Latency-aware Virtual Machine Placement for Mobile Edge Computing

Wei Wang

BUPT Ph.d candidate & UC Davis visiting student
Email: weiw@bupt.edu.cn, waywang@ucdavis.edu

UCDAVIS

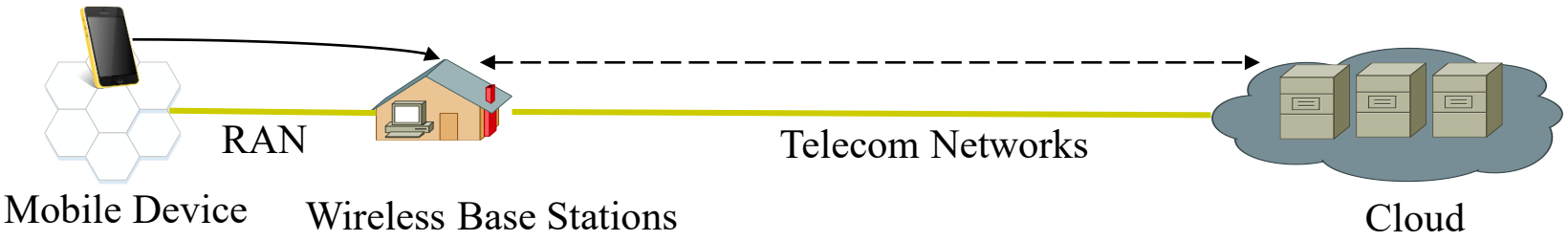
Group Meeting, Feb. 10, 2017

Contents

- Introduction
- Existing works
- Problem Statement
- Problem Formulation
- Future works

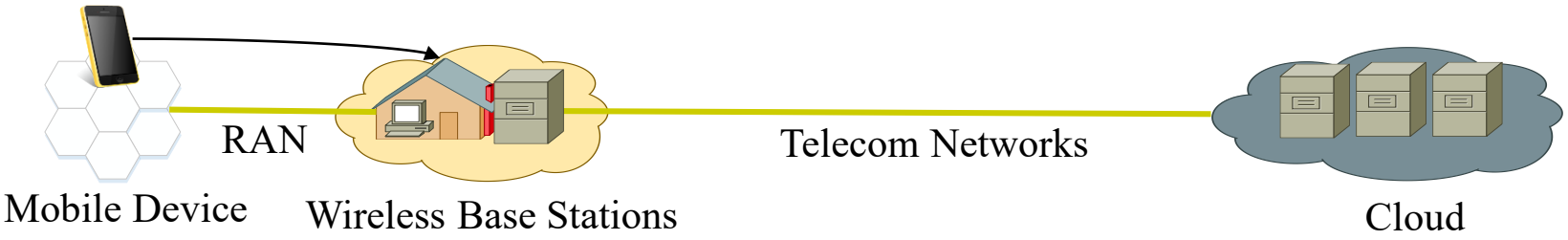
Introduction

- Cloud Computing
  - Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand.
  - Computer processing resources and data are usually deployed in centralized datacenters, which is far away from end users.
  - Drawbacks, long-distance network connection between user and cloud result in long service latency.



Introduction

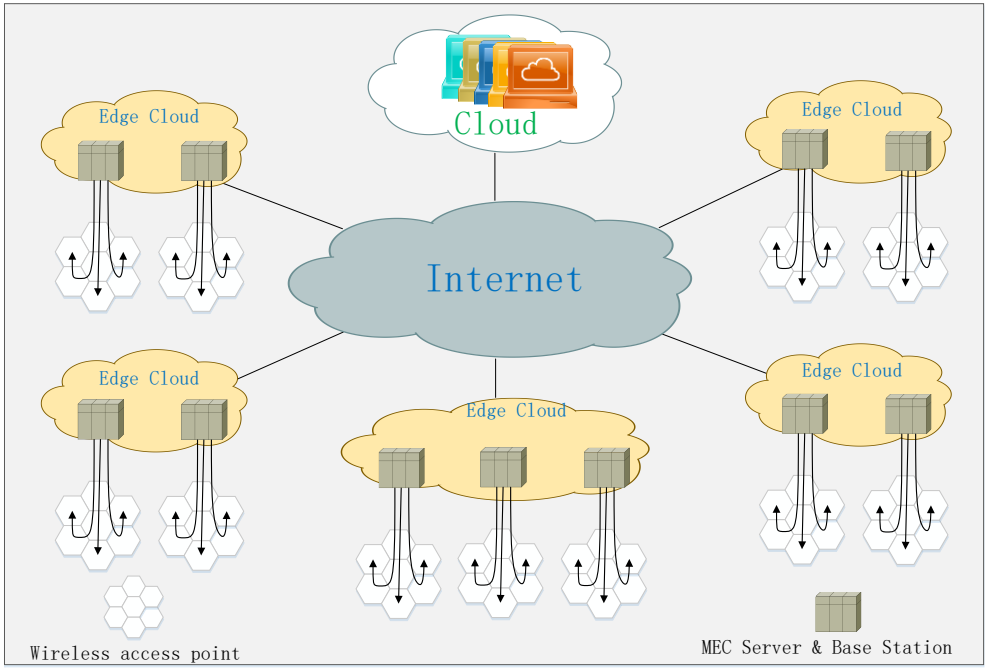
- Mobile Edge Computing(MEC)
  - Mobile Edge Computing provides an IT service environment and cloud-computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile subscribers. The aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience.[1]



[1] Hu, Yun Chao, et al. "Mobile edge computing—A key technology towards 5G." *ETSI White Paper* 11 (2015).

Introduction

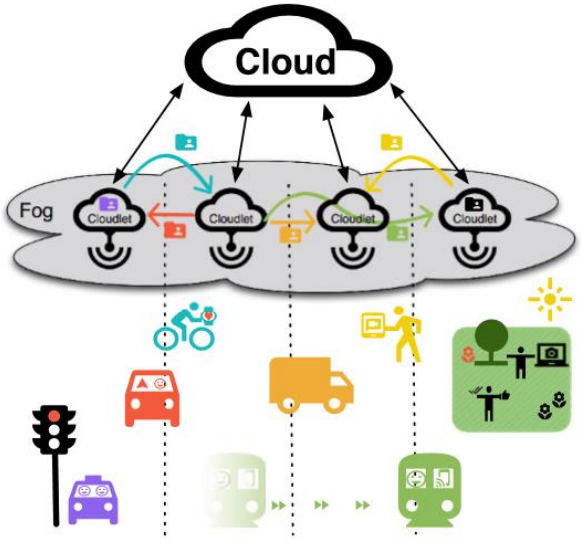
- MEC cloud and overall MEC system
  - Mobile operators are working on Mobile Edge Computing (MEC) in which the computing, storage and networking resources are integrated with the base stations.[2]



[2] Lav Gupta and Raj Jain, Mobile Edge Computing – An Important Ingredient of 5G Networks, IEEE Software Defined Networks

Existing work

- Mobility-driven service migration
  - Problem: mobility of user may increase the distance between user and its VM, and thus increase latency.
  - Solution: migrate user's VM across MEC clouds dynamically when user moves.



Wang, Shiqiang, et al. "Dynamic service migration in mobile edge-clouds." *IFIP Networking Conference (IFIP Networking), 2015*. IEEE, 2015.
Bittencourt, Luiz Fernando, et al. "Towards virtual machine migration in fog computing." *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on*. IEEE, 2015.

Existing work (cont.)

- SLA-driven VM Scheduling in Mobile Edge Computing
  - Problem: Service providers' cost of renting VMs at Edge clouds is calculated as $/time unit. Each type of VM has its maximum capacity to handle request. If the number of requests exceeds its capacity, some requests will go to remote clouds, and thus cause penalty for violating SLA. How to reduce cost while minimizing service penalty?
  - Approach: LYAPUNOV OPTIMIZATION-BASED scheduling algorithm for deploying and releasing VMs dynamically.

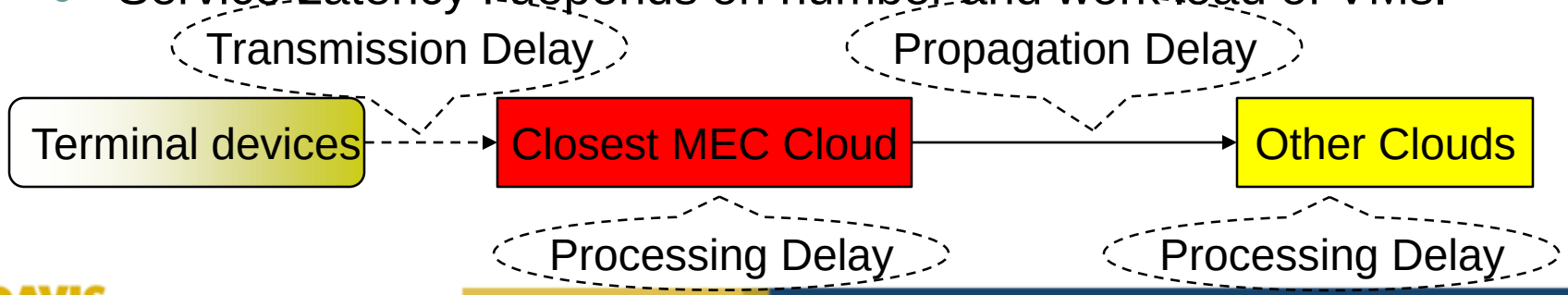
Katsalis, Kostas, et al. SLA-driven VM Scheduling in Mobile Edge Computing. 9th International Conference on Cloud Computing, IEEE, 2016

VM based service for MEC

- Role of VMs in MEC and central clouds
  - In general, user devices play as clients and MEC clouds perform as servers.
  - To accommodate users requests, MEC clouds need to provide not only the computing and storage ability, but also the service specific software and data, which can be packaged in Virtual Machines(VM).
- Options for service handling.
  - Handled at Local MEC Cloud
  - Handled at Other MEC Cloud
  - Handled at Centralized Cloud

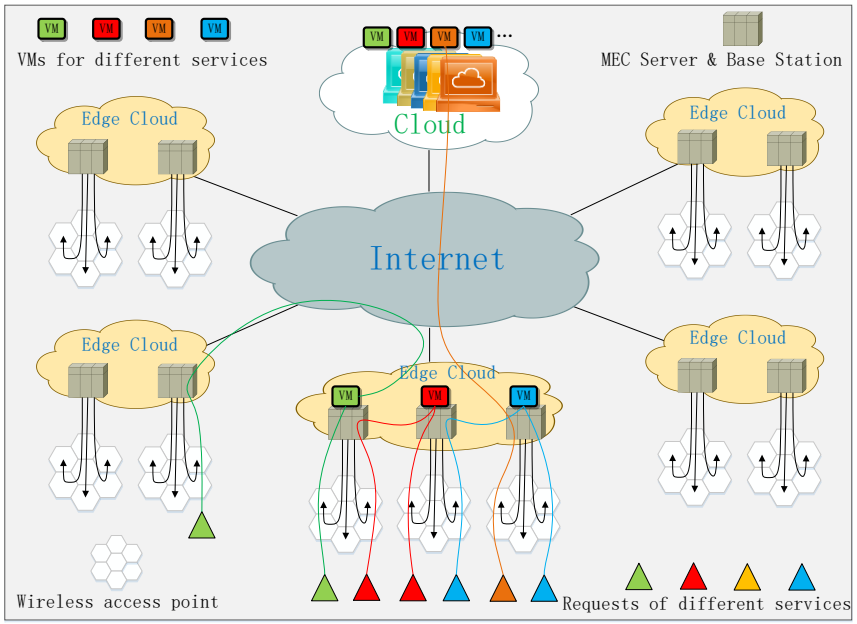
Latency of MEC service

- To get network service, users' requests need to go to base station first, and then find its destination VM to get service.
- Define: The MEC cloud, where the first hop base station locates at, is called origination cloud of request.
- These procedures will introduce network and processing latency, which is very critical for Quality of Experience.
- Major parts of overall Latency:
- Transmission Latency : depends on bandwidth;(out of scope)
- Propagation Latency : depends on length of network path;
- Service Latency : depends on number and work load of VMs.



VM's influence on Latency

- As the server for user's requests, the location of VMs has significant influence on the distance from user to server, and thus influence the propagation latency.
- The number of VMs decides average load of each VM, and is related with processing latency.



- Service routing map decides which VM is responsible for processing the requests originate from a specific Cloud, and thus has influence on both propagation and processing latency.

Problem statement

- Problem Description:
  - MEC clouds are already deployed at network edge, and each cloud has certain hardware resource to run VMs.
  - Network distance (latency) between each MEC cloud pair is known.
  - Several kinds of MEC service are already known, and each kind of service has its expected latency threshold.
  - Each kind of service corresponds to a specific kind of server VM, which has certain capacity of handling requests.
  - Expected request load of each kind of service that originates from each MEC cloud is known.
  - How to place VMs at each MEC cloud for each kind of service to meet their latency requirements?

Problem Formulation

- Given:
  - E: Set of edge clouds
  - $\mathrm{L}_{\mathrm{e1},e2}$: Network propagation latency from $\mathrm{e1} \in \mathrm{E}$ to $\mathrm{e2} \in \mathrm{E}$
  - C_e: Hardware capacity of edge cloud $\mathrm{e} \in \mathrm{E}$

  - S: Set of services
  - R_s: Computing capacity required to deploy a VM for service $s \in S$
  - T_s: Expected latency requirement of service $s \in S$
  - u_s: Handling capacity at VM for service $s \in S$
  - λ_{e}^{s}: Request load of service $s \in S$ that originates from cloud $\mathrm{e} \in \mathrm{E}$

Problem Formulation (cont.)

- Variables:
  - n_e^s: How many VMs need to be deployed for service $s \in S$ at edge cloud $\mathrm{e} \in \mathrm{E}$.
  - $\mathrm{x}_{\mathrm{src},dst}^s$: Whether the requests of service $s \in S$ from $\mathrm{src} \in \mathrm{E}$ are processed by edge cloud $\mathrm{dst} \in \mathrm{E}$.
- Object:
  - Minimize required hardware resource (cost) for deploying VMs.

$$\sum_{\mathrm{e} \in E} \sum_{s \in S} \mathrm{n}_{\mathrm{e}}^s * \mathrm{R}_s$$

Problem Formulation (cont.)

- Constraints about hardware capacity:
  - (1) Required resource for all VMs at each MEC cloud $e \in E$ should not exceed its hardware capacity.

$$\sum_{e \in S} n_e^s * R_s < C_e, \forall e \in E$$

Problem Formulation (cont.)

- Constraints about service strategies:
  - (2) Each kind of service $s \in S$ needs at least one corresponding VM all overall the MEC clouds.

$$\sum_{e \in E} n_e^s \geq 1, \forall s \in S$$

  - (3) The requests of service $s \in S$ from one MEC cloud $src \in E$ should be processed by one destination cloud $dst \in E$.

$$\sum_{dst \in E} x_{src,dst}^s = 1, \forall s \in S, \forall src \in E$$

Problem Formulation (cont.)

- Constraints about service strategies (cont.):
  - (4) There must be VM(s) to handle service $s \in S$ at MEC cloud $dst \in E$, if there are requests of s being routed to dst. And vice versa.

$$n_{dst}^{s} - \frac{\sum_{src \in E} x_{src,dst}^{s}}{M} \geq 0, \forall s \in S, \forall dst \in E$$

$$\sum_{src \in E} x_{src,dst}^{s} - \frac{n_{dst}^{s}}{M} \geq 0, \forall s \in S, \forall dst \in E$$

Problem Formulation (cont.)

- Constraints about service strategies (cont.):
  - (5) the requests of service $s \in S$ originate from MEC cloud $\mathrm{dst} \in \mathrm{E}$ should be processed locally, if there are VM(s) for service s deployed at dst.

$$\mathrm{n}_{dst}^{s} - \frac{\mathrm{x}_{dst,dst}^{s}}{M} \geq 0, \forall \mathrm{s} \in \mathrm{S}, \forall \mathrm{dst} \in \mathrm{E}$$

$$\mathrm{x}_{dst,dst}^{s} - \frac{\mathrm{n}_{dst}^{s}}{M} \geq 0, \forall \mathrm{s} \in \mathrm{S}, \forall \mathrm{dst} \in \mathrm{E}$$

Problem Formulation (cont.)

- Constraints about service latency:
  - (6) Requests of service $s \in S$ originates from $\mathrm{src} \in \mathrm{E}$ can not be processed at a dst, to where the propagation latency exceeds the threshold latency of service s.

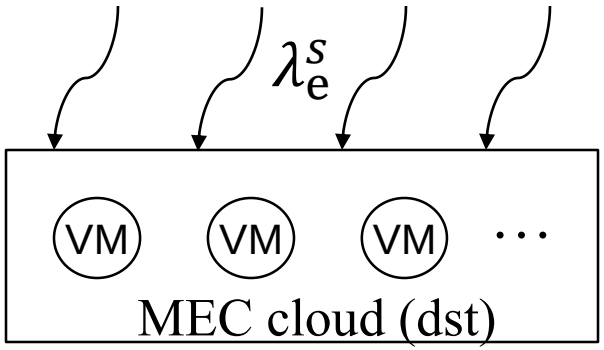
$$\left(T_s - \mathrm{L}_{src,dst}\right) * \mathrm{x}^{s}_{src,dst} \geq 0, \forall \mathrm{s} \in \mathrm{S}, \forall \mathrm{src} \in \mathrm{E}, \forall \mathrm{dst} \in \mathrm{E}$$

- Processing and queueing Latency?

Problem Formulation (cont.)

- Processing and queueing latency at a VM:
  - Queueing Theory. M/M/1 System with Poisson Process
  - n_e^s: number of VMs for service s at e.
  - $\sum_{src \in \mathrm{E}} \lambda_{src}^s * \mathrm{x}_{src,dst}^s$: overall load for service s at dst.
  - u_s: departure rate of service s at one VM.
  - Assumption: requests to one cloud are distributed evenly to all VMs, the system would be n_e^s M/M/1.
  - Expected average service latency is:

$$\frac{1}{\mathrm{u}_s - \frac{\sum_{src \in \mathrm{E}} \lambda_{src}^s * \mathrm{x}_{src,dst}^s}{\mathrm{n}_e^s}}$$



Problem Formulation (cont.)

- Constraints about service latency (cont.):
  - (7) The average arrive rate of each service $s \in S$ to a VM should not exceed its departure rate, so that the queueing system is stable.

$$\mathrm{u}_s * \mathrm{n}_{dst}^s - \sum_{src \in \mathrm{E}} \lambda_{src}^s * \mathrm{x}_{src,dst}^s \geq 0, \forall \mathrm{s} \in \mathrm{S}, \forall \mathrm{dst} \in \mathrm{E}$$

  - (8) The overall latency of the farthest customer should not exceed the required threshold latency.

$$\mathrm{L}_{max}(s, dst) + \frac{1}{u_s - \frac{\sum_{e \in \mathrm{E}} \lambda_s^{\mathrm{e}} * \mathrm{x}_{e,dst}^s}{\mathrm{n}_{dst}^s}} \leq T_s, \forall \mathrm{s} \in \mathrm{S}, \forall \mathrm{dst} \in \mathrm{E}$$

Results

```
set EDGECLOUDS := SFO LAX NYC MCO PEK HKG NRT; # BOM LCY FRA;
set SERVICES := AR MAP GAME FACERECG VEDIO SOCIAL SHOPPING;# BACKUP PAY TRAVEL;
```

```
param: cloudCapacity:=
SFO     90
LAX    100
NYC    100
MCO     80
PEK     90
HKG    100
NRT    100;
#BOM    100
#LCY    100
#FRA    100;
```

```
param:     muPerVM:=
AR         0.02
MAP        0.025
GAME       0.02
FACERECG   0.03
VEDIO      0.035
SOCIAL     0.04
SHOPPING   0.035;
#BACKUP     0.04
#PAY        0.03
#TRAVEL     0.045;
```

```
param netLatency:
     SFO LAX NYC MCO PEK HKG NRT:=# BOM LCY FRA:=
SFO    0   9   85  75 185 159 125 #280 151 158
LAX    9   0   71  68 170 157 121 #273 136 153
NYC   85  71    0  34 284 203 157 #203  67  82
MCO   75  68   34   0 322 235 164 #238  96 109
PEK  185 170 284 322   0 127  87 #260 187 390
HKG  159 157 203 235 127   0  49 #279 239 268
NRT  125 121 157 164  87  49   0; #364 209 273
#BOM 280 273 203 238 260 279 364   0  142 135
#LCY 151 136  67  96 187 239 209 142   0   16
#FRA 158 153  82 109 390 268 273 135  16   0;
```

```
param:     vmCapacity:=
AR         3
MAP        3
GAME       5
FACERECG   3
VEDIO      7
SOCIAL     2
SHOPPING   4;
#BACKUP     8
#PAY        4
#TRAVEL     3;
```

```
param:     requiredLatency:=
AR         150
MAP        250
GAME       180
FACERECG   200
VEDIO      350
SOCIAL     300
SHOPPING   350;
#BACKUP     500
#PAY        250
#TRAVEL     350;
```

```
param lambda(tr):
            SFO   LAX   NYC   MCO   PEK   HKG   NRT:=#   BOM   LCY   FRA:=
AR          0.01  0.02  0.03  0.015 0.03  0.02  0.03  #0.02  0.015 0.015
MAP         0.02  0.025 0.04  0.02  0.03  0.025 0.03  #0.015 0.015 0.018
GAME        0.01  0.03  0.015 0.025 0.035 0.015 0.01  #0.01  0.017 0.021
FACERECG    0.03  0.01  0.023 0.018 0.026 0.015 0.008 #0.011 0.013 0.015
VEDIO       0.013 0.021 0.023 0.011 0.034 0.016 0.03  #0.015 0.026 0.031
SOCIAL      0.042 0.025 0.050 0.031 0.042 0.017 0.021 #0.03  0.01  0.012
SHOPPING    0.012 0.021 0.033 0.009 0.032 0.044 0.03; #0.029 0.045 0.025
#BACKUP      0.013 0.03  0.04  0.021 0.015 0.021 0.033 0.035 0.021 0.012
#PAY         0.05  0.025 0.033 0.011 0.048 0.023 0.037 0.036 0.018 0.029
#TRAVEL      0.02  0.03  0.028 0.041 0.032 0.045 0.055 0.023 0.031 0.027;
```

Results

- n_e^s

vmNum [*,*]

:	AR	FACERECG	GAME	MAP	SHOPPING	SOCIAL	VEDIO	:=
HKG	2	1	3	0	0	4	0	
LAX	2	0	0	2	2	3	0	
MCO	4	1	4	1	0	0	0	
NRT	3	2	0	5	0	0	5	
NYC	0	3	0	2	0	0	0	
PEK	3	0	3	0	1	0	0	
SFO	1	0	3	1	3	0	0	

- $x_{\text{src},dst}^s$

[NYC,*,*]

:	AR	FACERECG	GAME	MAP	SHOPPING	SOCIAL	VEDIO	:=
HKG	0	0	0	0	0	0	0	
LAX	0	0	0	0	0	1	0	
MCO	1	0	1	0	0	0	0	
NRT	0	0	0	0	0	0	1	
NYC	0	1	0	1	0	0	0	
PEK	0	0	0	0	0	0	0	
SFO	0	0	0	0	1	0	0	

Future work

- Case A: Initial placement
  - Place VMs for multiple services on ALL empty MEC clouds.
- Case B: Incremental placement
  - Place new VMs for one or multiple services on MEC clouds, which already have other VMs.
- Case A and B can be covered by above formulations
- Case C: Dynamic VM management.
  - Heuristics algorithms for service load change.
  - Options: 1)VM clone & migration, 2)VM exchange, 3)Service map optimization.

Thank you!

Wei Wang