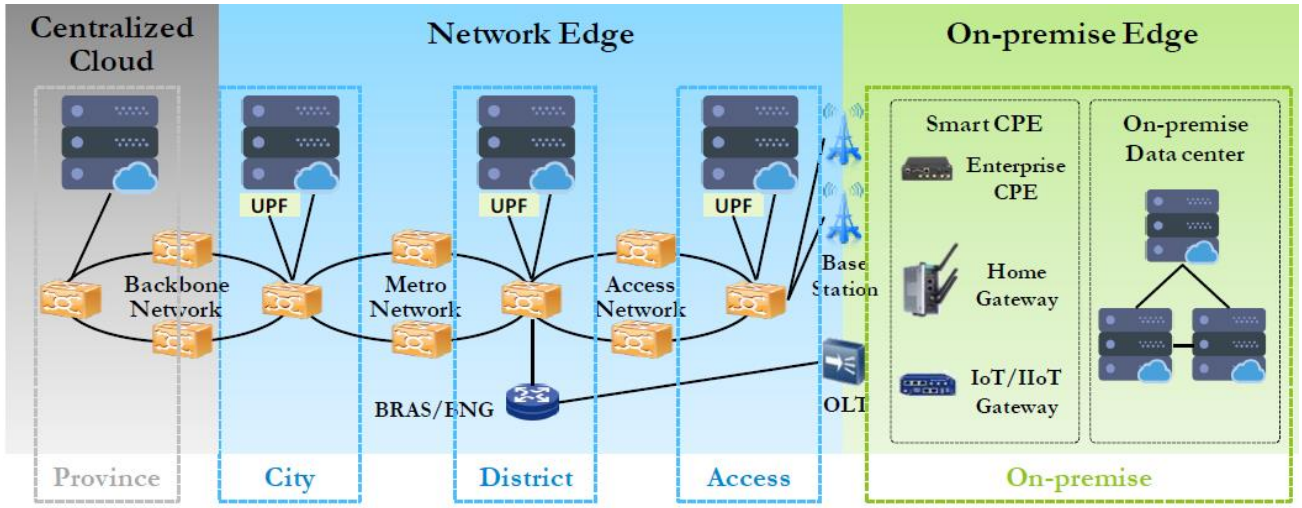
CFN (Computing First Network) Framework

draft-li-rtgwg-cfn-framework-00

Typical Multi-edge Computing Usage Scenario & Problems



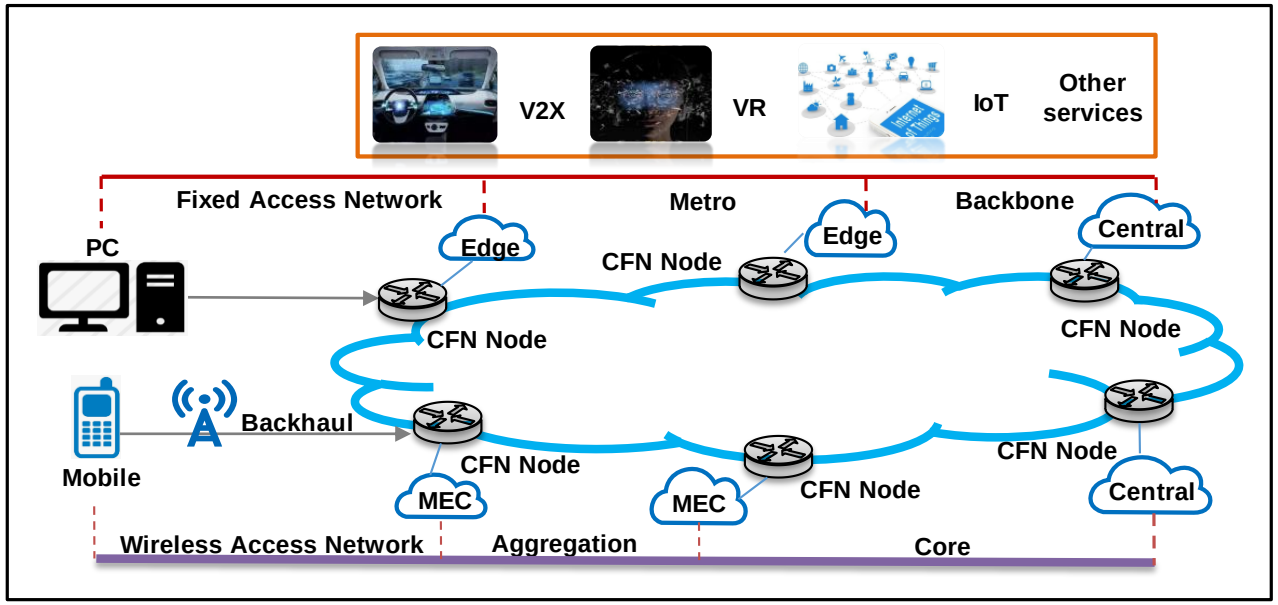
- Computing resources located on edges:
  - Large number of edges
- Edges have different characteristics
  - Computing capacity & load
- Computing characteristics unknown to network
- Question: Which edge is the best to serve a request?
- Problems:
  - Resource point of view: unbalance, hotspot…
  - User point of view: experience goes bad, longer job completion time

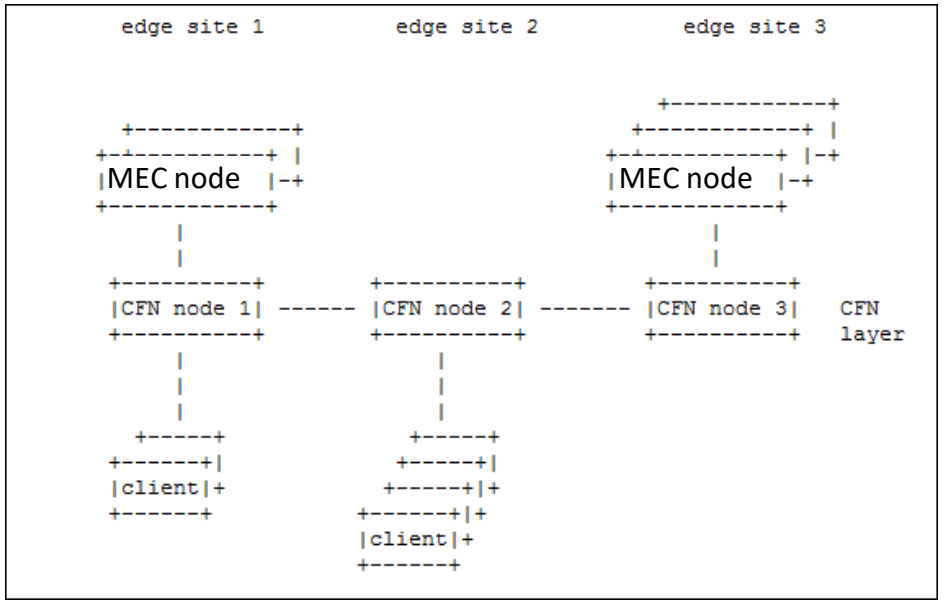
Current Practices and Gaps

- One or more mechanisms used:
  - Use least network cost, computing load is not a concern
    - Computing load is a key to consider in edge computing
  - Use geographical location, pick closest
    - Edges are not so far apart. Location matters but may not matter most.
  - Health check in an infrequent base (>1s), switch when fail-over
    - Limited computing resources on edge, change rapidly (<1s)
  - Random pick, network cost is not a concern
    - Edges are not deployed in equal cost way like in DC, network status is a concern
  - Early binding: query first and then steer traffic. Normally in a centralized way.
    - Edge computing flow can be short. Early binding has high overhead.
  - Single request/reply based scheduling
    - Multi-edge computing requires flow affinity.

CFN Network Topology

CFN is a networking infrastructure to interconnect multiple edges and provide service dispatch and forwarding based on both computing and networking status to the most appropriate edge





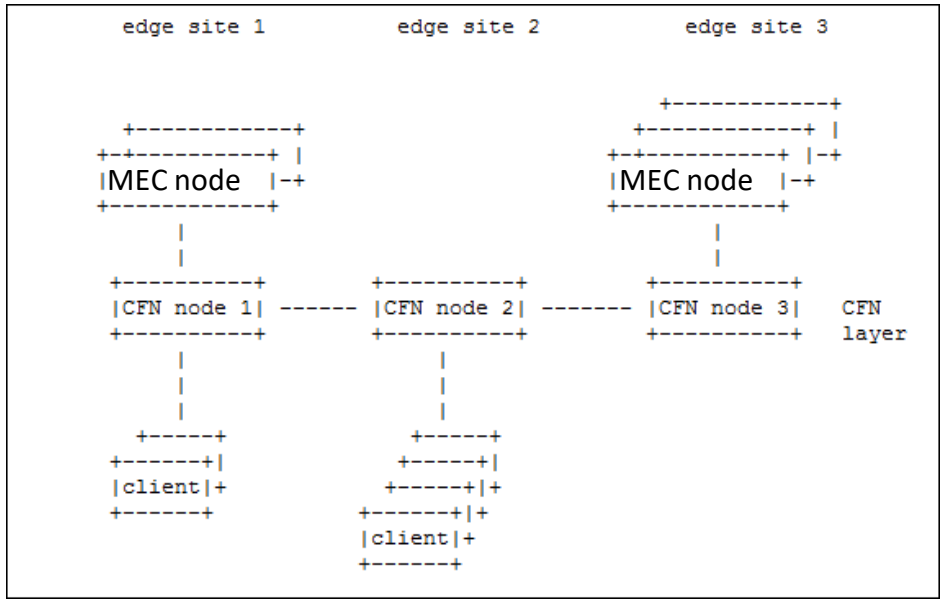
- MEC node:

serve requests, limited computing resource

- CFN node:

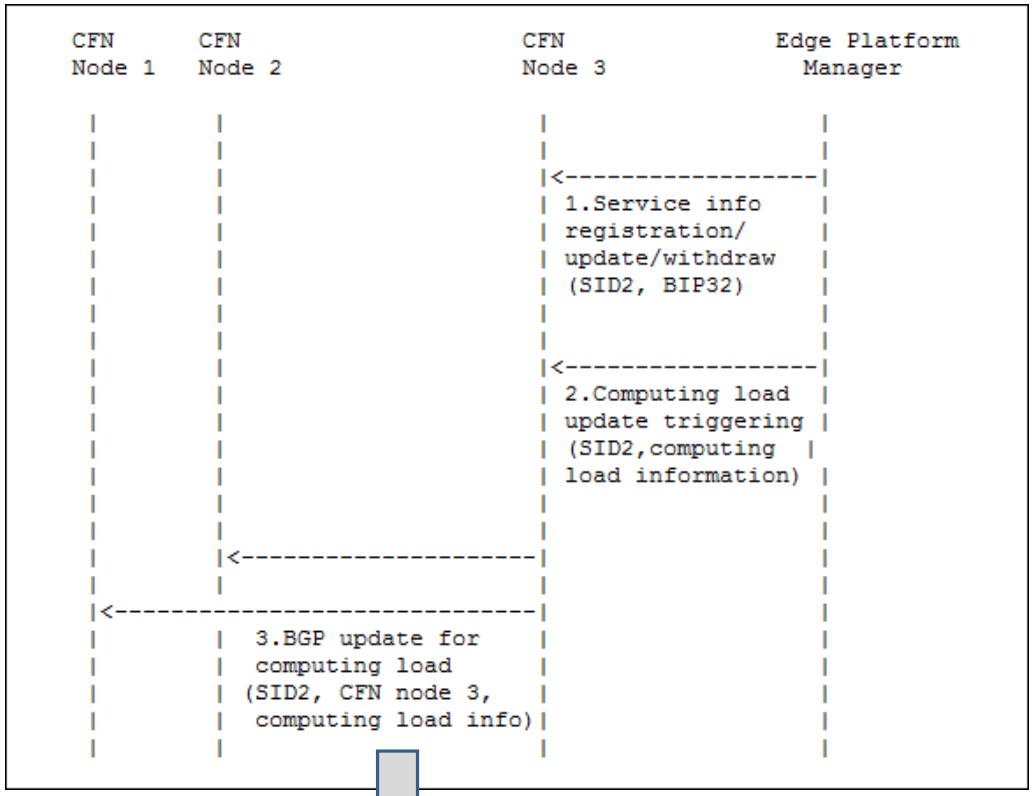
with computing resource locally attached, and/or handles clients' request

CFN Framework



- 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

Example: Control Plane

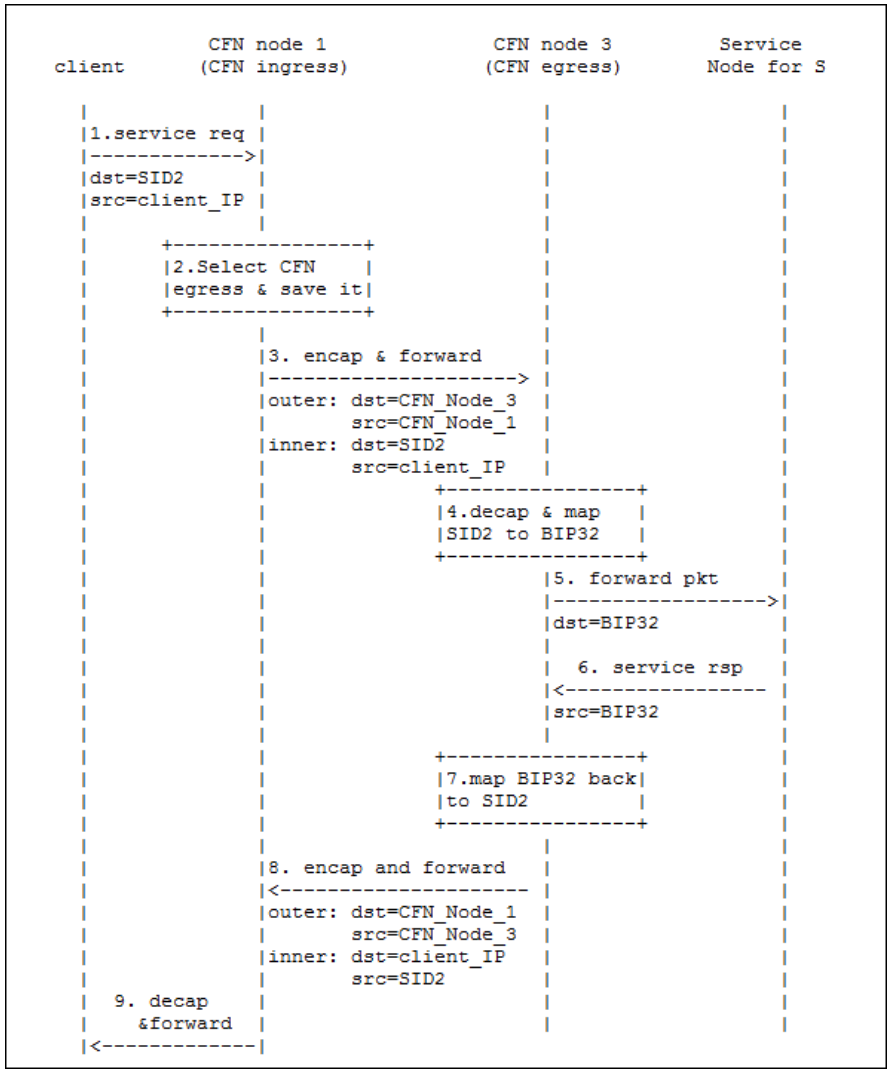


Destination	Computing Load	Network Cost	Next Hop
SID 2	30	50	CFN Egress node 3

- 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

Example: Data Plane



Data plane for the first request

- 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, to save memory
- 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

Summary

- Two-D feature: Dynamic & Distributed
- Dynamic anycast (Dyncast)
  - Anycast address to identify an MEC service
  - Consider computing load info
  - Dispatch on-the-fly, late binding of edge
  - Ensure flow affinity
- Work in IETF:
  - Control plane: BGP/IGP extension, any other protocol?
  - Data plane: binding, data encap/forwarding

backup

Inter-region service access