Problem Statement & Use Cases of CATS

draft-yao-cats-ps-usecases-00

K. Yao, China Mobile
P. Eardley
D. Trossen, Huawei
M. Boucadair, Orange
LM. Contreras, Telefonica
C.Li,Y.Li,Huawei
P. Liu, China Mobile

Draft status

The CATS WG is chartered to work on the following items:

o Groundwork may be documented via a set of informational Internet-Drafts, not necessarily for publication as RFCs:

* Problem statement for the need to consider both network and computing resource status.

* Use cases for steering traffic from applications that have critical SLAs that would benefit from the integrated consideration of network and computing resource status.

** Copied from CATS Charter*

Table of Contents

Introduction

- Multiple service instances on geographically distributed edge sites are provided to meet different service requirements
  - Users want the best user experience, expressed through **low latency** and **high reliability,** etc. .
  - Users want **stable** service experience when moving among different areas and in times of changing demand.
- How to meet user requirements?
  - **Deploy instances for the same service across various edge sites for better availability**
    - Provide functional equivalency
  - **Steer traffic dynamically to the "best" service instance**
    - Traffic is delivered to optimal edge sites based on information that includes computing information
    - The definition of 'best' may be service-specific
- However, the **problem** is **the "closest" might NOT be the "best"**
  - The closest site may not have enough resources, particularly when load fluctuates.
  - The closest site may not have enough specific resources, e.g., support for specific HW or SW.

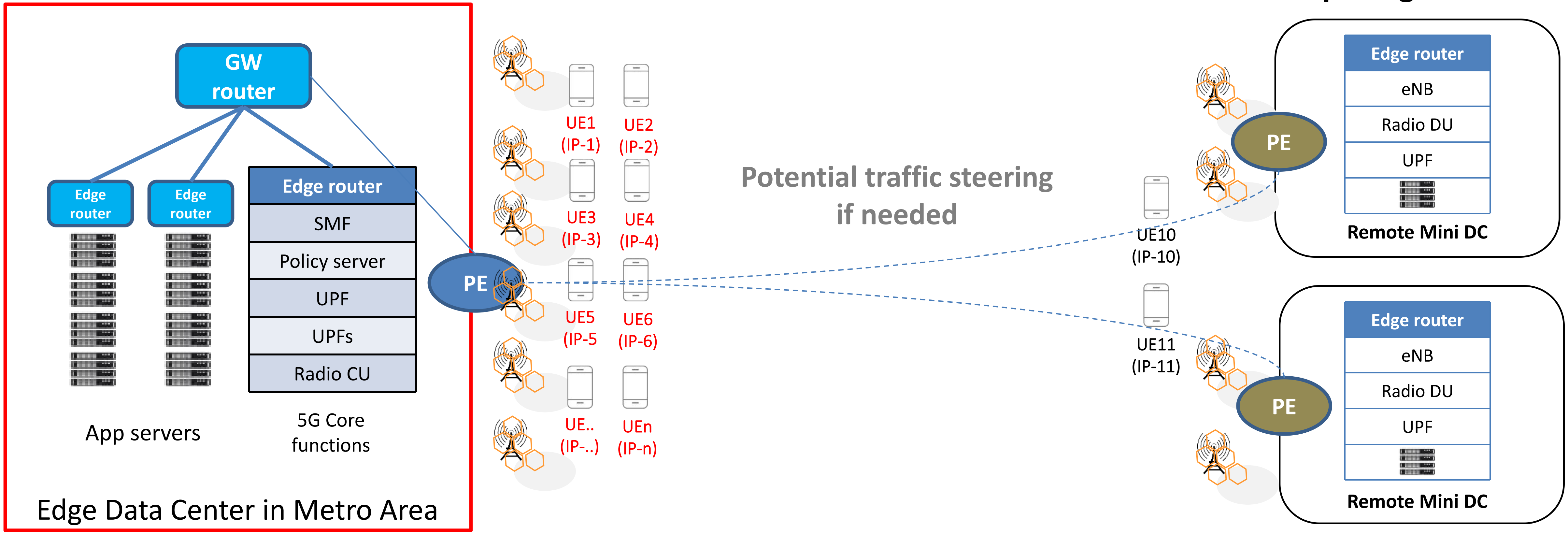
Problem Statement

High computing resources allocated at Metro Edge DCs

(for large numbers of UEs at working time)

- **Many UEs in Metro Area**
- **High computing resource**

- **Few UEs close to remote edge**
- **Limited computing resource**



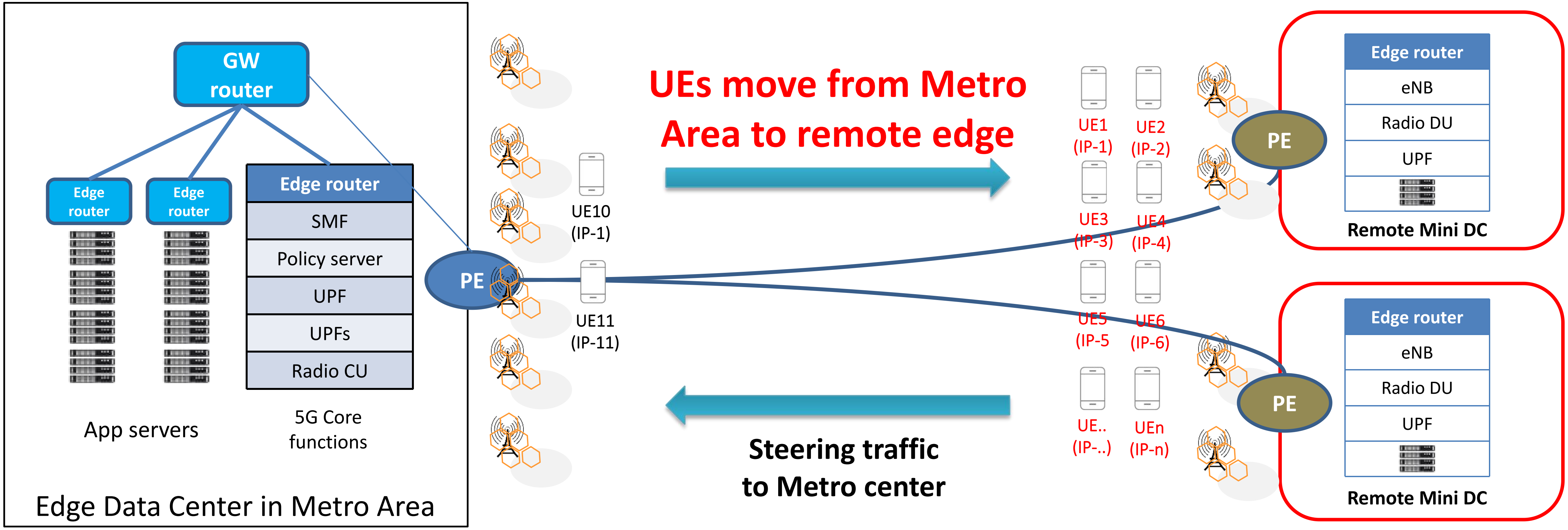
Problem Statement

Weekend events at a remote site require high computing usage

(only for 1~2 days, can't justify adding servers to the remote site)

- **Few UEs in Metro Area**
- **High computing resource**

- **Many UEs close to remote edge**
- **Limited computing resource**



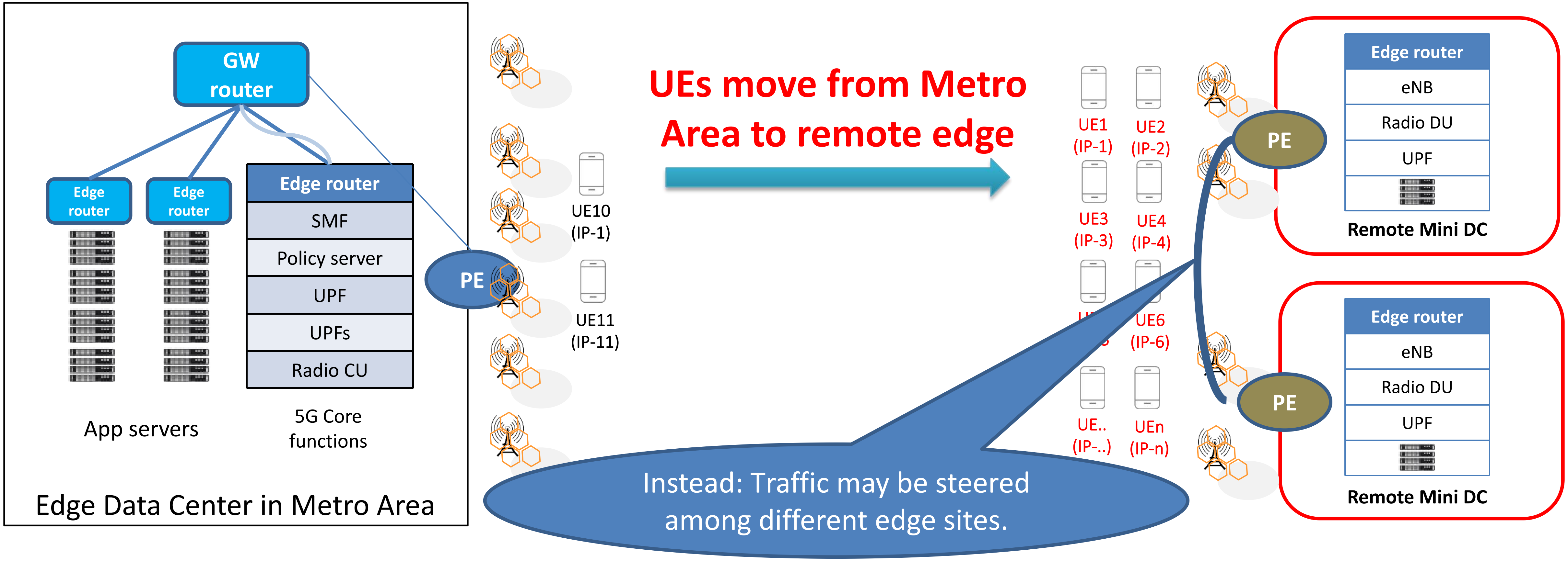
Problem Statement

Sudden events at a remote site require high computing usage

(unplanned and brief occurrence, thus can neither justify adding servers to the remote site)

- **Few UEs in Metro Area**
- **High computing resource**

- **Many UEs close to remote edge**
- **Limited computing resource**



Traffic may be steered among different edge sites.

- High computing resources needed by UEs at a remote site for short period of time, which is not long enough to justify adding more computing resources at the remote site.

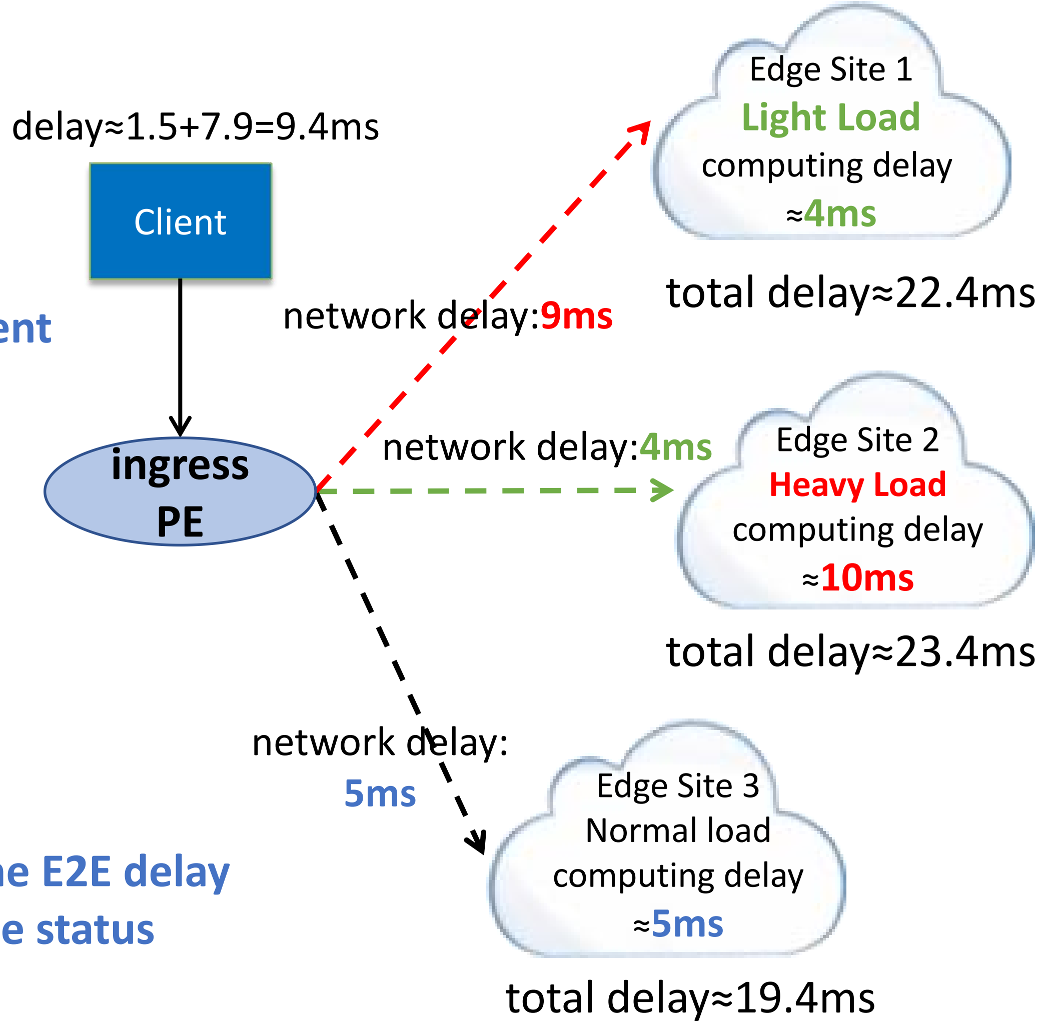
When steering traffic, what factors should be considered?

- Some apps require both low latency and high computing resource usage or specific computing HW capabilities (such as GPU);
- hence **joint optimization** of network and computing resources may be needed to guarantee the QoE.

Use Cases: Computing-Aware AR/VR

Upper bound latency for motion-to-photon(MTP): less than **20ms** to **avoid motion sickness,** consisted of:

1. sensor sampling delay: <1.5ms (client)
2. display refresh delay: ≈7.9 ms(client)
3. frame rendering computing delay with **GPU**≈ **5.5ms** (server)
4. network delay(budget) =20-1.5-7.9-5.5 = **5.1ms**(network)



Budgets for computing delay and network delay are almost equivalent

\+

- choose edge site 1 according to load only, total delay≈22.4ms
- choose edge site 2 according to network only, total delay≈23.4ms
- **choose edge site 3 according to both, total delay≈19.4ms**

Only according to the network or computing resource status, can not find the "best" server instance

↓

Require to dynamically steer traffic to the appropriate edge to meet the E2E delay requirements by considering both network and computing resource status

PS: Compute resources vary greatly at different edges, and "closest site" may be good for latency, but lacks GPU support and therefore should not be chosen.

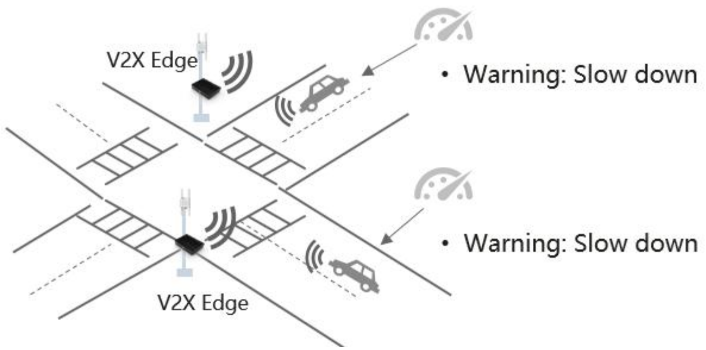
Use Cases:Computing-Aware V2X

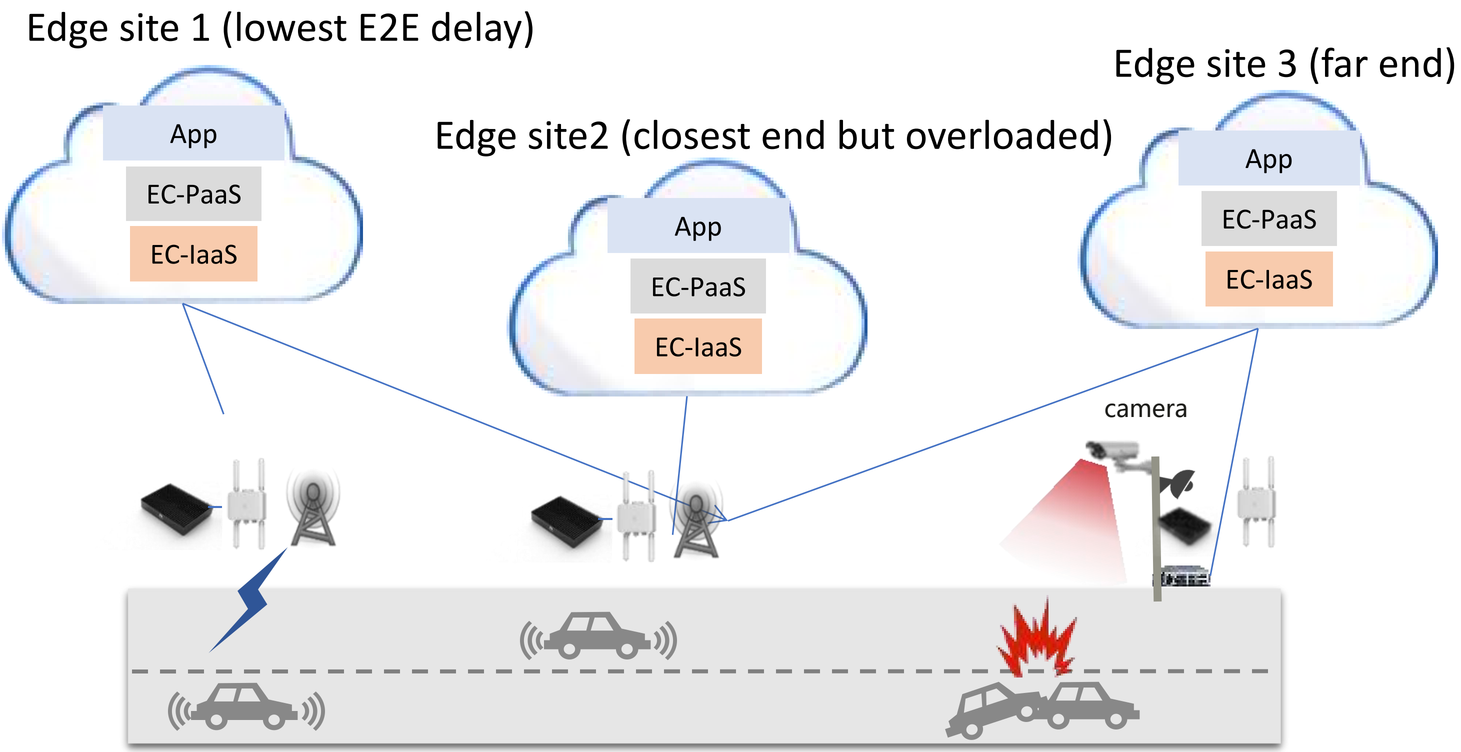
Autonomous driving

Function	Requirement
Driving-assist	**Low Latency**
HD and HP Map	**High bandwidth**

Video recognition at intersection

Function	Requirement
Safety Monitoring	**Low Latency**
Data analysis	**High bandwidth**





Shorter latency, better safety.

For example. If the latency is reduced by 100 ms,
the braking distance of a vehicle at 80 km/h can be reduced by **2.2 meter**.

The load of network and edge sites may change **dynamically and rapidly**

Conclusions

Those apps require both **low latency** and **high/specific computing resources** have the almost **equivalent budgets** for computing delay and network delay, and the load of network and edge sites may **change dynamically and rapidly.**

When steering traffic, the real-time **network and computing** resource status should be considered **simultaneously** in an effective way.

Next Steps

Welcome more discussion and contribution!

Thank you!