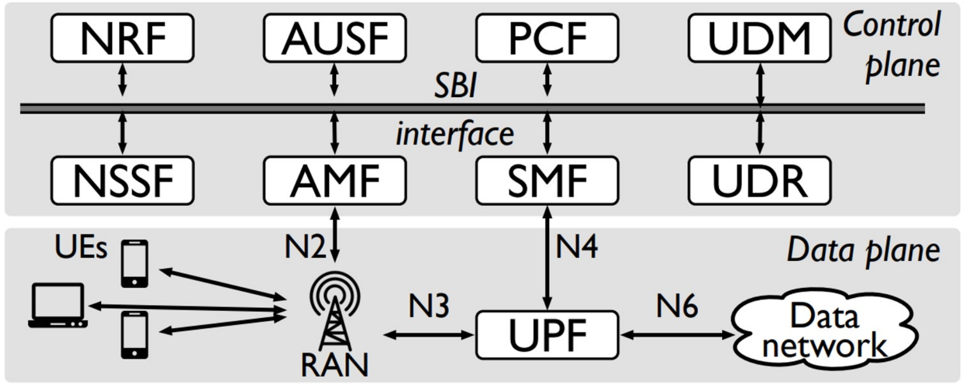
Moving from monolithic services to microservices

Monolith LTE EPC



- All-in-one
- ***Hard*** to scale out
- ***Poor*** modularity

Microservices 5GC



- ***Independently*** deployable
- ***Loosely*** coupled
- ***Easy*** to scale out
- ***Good*** modularity

Challenges for 5G Cellular Core control plane

3GPP-recommended Service Based Interface (SBI)

- Control plane NFs communicate over 3GPP SBI
  - Kernel-based HTTP/REST API
- Penalties:
  - copies, serialization/deserialization, protocol processing,
- ***Increased control plane latency!***

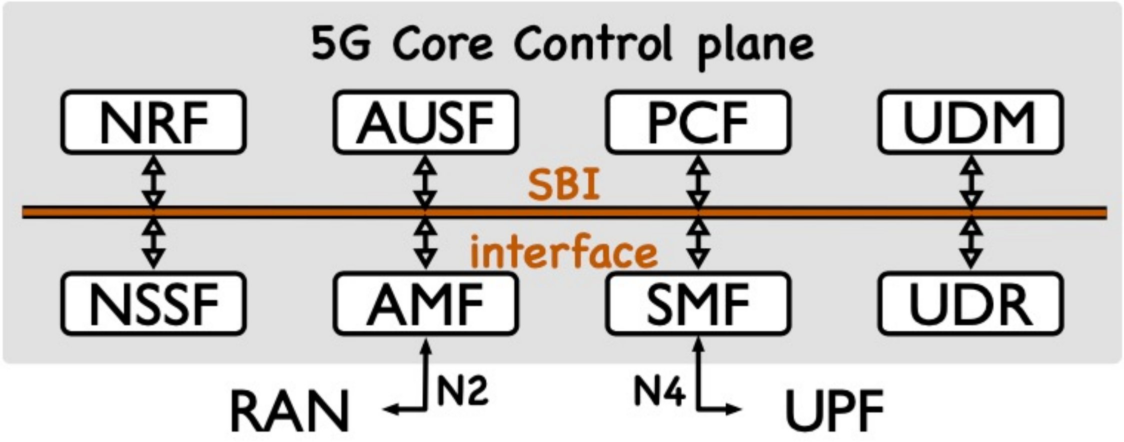


Fig. 1: The architecture of 5G core control plane

Control Plane needs speed and efficiency improvements too, not just speed up of the Data Plane