Latency Aware Provisioning of Mobile Edge Computing (MEC) Services under MEC Server Congestion

Wei Wang

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

UCDAVIS

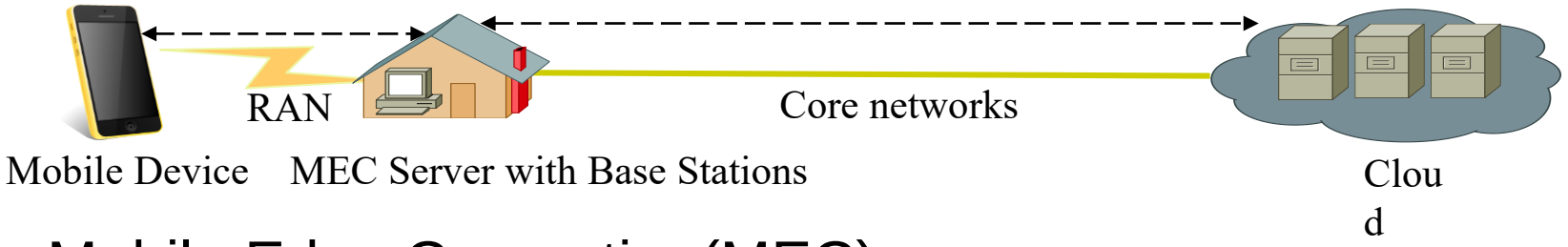
Group Meeting, December 16, 2016

Contents

- Introduction
- Problem statement
- Latency for MEC requests
- Heuristic algorithms
- ILP formulation

Mobile Edge Computing

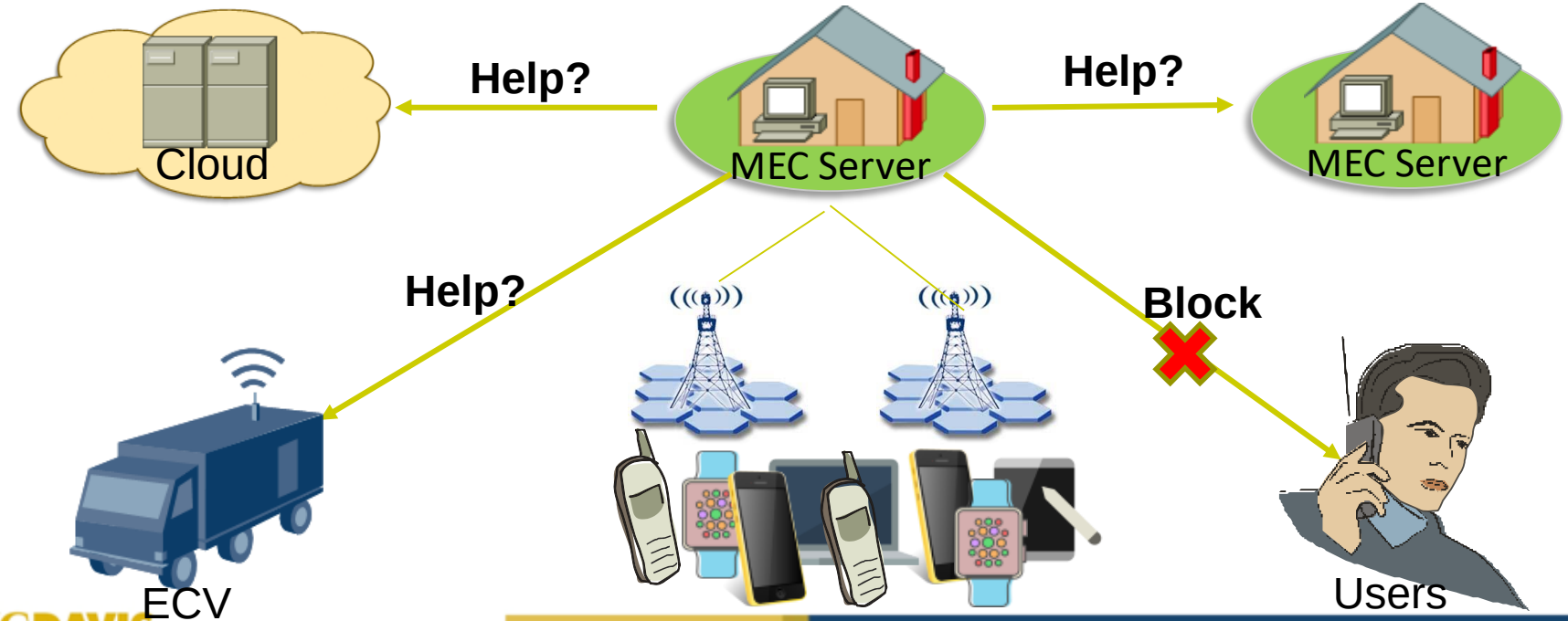
- Cloud computing
  - User requests from network edge get computing services from cloud datacenter via traveling through core networks.
  - Drawbacks, long-distance network connectivity results in long service latency and extra traffic in core networks.



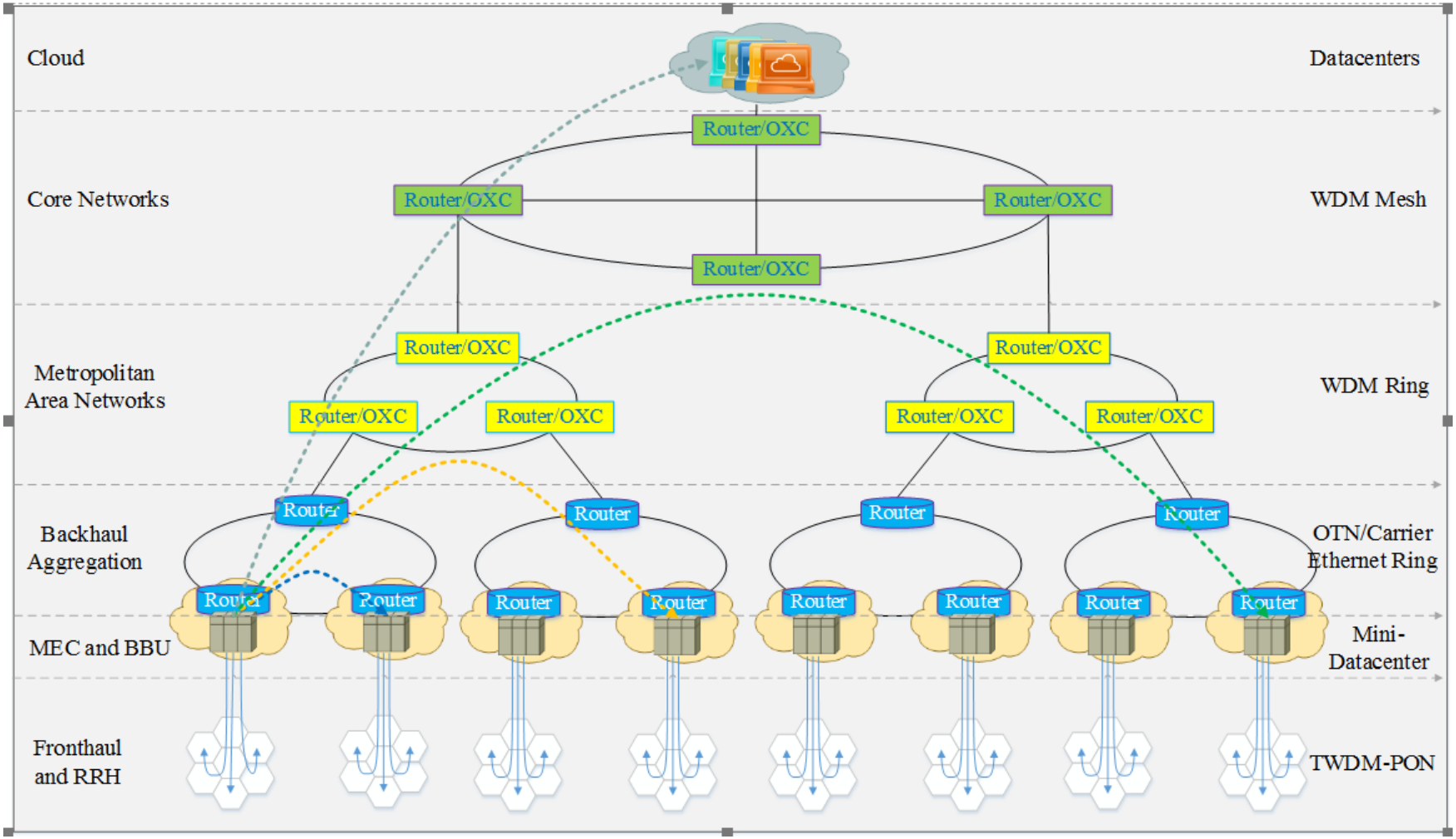
- Mobile Edge Computing(MEC)
  - As supplement to cloud computing, Mini-datacenters are deployed at mobile edge networks to provide computing services to nearby users via edge networks.
  - Limited-distance network connectivity reduces service latency and traffic in core networks.

Congestion of MEC resource

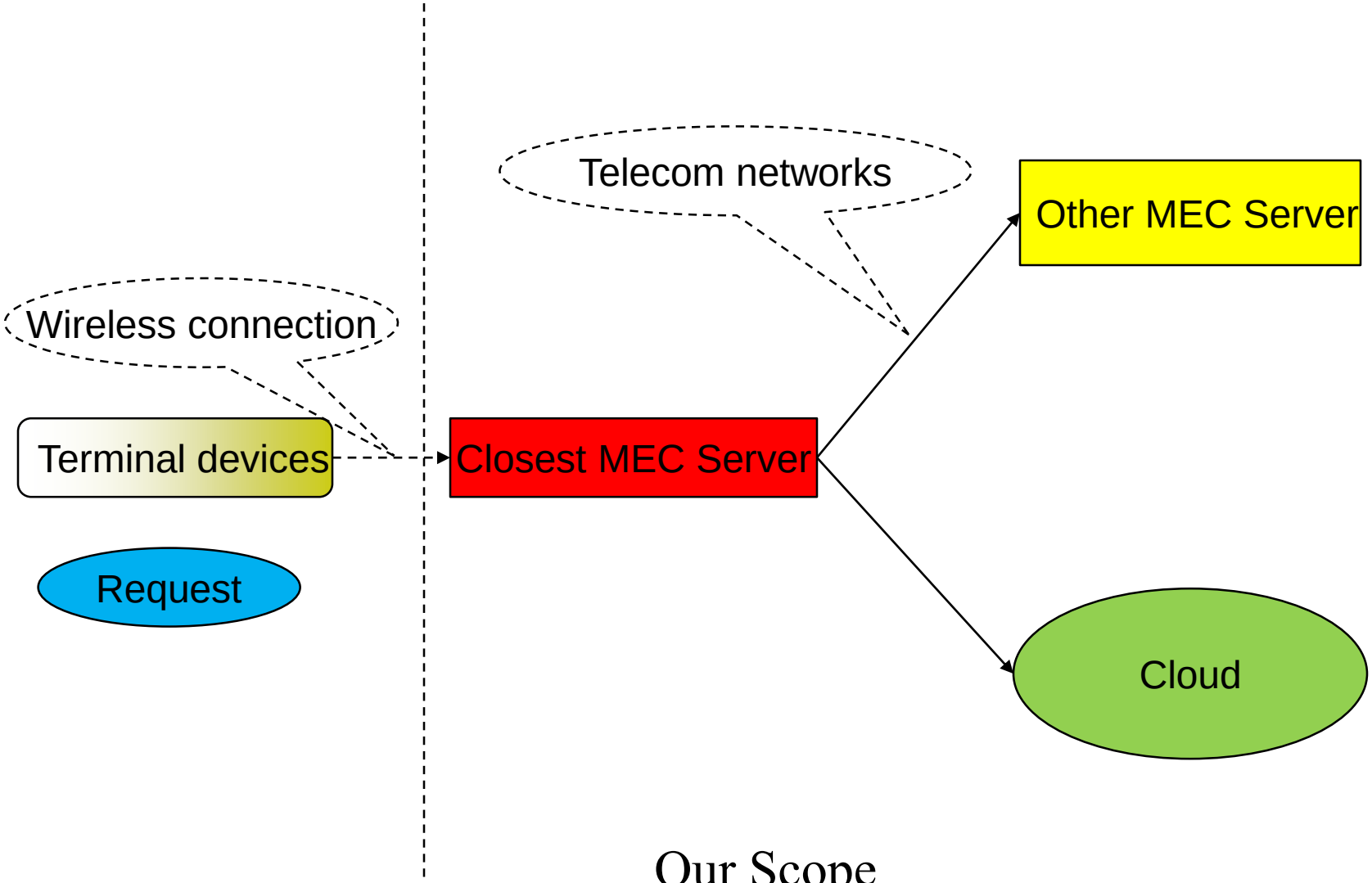
- Due to mobility of mobile users, some social events (e.g., Olympic Games) may attract large amount of users to one area, and thus cause congestion of mobile edge resource, including access ability of RAN and computing ability of MEC server.
- How to deal with congestion of edge computing resource?



Network and DC infrastructure for MEC applications



Provisioning of MEC applications

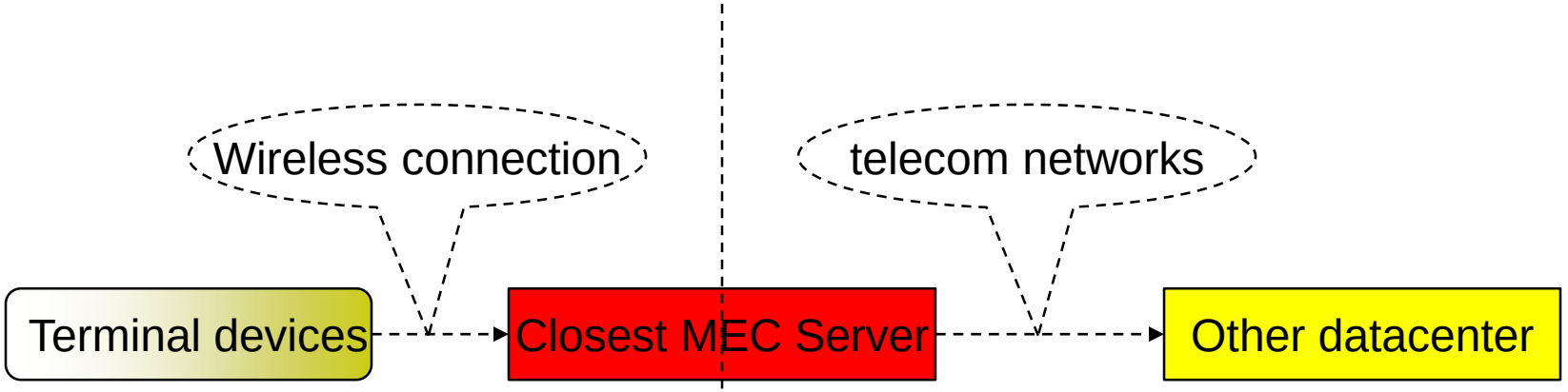


Problem statement

- For a request r, which originates from local server M, make the following decisions to achieve lowest average latency of all requests.
  - 1) whether it is processed by closest local MEC server?
  - 2) if not, which datacenter is destination to process r?
  - 3) if not, which path is the best to offload request r from M to its destination?

Network latency of MEC request

- Assumption A: no data queueing at any transit node.



Transmission delay
(depend on bandwidth)

Propagation delay(depend
on length of wireless path)

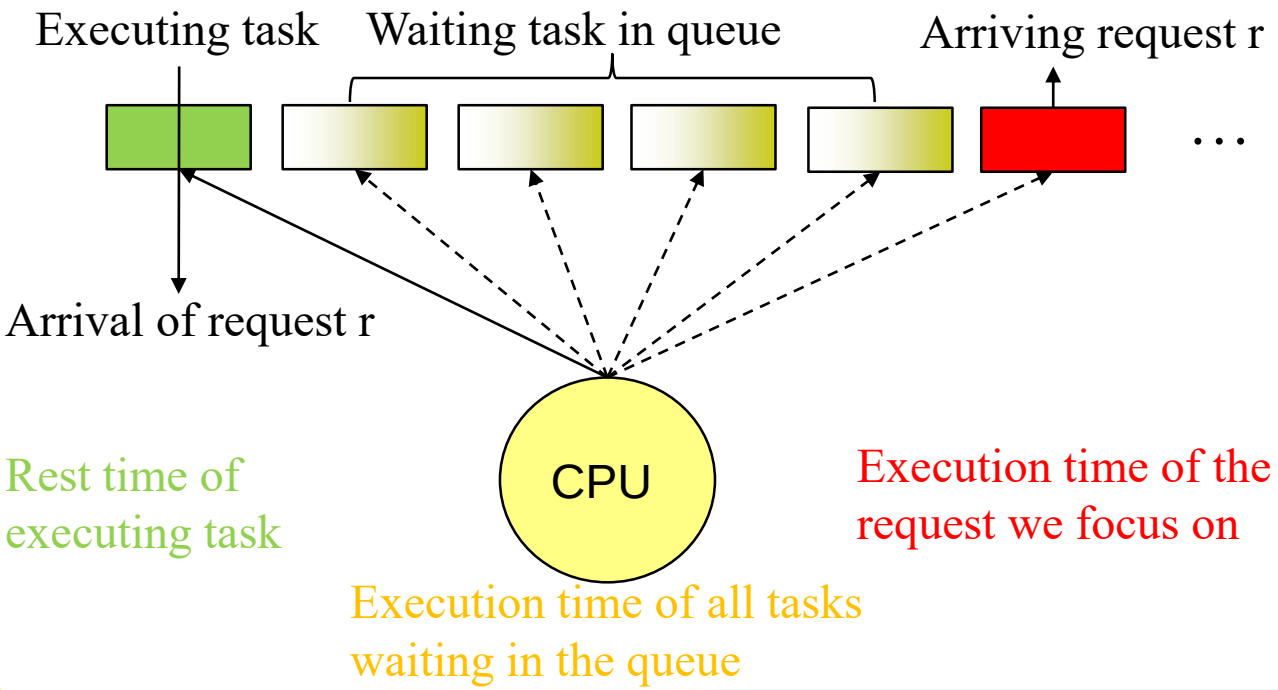
will not be affected by our solutions.

within our scope

Propagation delay of telecom networks
(in case of offloading, depends on
length of offloading path)

Computing latency of MEC request

- Assumption B: First Come, First Served (FCFS) Scheduling.
- Assumption C: One task queue in one datacenter.



Definition of infrastructures

- $\mathrm{G(V, E)}$, topology of networks; V is set of nodes and E is set of links.

- $V_{dc} \in V$, set of datacenter node locations.
- $\mathrm{W}_{i,j}$: bandwidth capacity of link $\mathrm{(i, j)} \in \mathrm{E}$, in units of bps.
- U_{dc}: computing capacity of datacenter $dc \in V_{dc}$, in units of operations per second.

Definition of MEC request

- $R = \{r\}$: set of requests
- For each request $r \in R$,
  - ghj

- $\mathrm{M_r}$: originate (closest) MEC server of r;
- $\mathrm{C_r}$: required computing resource of request r. (in unit of circles)
- $\mathrm{B_r}$: required bandwidth of request r.
- A_r: arrive time of request r.

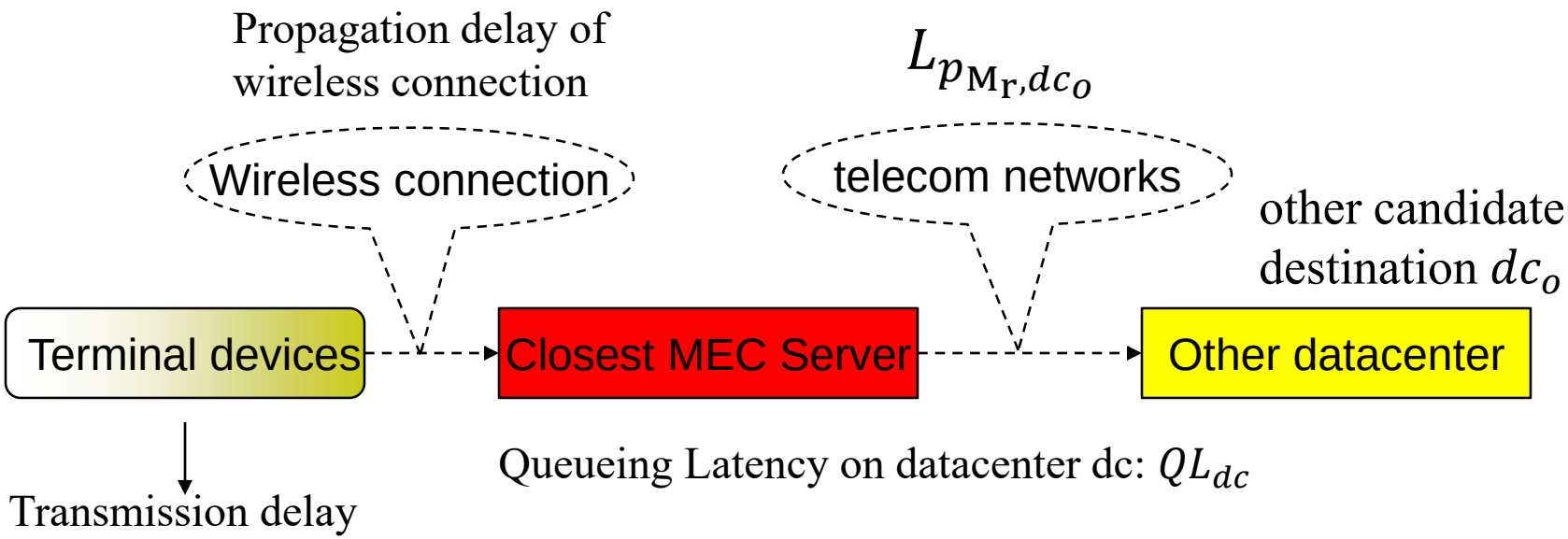
Auxiliary definitions

- $P_{s,d}^{k} = \{p_{s,d}\}$:k shortest paths between s and d.
- $L_{p_{s,d}}$:propagation latency of path $p_{s,d} \in P_{s,d}^{k}$, in unites of seconds.

- t: observing time.
- T_r:start time for processing request r.
- $\mathrm{r}_e^{t,dc}$:current executing task in datacenter dc at time t.
- $R_w^{t,dc} = \{\mathrm{r}_w^{t,dc}\}$: set of waiting tasks in datacenter dc at time t.

End to end latency of MEC request

- For a request r, who originate from M_r and arrives at T_r



$$QL_{dc} = C_{r_e^{Tr,dc}}/U_{dc} - \left(A_r - T_{r_e^{Tr,dc}}\right) + \sum_{r_w^{Tr,dc} \in R_w^{Tr,dc}} C_{r_w^{Tr,dc}}/U_{dc}$$

Processing Latency on datacenter dc: $PT_{dc} = C_r/U_{dc}$

End to end latency of MEC request

- Two options for provisioning request r:
- 1) local processing at $\mathrm{M_r}$
- $$QL_{\mathrm{M_r}} + PT_{\mathrm{M_r}} = \frac{\mathrm{C}_{\mathrm{r}_e^{Tr,\mathrm{M_r}}}}{U_{\mathrm{M_r}}} - \left(A_r - T_{\mathrm{r}_e^{Tr,\mathrm{M_r}}}\right) + \sum_{\mathrm{r}_w^{Tr,\mathrm{M_r}} \in R_w^{Tr,\mathrm{M_r}}} \mathrm{C}_{\mathrm{r}_w^{Tr,\mathrm{M_r}}} / U_{\mathrm{M_r}} + \frac{\mathrm{C_r}}{U_{\mathrm{M_r}}}$$
- 2) offloading to dc_o
- $$QL_{\mathrm{dc_o}} + PT_{\mathrm{dc_o}} + L_{p_{\mathrm{M_r},dc_o}} = \frac{\mathrm{C}_{\mathrm{r}_e^{Tr,\mathrm{dc_o}}}}{U_{\mathrm{dc_o}}} - \left(A_r - T_{\mathrm{r}_e^{Tr,\mathrm{dc_o}}}\right) + \sum_{\mathrm{r}_w^{Tr,\mathrm{dc_o}} \in R_w^{Tr,\mathrm{dc_o}}} \mathrm{C}_{\mathrm{r}_w^{Tr,\mathrm{dc_o}}} / U_{\mathrm{dc_o}} + \frac{\mathrm{C_r}}{U_{\mathrm{dc_o}}} + L_{p_{\mathrm{M_r},dc_o}}$$

Heuristic-Basic idea

- Lowest Latency First (LLF)
- For each request r,
- Calculate response time of each available dc,
- Choose optimal destination with path to handle request r.

- Drawbacks:
- 1) introduce extra traffic between datacenters.
- 2) may cause unnecessary offloading chain.

Heuristic-DTO

- To avoid unnecessary offloading chain
- Set a constant as threshold of latency difference between two options.
- Difference Triggered Offloading (DTO)
- For each request r,
- Find candidate datacenter with lowest latency;
- Compare latency difference between local MEC with remote candidate dc.
- If difference exceeds threshold, then offload it,
- Else, keep it at the local MEC.

Heuristic-LW_LLF

- To reduce extra inter-DC traffic
- Take offloading path length into consideration
- Length Weighted Lowest Latency First
- For each request r,
- Calculate actual response time of each available dc,
- Calculate length weight and multiplex it with the actual response time as length weighted latency of each candidate solution,
- Choose the optimal destination with path to handle request r.

ILP Formulation

Given

$G(V, E)$, topology of overall networks; V: set of nodes; E: set of links.

$V_{DC} \subset V$, set of datacenter node locations.

$BC_{i,j}$: bandwidth capacity of link $(i, j) \in E$. in unites of bps

CC_{dc}: computing capacity of datacenter $dc \in V_{DC}$. In unites of Hz

$K_{s,d}$: K shortest path from source server $s \in V_{dc}$ to destination server $d \in V_{dc}$.

$Q_{s,d}^{i,j}$: set of paths from source s to destination d passing through link $(i, j) \in E$.

$PL_k^{s,d}$: propagation latency of path p between source s to destination d. in units of second.

R: set of requests, each of which has an index from 1 to N.

M_{r_k}: origination datacenter of kth request $r_k \in R$, $M_{r_k} \in V_{DC}$.

C_{r_k}: required computing resource of request $r_k \in R$.

B_{r_k}: required bandwidth of kth request $r_k \in R$.

T_{r_k}: arrive time of request $r_k \in R$.

H_{r_k}, hold time for transmit r_k.

ILP Formulation

Variables

1. $x_{dc}^{r_k}$, binary, equals one if $r_k \in R$ is served by datacenter $dc \in V_{DC}$.
2. $y_{p,dc}^{r_k}$, binary, equals one if $r_k \in R$ is routed to destination datacenter $dc \in V_{DC}$ via path $p \in K_{M_{r_k},dc}$.
3. $t_{dc}^{r_k}$, integer, scheduled start time of request $r_k \in R$ on $dc \in V_{DC}$.

Objective

Min the average latency of all the given requests

Execution Time

$$\sum_{r_k \in R} \left(\sum_{dc \in V_{DC}} x_{dc}^{r_k} * t_{dc}^{r_k} \right) - T_{r_k} + \sum_{r_k \in R} \sum_{dc \in V_{dc}} x_{dc}^{r_k} * C_{r_k} / CC_{dc}$$

Waiting Time

$$+ \sum_{r_k \in R} \sum_{dc \in V_{DC}} \sum_{p \in K_{M_{r_k},dc}} y_p^{r_k,dc} * L_p^{M_{r_k},dc}$$

Propagation Time

ILP Formulation

Constraints:

(1) Used bandwidth of each link cannot exceed its capacity:

$$\sum_{\mathrm{dc}\in \mathrm{V}_{DC}} \sum_{\mathrm{p}\in \mathrm{Q}^{i,j}_{M_{\mathrm{r}_{k'},dc}}} \mathrm{x}^{\mathrm{r}_k}_{dc} * \mathrm{y}^{\mathrm{r}_k,dc}_{p} * \mathrm{B}_{\mathrm{r}_k} + \sum_{n=k+1}^{N} \sum_{\mathrm{dc}\in \mathrm{V}_{DC}} \sum_{\mathrm{p}\in \mathrm{Q}^{i,j}_{M_{\mathrm{r}_{k'},dc}}} \mathrm{x}^{\mathrm{r}_n}_{dc} * \mathrm{y}^{\mathrm{r}_n,dc}_{p} * \mathrm{B}_{\mathrm{r}_n} * P$$

$$+ \sum_{m=1}^{k-1} \sum_{\mathrm{dc}\in \mathrm{V}_{DC}} \sum_{\mathrm{p}\in \mathrm{Q}^{i,j}_{M_{\mathrm{r}_{m'},dc}}} \mathrm{x}^{\mathrm{r}_m}_{dc} * \mathrm{y}^{\mathrm{r}_m,dc}_{p} * \mathrm{B}_{\mathrm{r}_m} * Q \leq \mathrm{BC}_{i,j}, \forall(\mathrm{i},\mathrm{j}) \in \mathrm{E}, \forall \mathrm{r}_k \in \mathrm{R}$$

$$\mathrm{P} = \begin{cases} 1, if & \mathrm{AT}_{\mathrm{r}_m} + \mathrm{HT}_{\mathrm{r}_m} \geq \mathrm{AT}_{\mathrm{r}_k} \\ 0, if & \mathrm{AT}_{\mathrm{r}_m} + \mathrm{HT}_{\mathrm{r}_m} < \mathrm{AT}_{\mathrm{r}_k} \end{cases}, \mathrm{Q} = \begin{cases} 1, if & \mathrm{AT}_{\mathrm{r}_n} < \mathrm{AT}_{\mathrm{r}_k} + \mathrm{HT}_{\mathrm{r}_k} \\ 0, if & \mathrm{AT}_{\mathrm{r}_n} \geq \mathrm{AT}_{\mathrm{r}_k} + \mathrm{HT}_{\mathrm{r}_k} \end{cases}$$

(2) There should be one datacenter to handle each request.

$$\sum_{dc\in \mathrm{V}_{DC}} \mathrm{x}^{\mathrm{r}_k}_{dc} = 1, \forall \mathrm{r}_k \in R$$

ILP Formulation

(3) There would be one path if one request is offloaded to other datacenters.

$$\sum_{dc \in V_{DC}} \sum_{p \in K_{M r_k, dc}} y_p^{r_k, dc} \leq 1 \,, \forall r_k \in R$$

(4) Selected path should connect source node to selected destination node.

$$\sum_{p \in K_{M r_k, dc}} y_p^{r_k, dc} = x_{dc}^{r_k}, \forall r_k \in R, \forall dc \in V_{DC}$$

(5) There is no overlap between two tasks in time dimension according to FCFS

$$x_{dc}^{r_k} * \left(t_{dc}^{r_k} + \frac{C_{r_k}}{CC_{dc}} \right) < x_{dc}^{r_m} * t_{dc}^{r_m}, \forall r_k \in R, \forall k < m < N$$

Other considerations

- To reduce overall queueing time
- Biger Task to Faster Datacenter
- ……

Comments?

Thank you!

Wei Wang