Prototyping and Load Balancing the Service Based Architecture of 5G Core using NFV

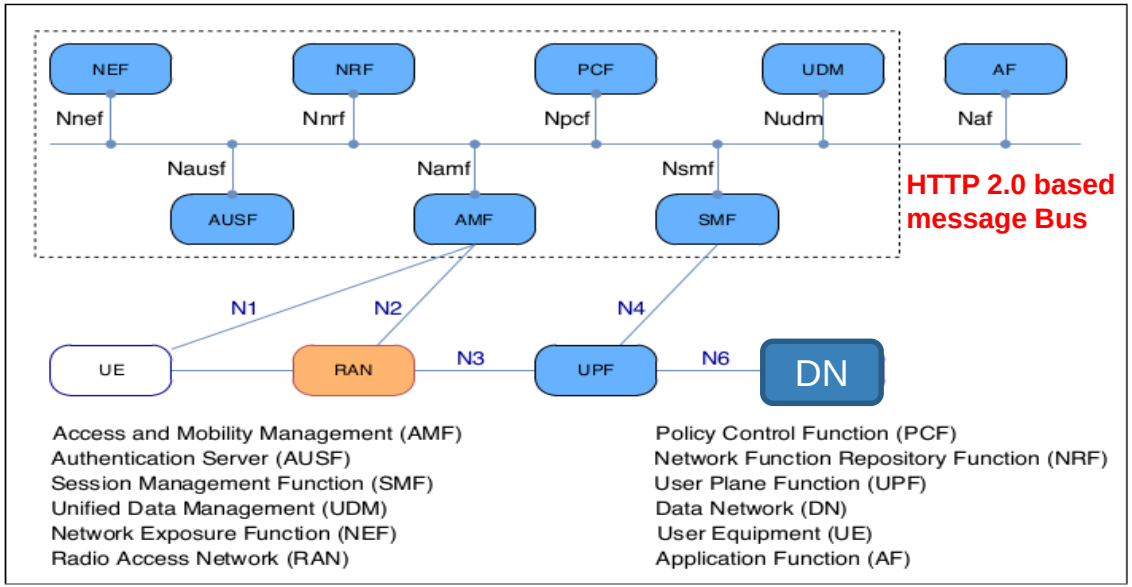
Vamshi Kiran Buyakar, Harsh Agarwal,Bheemarjuna Reddy Tamma, and Antony Franklin A

Indian Institute of Technology Hyderabad, India

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

IEEE NETSOFT 2019

5G: Service Based Architecture



Service-based representation using Service based interfaces (SBIs) for interaction in CP of 5G Core

- Client-Server Based Architecture
- Each NF is registered to a Central Repository Function (NRF)
- Stateless, Cacheable, Layered system Communication

Motivation

- The heavy bursts of signaling traffic in the 5G Core require it to be robust and scalab
  - Explosive traffic demand from diverse verticals & massive # of IoT devices
  - Heterogenous & dense deployment of cells
- It is necessary to implement a highly scalable and resilient architecture of the contro that can dynamically respond to any kind of network situation
- Cloud computing, SDN & NFV could offer cost-effective and competitive architectural solutions for mobile operators as well

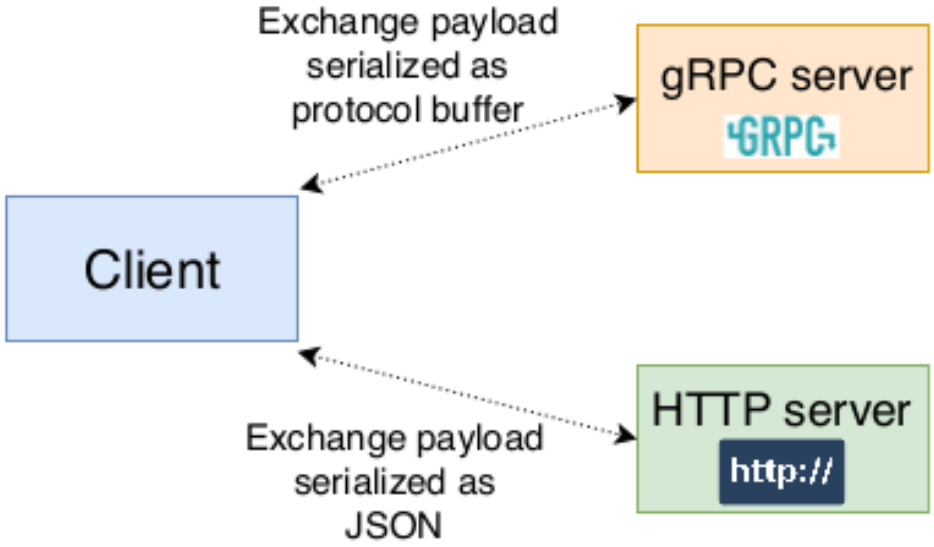
Main Contributions

- In this work, we implement the **SBA of 5GC** and deploy it in an **NFV environment**
- To reduce the communication latency and the load on the NFs, we use **Google Remote Procedure Call (gRPC)**, a modern open-source RPC framework, instead of **HTTP REST API as SBI**
- We implement a distributed setup of the **NRF for service registration and discovery, using Consul**, an open-source distributed &highly available service discovery/configuration system
- We propose using a **look-aside load balancer** instead of a proxy based load balancer to meet the high scalability and low latency requirements of the 5G control plane

Motivation for choosing gRPC over REST for SBI

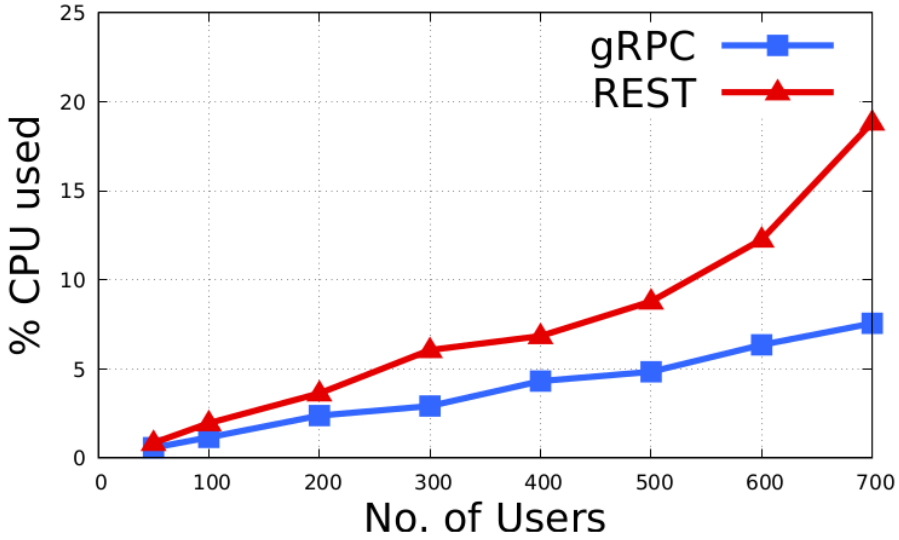
- Comparison of gRPC and REST
  - Protobuf vs. JSON
    - REST messages typically contain JSON objects
    - gRPC accepts and returns Protobuf messages
      - Protobuf is very efficient in terms of performance
  - HTTP/2 vs. HTTP 1.1
    - REST depends heavily on HTTP (usually HTTP 1.1) while the gRPC uses the newer HTTP/2 protocol
      - HTTP/2 reduces RTT by multiplexing REQ/RES and minimizes overhead by compression of Headers

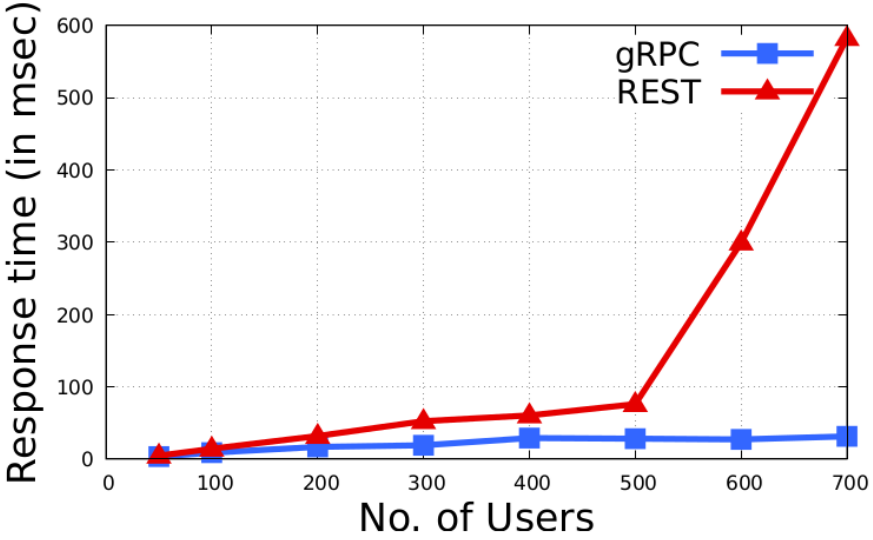
Benchmarking setup of gRPC and HTTP REST



- Client threads are set up which periodically query the two endpoints
- The average response time taken to query a request and CPU utilization of the server to serve the requests are measured by varying the number of clients.

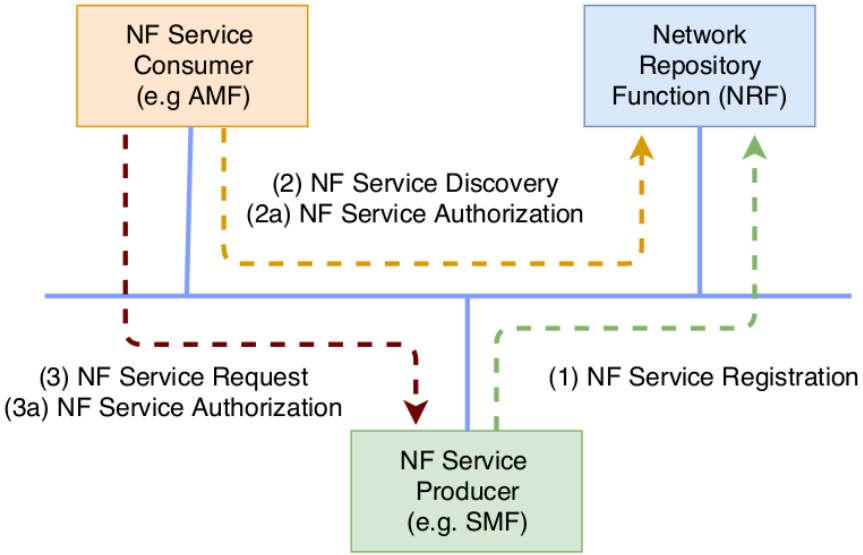
Comparison of gRPC and REST





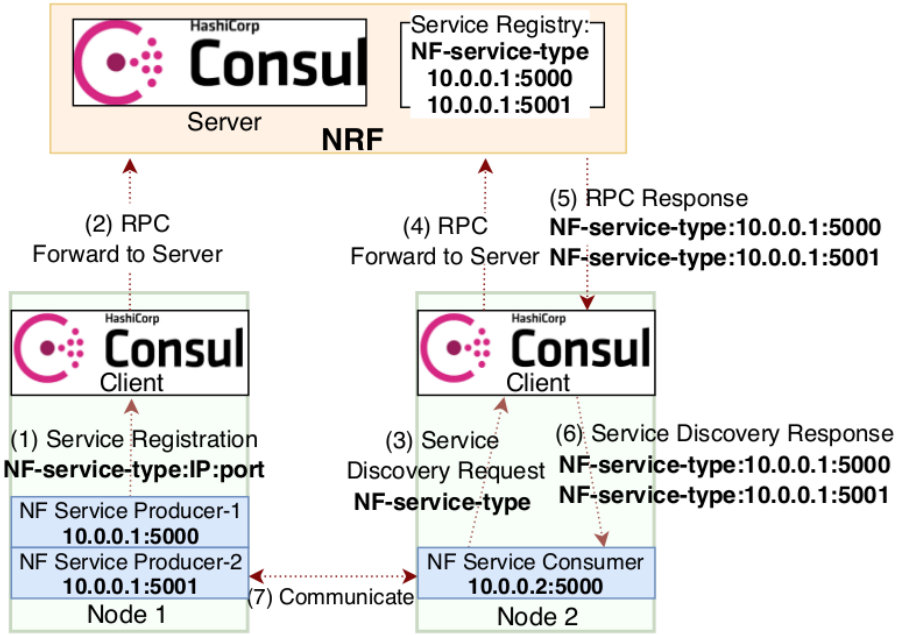
⟶ Unmarshalling JSON is a computationally expensive task & HTTP 1.1 is less efficient, hence REST is performing poorly

Network Function Repository Function (NRF)



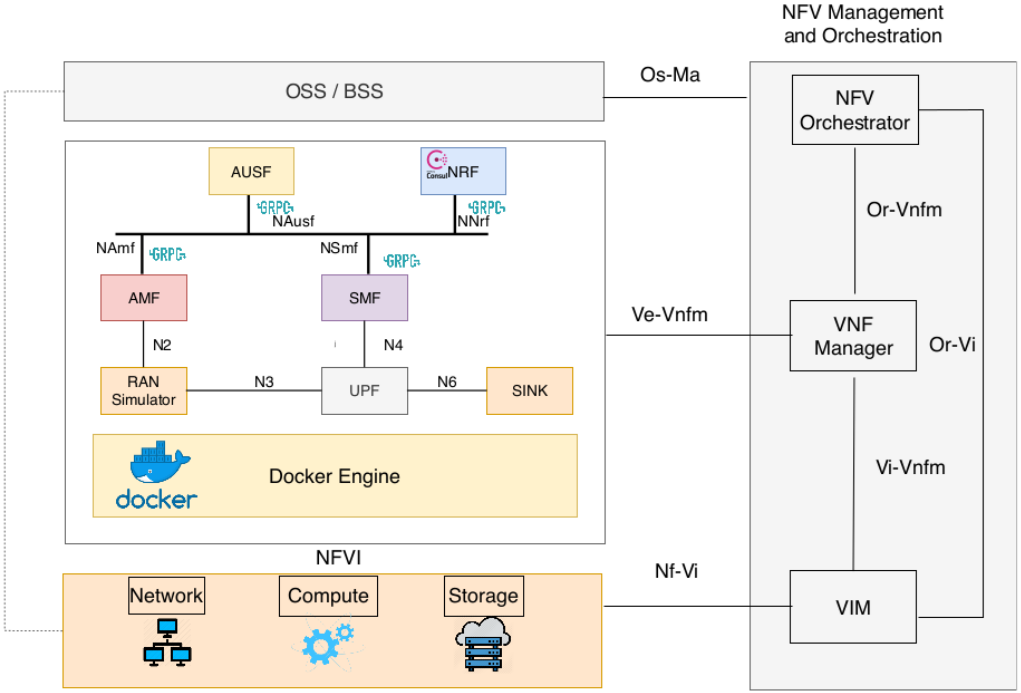
- NRF provides registration and discovery functionality so that the instances of network functions (NFs) can discover each other and communicate via APIs.
- The service registration and discovery procedures are followed as depicted in figure.

NRF implementation using Consul



- Consul server on a dedicated server node
- NF service producers and consumers are on separate server nodes with every node running a Consul client
- New NF Service Producer spawned, registers itself with Consul
- NF Service Consumer sends a service discovery request containing the type-of-service to the Consul
- Consul server forwards apt instance of NF Service Producer to the NF Service Consumer

Proposed gRPC based 5G Core



gRPC based 5G architecture aligned with ETSI NFV [6] reference architecture

1. Evaluation of gRPC based 5G Core

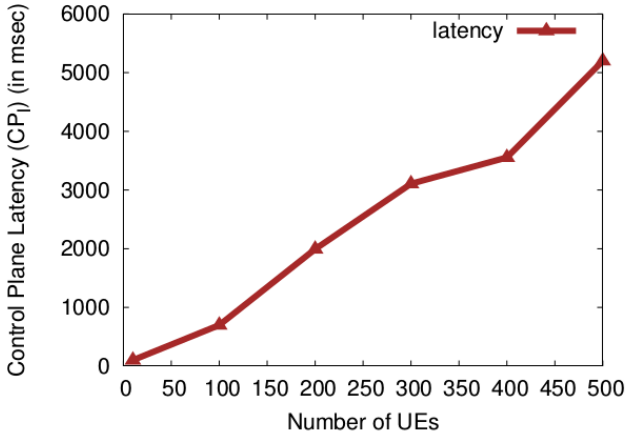
Experimental Setup

Entity	Cores	RAM	OS
Server Node	56	64GB	Ubuntu 16.10, 64 bit

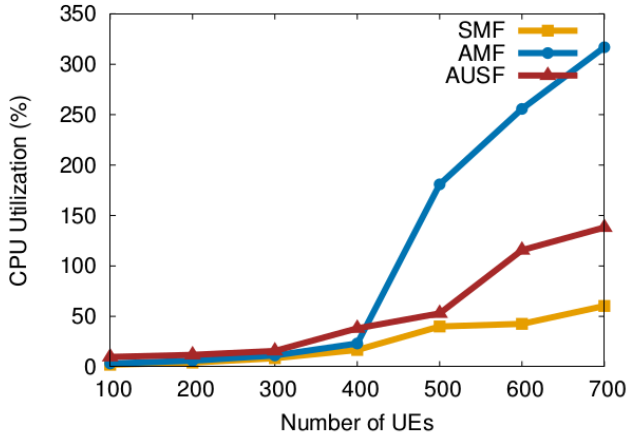
Parameter	Value
Number of UEs	10 to 700
Simulation time	120 Minutes
UE Data Transfer	Iperf3 - TCP Traffic
Virtualization platform	Docker
RAN &Core Simulator	gRPC-5G [12]
Live status monitor	Prometheus 1.6.2

In this experiment, only a single instance of each NF of 5G System is considered for processing UE traffic.

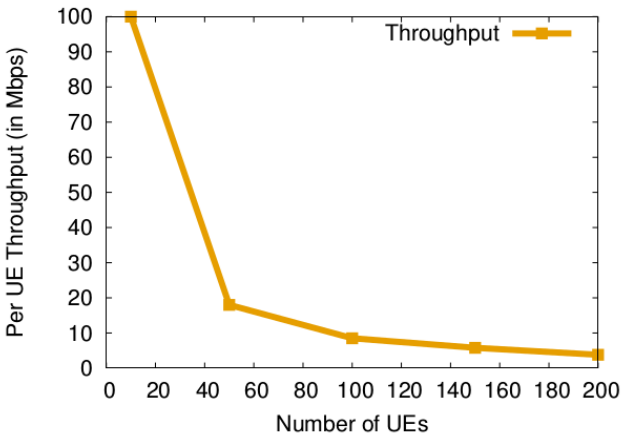
Evaluation of gRPC based 5G Core



Control Plane Latency



CPU Utilization of the Host Machine



Per UE Throughput

⟶ Need for multiple instances of AMF/SMF/etc to distribute (balance) the load and thereby keep a check on latency and improve UE throughput

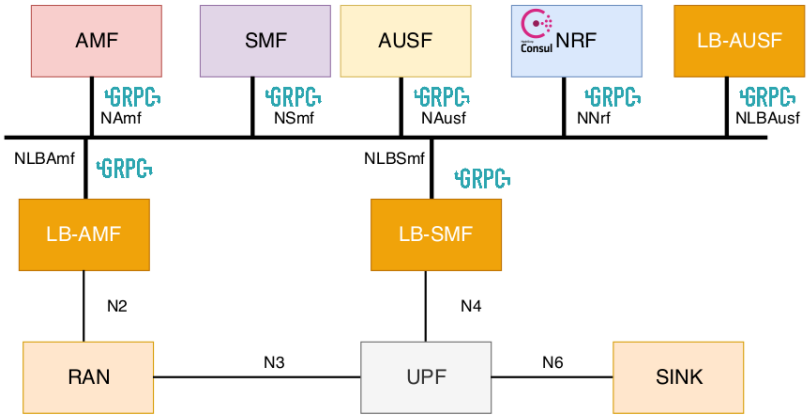
Load Balancing among multiple NF instances

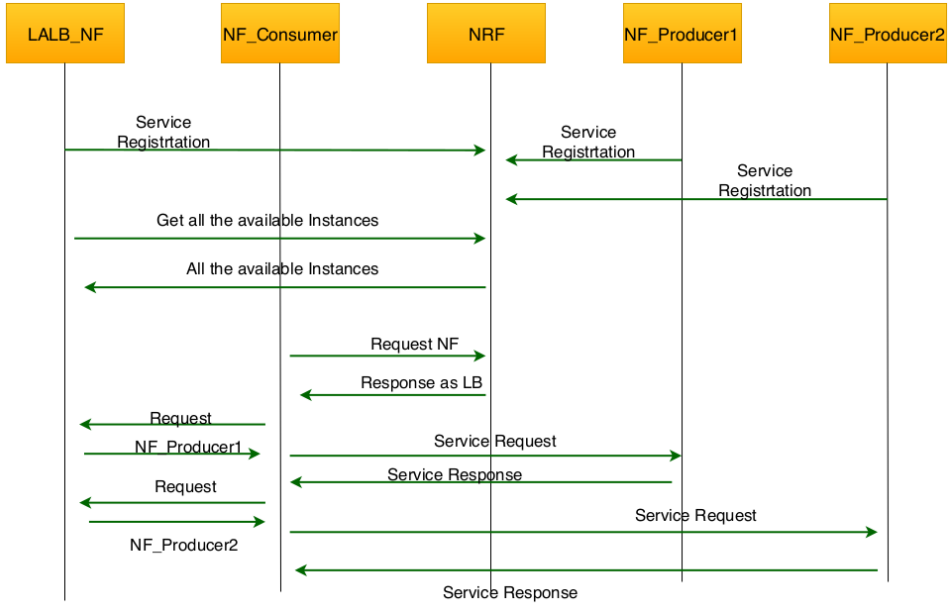
- Load balancing architectures
  - **Proxy load balancing**
    - simple to implement
    - works with untrusted clients
    - higher latency (since the LB is in the data path)
  - **Client side load balancing**
    - high performance (because of the elimination of an extra hop)
    - adds to the complexity of the client and adds a maintenance burden
    - clients must be trusted

Look Aside Load Balancing

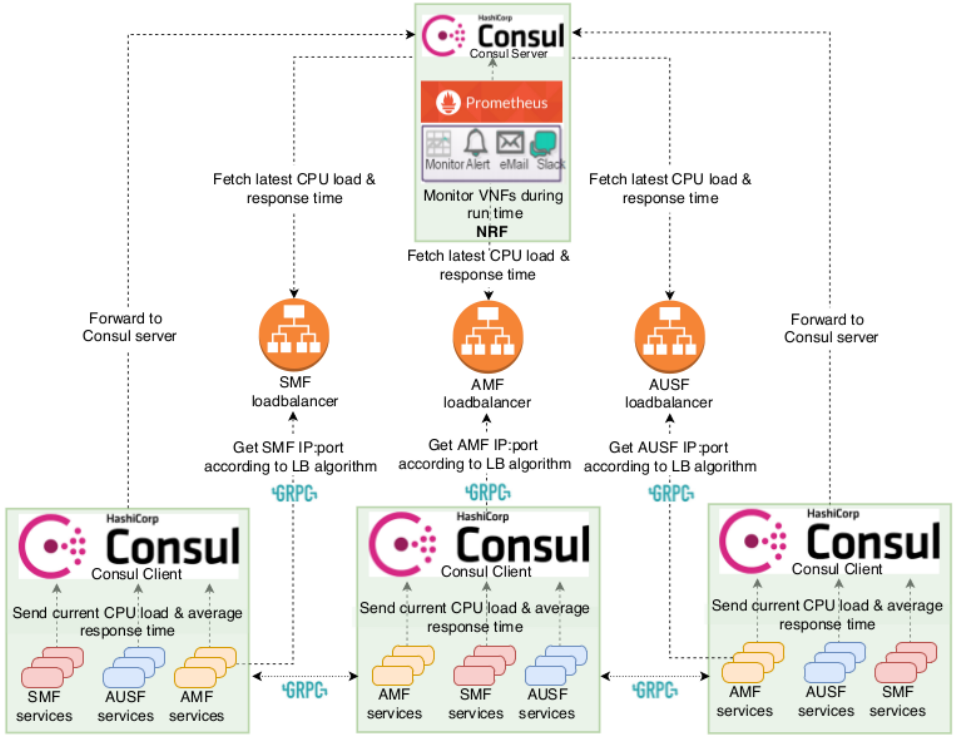
- Hybrid of client-side and proxy based load balancing
- There is a special LB server called the Look Aside Load Balancer (LALB)
  - The clients query the LALB, and the LALB responds with the best server to use
  - The client then directly interacts with that backend server. The servers share th reports with the LALBs regularly.
- Benefits
  - Clients can be untrusted
  - Low latency
  - Scalable

LALB Architecture for gRPC based 5G Core





Load Balancer Implementation Framework



2. Evaluation of Load Balancer

Experimental Setup

Entity	Core	RAM	OS
Server Node	56	64GB	Ubuntu 16.10, 64-bit

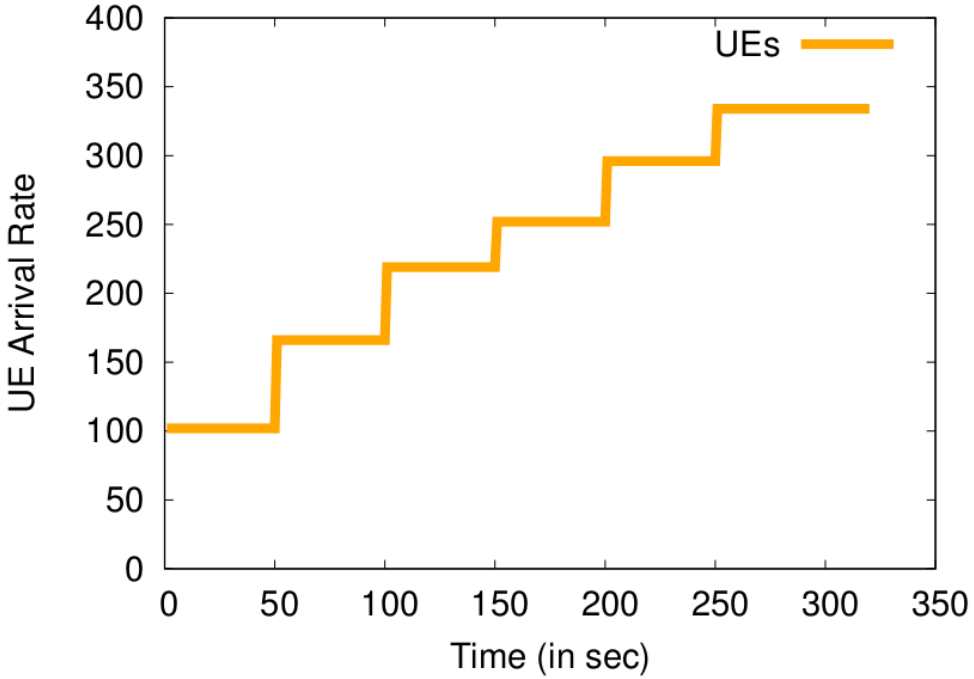
Parameter	Value
Number of UEs	0 to 600
Simulation time	350 Seconds
Virtualization platform	Docker
RAN & EPC Simulator	gRPC-5G [10]
Live status monitor	Prometheus 1.6.2

Evaluation of LALB

1. Measuring the reduction of CP latency by increasing the number of AMF instances

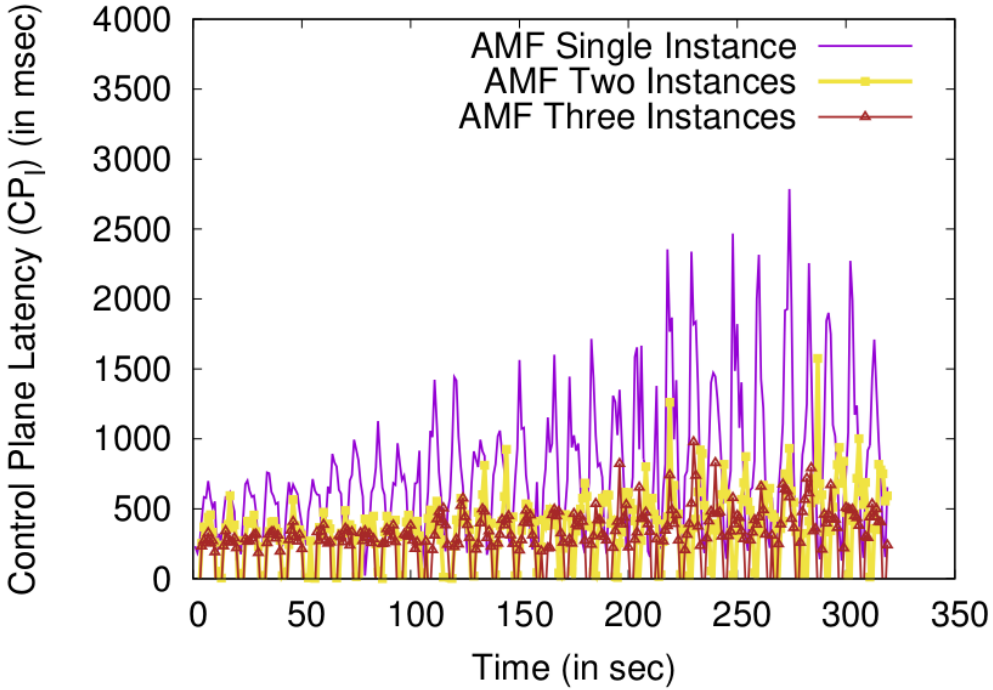
2. Observingthe variation of CP latency with various load balancing schemes

Load variation over simulation time



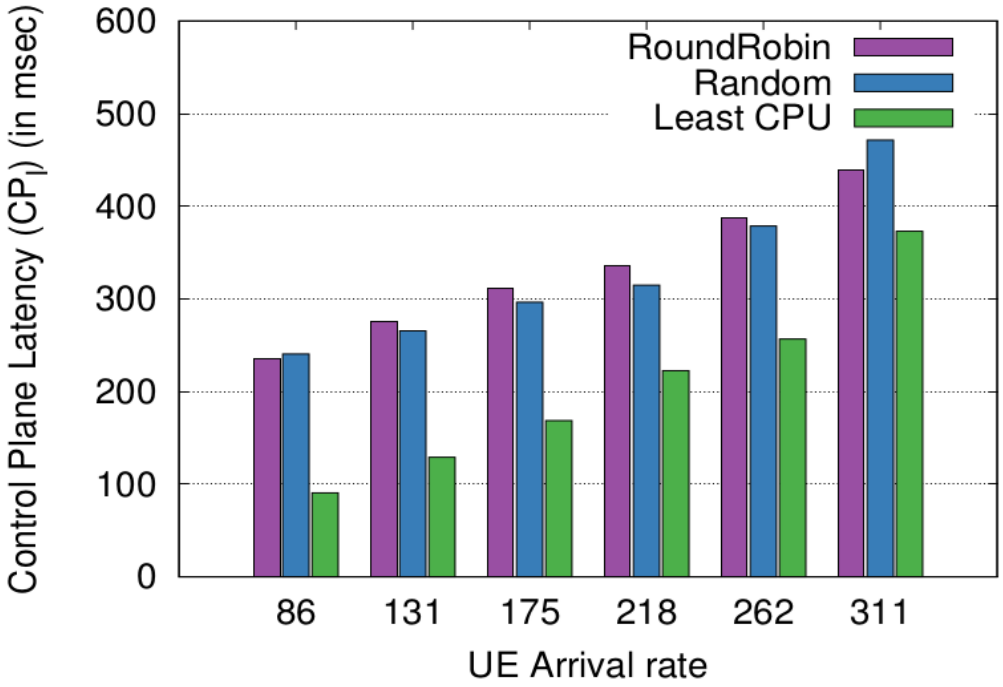
- RAN simulator [10] generates continuous control signaling traffic to EPC.
- UE Arrival Rate is increased at every 50 sec
- UEs continuously perform attach, data transfer, and detach activities

Control Plane Latency



- With multiple instances of AMF, it is observed there is much reduction in CP latency when compared to a single instance.
- This difference is mainly due to high concurrency rate provided by the multiple instances of AM F.

Control Plane Latency with various LB Schemes



- The CP latency for LCU is lesser than both RR and RD because in LCU the consumer accesses the currently least loaded AMF
- Hence the consumer's request faces very less contention in the AMF and is processed at a much faster rate.
- Therefore picking an appropriate load-balancing policy plays a vital role in building a scalable SBA for 5GC.

Conclusions

- We designed and implemented a gRPC based 5G Core architecture to handle huge signaling overhead in mobile networks. We used Consul for realization of NRF.
- We proposed a Look Aside Load Balancer (LALB) which suits the Service Based Architecture of 5G
- We evaluated our LALB with various load balancing algorithms
  - Experimental results suggest that carefully chosen load balancing algorithms ca significantly lessen the control plane latency when compared to simple random round-robin schemes

References

[1] 5G System Architecture 3GPP TS 23.501 and TS 23.502

[2] gRPC (https://grpc.io/)

[3] HTTP REST (https://restfulapi.net/)

[4] Consul (https://www.consul.io/)

[5] Look aside load balancing

[6] "ETSI NFV" http://www.etsi.org/technologies-clusters/technologies/nfv.

[7] REST versus gRPC Comparison

[8] Proxy Load Balancing

[9] Client side Load Balancing

[10] gRPC based Service Based Architecture for 5G Core (https://github.com/sipian/5G-gRPC-SBA)

Acknowledgements

This work is partially supported by the projects “Visvesvaraya PhD Scheme” and “Converged Cloud Communication Technologies”, MeitY, Govt. of India.

सत्यमेव जयते

Ministry of Electronics
& Information Technology
Government of India

THANK YOU!

Queries ?

Email: tbr@iith.ac.in
Homepage: http://www.iith.ac.in/~tbr
Google Scholar Profile: http://goo.gl/JdgRB
NeWS Lab: https://newslab.iith.ac.in/

Motivation for choosing gRPC over REST

- Comparison of gRPC and REST
  - Messages vs. Resources and Verbs
    - REST does n't just use HTTP as a transport, but embraces all its features a builds a consistent conceptual framework on top of it.
      - It is actually quite challenging to map business logic and operations int strict REST world.
    - The conceptual model used by gRPC is to have services with clear interfac structured messages for requests and responses.
      - It allows gRPC to automatically generate client libraries.