Side Meeting
Compute First Networking (CFN)
Use cases for edge, and an example

- Edge-Cloud Based Recognition in Augmented Reality (AR)

- What kinds of services are suitable for edge?
  Generally: low latency + high bandwidth

- Rely on center cloud to help edge (failover, load balancing)?
  Probably not!
  Edge helps edge
Requirements & Challenges

**Requirement 1- Service equivalency:** 100s of edge sites may provide equivalent services for clients

**Requirement 2- Service dynamics:** the optimal service instance for a specific request from a client may vary: subject to proximity, load, network conditions, service failure etc. One Edge has limited resources, e.g. up to 10 servers, and edge is also less reliable than cloud.

- Use Anycast for Service equivalency
- But make it dynamic: be adaptive to conditions to get to optimal service instances

**Challenge 1- Flow Affinity:** it should be avoided to route different packets in the same flow to different service instances

**Challenge 2- Incremental change:** minimize the change to current routers, especially the data plane, e.g avoiding huge flow tables in routers: e.g. 100K users * 100 services = 10M flows
• Clients use Anycast IP address to access an MEC service
  • More than one edge are reachable with it
  • Choose the binding edge to serve the request upon the first packet
  • Keep binding edge same for subsequent requests of the flow
• CFN nodes exchange info
  • Computing load for MEC
  • Network cost
• CFN ingress & egress can be the same node
• CFN nodes exchange computing load info
• Metrics to be defined
  • capacity, number of connections being served...
  • quantized value, boolean...
• CFN ingress select the egress based on computing load info + network info
• CFN ingress & egress can be the same node

SID – IP anycast address for an MEC service
BIP – binding IP, i.e. real IP, for an MEC service instance
Example: Data Plane

- CFN ingress selects the egress based on upon receiving the 1st packet
- Save binding table about (anycast IP, CFN egress) for active flows
- Binding table can be saved closer to clients, e.g. at UPF
- Flow affinity: subsequent packets from the flow always sent to the same egress
- Overlay or SR based encap

SID – IP anycast address for an MEC service
BIP – binding IP, i.e. real IP for an MEC service instance
Proof-of-Concept

Setup
- Service randomly accessed from 3 Edges
- 10~30 request per client
- 2~5 ms processing time for each service

Objective
- Use JCT (Job Completion Time) as key KPI for comparison
Some preliminary tests

- Case 1: Scheduling Multi MEC traffic considering of **Network Metric**
- Case 2: Scheduling Multi MEC traffic considering of **Computing Metric**
- Case 3: Centric VS Distributed resolution
- Case 4: Sync interval impact

- Gain during dynamic changing status (e.g.: changing server capacity, dynamic service traffic etc.)
- To achieve good performance, fast sync of status change is important
Leveraging anycast in context of SFC/NFV,
  but could be also extended to others e.g. edge computing

Control Plane:
  Implemented on OSPF (BGP planned)

Experiment: Successfully distribute load
  Metric: link cost + NVF load
  Topologies with hundred of nodes selected from: https://sites.uclouvain.be/defo/
  Topologies reproduced on Grid'5000 large-scale and flexible testbed using VxLAN
    • https://www.grid5000.fr/w/Grid5000:Home
Summary

• Two-D feature: Dynamic & Distributed
• Dynamic anycast (Dyncast)
  • Identify a service at network layer
  • Consider computing load info, not always least cost
  • Dispatch on-the-fly, late binding of egress edge
  • Ensure flow affinity
• Control plane: BGP/IGP extension, any other protocol?
• Data plane: binding table, data encap/forwarding
Q & A during the side meeting

• Q: What is “service”? Is service placement in the scope?
  • A: Service placement and selecting path to which service instance are separate, this proposal is mainly emphasizing on the latter

• Q: Relationship with COINRG
  • A: Will present to COINRG on Friday
  • A: This proposal is focusing on routing optimization
Thank you!