

令和 3 年度  
修士学位論文

# エッジコンピューティング環境における AP 実行環境の起動時間を考慮した リソース予約手法

Resource Reservation Method Considering AP  
Execution Environment in Edge Computing  
Environment

1245114 有蘭 里奈

指導教員 横山 和俊

2022 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻  
情報学コース

# 要 旨

## エッジコンピューティング環境における AP 実行環境の起動時間を考慮した リソース予約手法

有蘭 里奈

モバイル端末や IoT 機器の普及に伴い、端末内だけではなくクラウドを使った処理を必要とするアプリケーションが増えている。クラウドはデータセンタにあるサーバで処理を行っているが、遠隔にあるため、処理に遅延が生じる場合がある。そこで、モバイルエッジコンピューティングでは、ネットワーク周縁部分にサーバマシンを配置し、アプリケーション処理を分散することで問題解決を図る。しかし、周縁に配置されているサーバは、データセンタにあるサーバマシンのような豊富なリソースをもっていないため、モバイル端末へのリソース割当ては効率的に行う必要がある。我々はこれまでに、モバイル端末の混雑度を考慮したサーバマシンのリソース割当て手法を提案している。しかし、これまでの研究では、アプリケーションの起動時間が考慮されていない。そこで本研究では、アプリケーションの起動時間を考慮し、アプリケーションが起動する時刻に合わせてサーバマシンのリソースを予約する手法を提案する。

**キーワード**    モバイルエッジコンピューティング, エッジコンピューティング

# Abstract

## Resource Reservation Method Considering AP Execution Environment in Edge Computing Environment

Rina Arizono

With the spread of mobile device and IoT devices, more and more applications require processing not only within the device but also in the cloud. The cloud is processed by servers located in data centers, but due to their remote location, there may be delays in processing. Mobile edge computing solves this problem by deploying server machines at the periphery of the network and distributing the application processing. However, servers located at the periphery of the network do not have abundant resources like server machines in data center, so the allocation of resources to mobile terminals needs to be efficient. We have previously proposed a resource allocation method for server machines that takes into account the congestion level of mobile terminals. However, in our previous work, the startup time of applications was not taken into account. In this paper, we propose a method of reserving the resource of the server machine according to the time when the application starts.

**key words**     Mobile edge computing, edge computing

# 目次

<b>第 1 章</b>	<b>はじめに</b>	<b>1</b>
<b>第 2 章</b>	<b>関連研究</b>	<b>3</b>
2.1	関連研究 . . . . .	3
2.2	過去研究 . . . . .	4
2.2.1	Cloudlet 環境における移動経路計画を用いたリソース割当て手法 . .	4
2.2.2	分散アルゴリズムを用いた移動予測を利用した割当て . . . . .	5
2.2.3	過去研究の課題点 . . . . .	7
<b>第 3 章</b>	<b>システムモデル</b>	<b>8</b>
3.1	ネットワーク構成 . . . . .	8
3.1.1	転送・中継設備 . . . . .	9
3.2	MEC サーバ . . . . .	9
3.3	モバイル端末 . . . . .	10
3.4	通信遅延 . . . . .	10
<b>第 4 章</b>	<b>提案手法</b>	<b>13</b>
4.1	概要 . . . . .	13
4.2	予約アルゴリズム . . . . .	14
4.3	動作例 . . . . .	16
<b>第 5 章</b>	<b>評価</b>	<b>18</b>
5.1	評価内容 . . . . .	18
5.2	利用するデータ . . . . .	18
5.2.1	道路ネットワークデータ及び基地局の配置データ . . . . .	18

## 目次

5.2.2	車両データ . . . . .	21
5.3	評価方法 . . . . .	22
5.4	結果 . . . . .	24
5.4.1	MEC リソース量を変化した時の結果 . . . . .	25
5.4.2	モバイル端末数変化を増やした時の結果 . . . . .	25
<b>第 6 章</b>	<b>おわりに</b>	<b>30</b>
6.1	おわりに . . . . .	30
6.2	今後の課題 . . . . .	30
<b>謝辞</b>		<b>32</b>
<b>参考文献</b>		<b>33</b>

# 目次

2.1	格子状マップ	5
2.2	通信している MEC サーバと周辺 MEC サーバ	6
3.1	ネットワーク構成	8
3.2	モバイル端末の走行	11
4.1	割当て例	16
5.1	岡山駅周辺の道路ネットワーク	19
5.2	岡山駅周辺の基地局の配置図	20
5.3	集約局の配置図	21
5.4	MEC リソース 5, モバイル端末 500 台の時の平均ホップ数	24
5.5	MEC リソース 10, モバイル端末 500 台の時の平均ホップ数	24
5.6	MEC リソース 5, モバイル端末 1250 台の時の平均ホップ数	26
5.7	MEC リソース 5, モバイル端末 2000 台の時の平均ホップ数	26
5.8	MEC リソース 5, モバイル端末 2750 台の時の平均ホップ数	27
5.9	ホップ数 1 となる MEC サーバの選択率 (MEC リソース 5)	29
5.10	ホップ数 1 となる MEC サーバの選択率 (MEC リソース 10)	29

# 表目次

3.1	モバイル端末と MEC サーバがもつ情報 . . . . .	10
3.2	システムで用いる記号の定義 . . . . .	12
5.1	岡山駅周辺の座標 . . . . .	19
5.2	各集約局通信エリア内の基地局数 . . . . .	22
5.3	実験パラメータ . . . . .	23

# 第 1 章

## はじめに

近年，スマートフォンを始めとするモバイル機器や IoT の利用が増加し，それに伴いクラウドを使った処理へのニーズが高まっている．クラウドとは，ネットワークを通して，自分が持つストレージへアクセスし，利用することができるサービスの一つである [1]．例えば，写真や動画データ，音楽ファイル，ゲームのセーブデータなど，クラウド上にあるストレージへ保存することで，インターネットへ接続できる端末であれば，特定の端末以外でもアクセスすることが可能となっている．それに伴い，手元にあるモバイル端末ではなく，クラウドサービスを利用して処理を行うアプリケーション（以下 AP と略す）も増えてきている．このように，人々の生活の中で当たり前に使われ，普及している一方で，クラウドへ要求される処理は膨大であり，大きな負荷がかかっていると考えられる．また，クラウドの多くはデータセンタにあるサーバマシンを用いているが，データセンタは，遠隔地に建てられているものが多い．そのため，リアルタイム性を求められるような処理や，大きなデータの処理を行う場合，サーバへの負荷による処理の遅れや物理な距離の関係で処理が遅延する場合がある．そこで，ネットワークの周縁にサーバマシンを配置し，処理を分散するモバイルエッジコンピューティング（以下 MEC と略す）という技術が注目されている [2][3]．ネットワークエッジに位置することから，モバイル端末は，AP の処理をデータセンタまで通信を行わずにサーバでの処理を要求できる．しかし，ネットワーク周縁に配置されるサーバマシンの多くは，データセンタにあるサーバマシンのような豊富なリソースを持っていないため，モバイル端末が集中し処理要求が増えてしまうと，負荷がかかり，リソースが枯渇してしまう可能性がある．そのため，リソースを要求するモバイル端末の割当てを効率的に行う必要がある．この問題に対し，過去研究では，モバイル端末が集中し混雑している具合を利



用して，モバイル端末からの処理要求に対し MEC サーバのリソースを効率よく割当てて手法を提案している [6][7]．しかし，過去研究の提案手法では，シミュレーションによってその有効性は示しているものの，AP の実行環境が起動するために要する時間を考慮していない．そこで，本研究では，AP の実行環境が起動する時刻と，その時刻のモバイル端末の位置から，MEC サーバのリソースの確保を予約する手法を提案する．

## 第 2 章

# 関連研究

この章では、関連研究の紹介と、過去研究の内容について説明する。

### 2.1 関連研究

エッジコンピューティングを利用する上で、本研究で注目しているコンピュータリソースが豊富ではない問題の他に、アクセス集中によるネットワークトラフィックの増加が懸念されている。文献 [4] では、この問題に対し、エッジサーバへ処理要求を出すのではなく、エッジサーバからモバイル端末自身に実行環境を移行するアプローチを取っている。結果として、移行によりトラフィックの問題は解決の見込みがある一方、モバイル端末自身のリソースを大きく消費する問題がある。

文献 [5] では、エッジコンピューティングによって高度道路交通システムへインターネットを接続できるようになった反面、利用できるリソースに制限があることから、車両間のネットワークを利用することで、リソースの共有を行うような手法を提案している。提案手法の中に、仮想マシン移行のためのリソース予約に関する手法がある。その手法では、サービスのドロップ率を削減することを目的としている。リソース割当てを行うタイミングを決定し、予めリソースの予約を行うことで、割当てできなかった場合の再度割当てを考慮する必要がなくなる。

## 2.2 過去研究

我々の過去研究では、モバイル端末の混雑具合を利用した割当て手法を提案している。その提案手法と課題点について紹介する。ここでは、文献 [6] と文献 [7] についてそれぞれ紹介する。

### 2.2.1 Cloudlet 環境における移動経路計画を用いたリソース割当て手法

まずは文献 [6] について紹介する。モバイル端末のアクセスが集中し、混雑しているエリアから優先的にモバイル端末の割当てを行う手法を提案している。この手法での目的は、MEC サーバのリソースの枯渇を抑えることと、AP の実行環境が起動する回数の抑制である。提案手法の内容は、このシステムモデルで定義されている AP の許容される遅延時間内で、混雑したエリアで処理要求を行なっているモバイル端末へのリソース割当てを、混雑していないエリアにあるサーバで行うものである。AP 起動回数の抑制では、継続割当ての考えを用いている。継続割当てとは、リソース割当て実行時刻の一つ前の時刻で割当てられていたサーバへ継続して割当てを行う方法である。継続割当てが可能であるかの判定として、継続割当て距離という指標を用いている。この継続割当て距離内であれば、サーバを継続して利用する、つまり、サーバの起動回数の頻度を下げることが可能である。また、この手法では、二次元の格子状に作られたマップを元にサーバの配置を行なっている。そのマップに合わせ、モバイル端末も二次元的に移動するモデルを利用していた。図 2.1 にこのモデルを示す。

この研究の課題点として、一極集中型のモデルである集中アルゴリズムの利用を想定している点と、格子状のマップのモデルを利用している点が挙げられる。集中アルゴリズムの利用は、モバイル端末数が増え続けると一極集中管理では、一台ずつに対して処理環境の手配を行うのは困難であるため現実的ではないと言える。また、格子状のマップでは、現実にとし込んだ際、格子状になっている地形があまり存在しないことから、シミュレーションによる疑似的なものであるため、実用性があるとは言えない。

## 2.2 過去研究

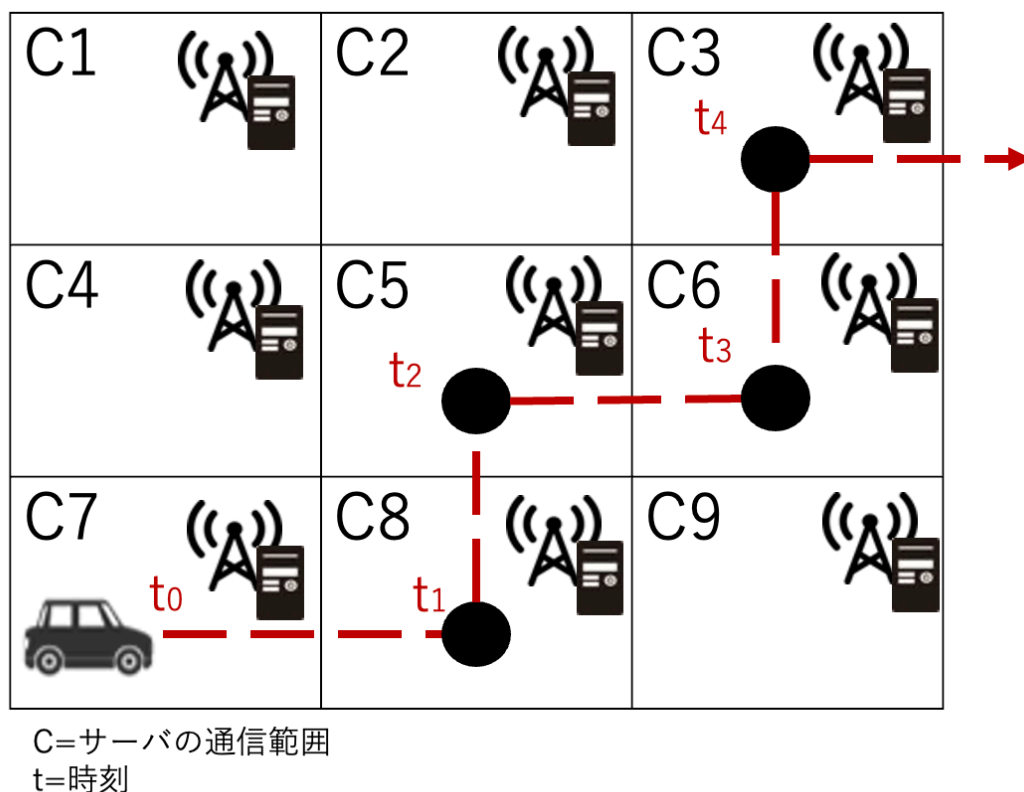


図 2.1 格子状マップ

### 2.2.2 分散アルゴリズムを用いた移動予測を利用した割当て

この手法では、分散アルゴリズムを用いてモバイル端末へのリソース割当てを、MECサーバ自身で自律分散的に処理を行う手法を提案している。分散アルゴリズムを利用することによって、次に割当てを行うサーバを MEC サーバ自身で判断しなければならない。そこで、この研究では、モバイル端末の過去の移動経路から次に移動する位置を予測し割当て先の決定を行う手法を提案している。もし、割当てが決定したサーバのリソースが不足し割当て不可となった場合、その MEC サーバから半径  $n\text{km}$  以内にある周辺の MEC サーバへ割当て依頼を行い、割当て先の再決定を行う。MEC サーバとその周辺の MEC サーバの様子を図 2.2 に示す。図 2.2 では、モバイル端末を車両で表し、通信している MEC サーバを赤い枠の円、その MEC サーバから半径  $n\text{km}$  以内にある周辺の MEC サーバを緑の枠の円で表している。また、半径  $n\text{km}$  外にある MEC サーバは、通信対象外のサーバとして表して

## 2.2 過去研究

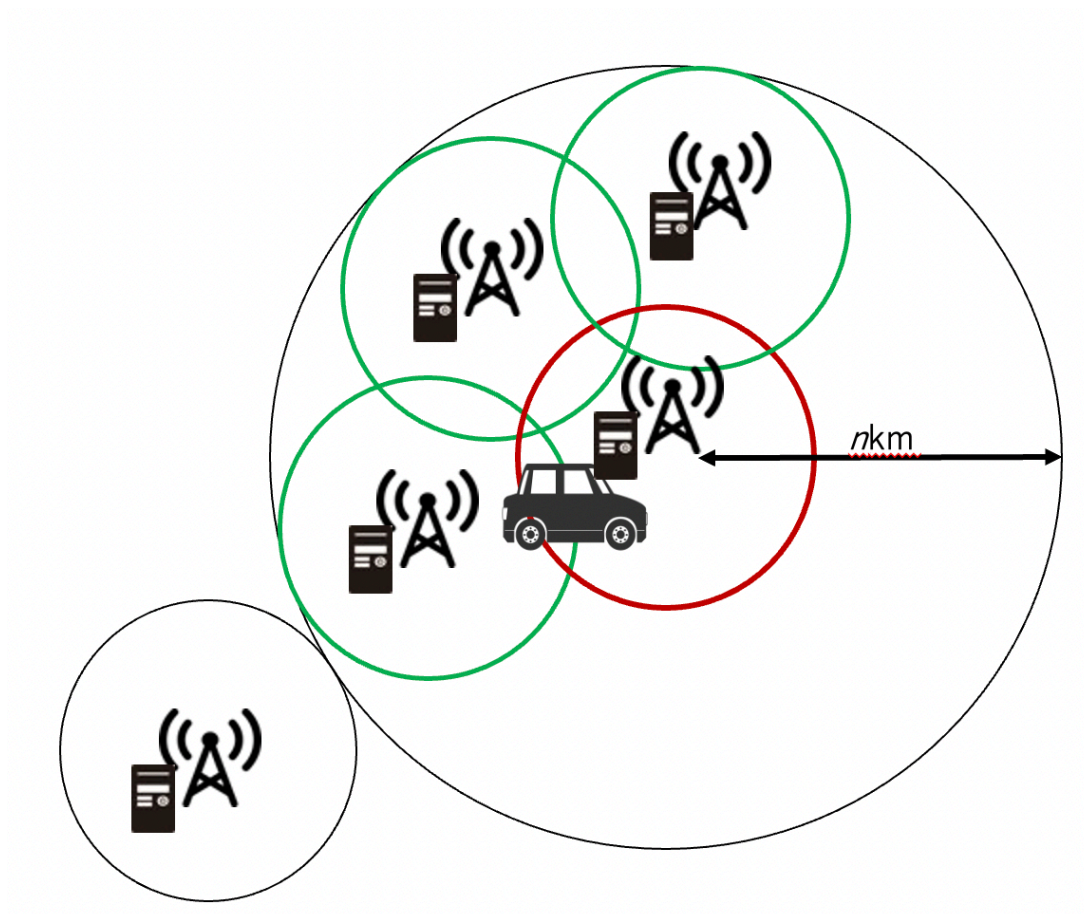


図 2.2 通信している MEC サーバと周辺 MEC サーバ

いる。この手法のシステムモデルは、実際に利用されている MEC の通信モデルを採用している。本研究では、このシステムモデルを採用する。モデルの詳細は次の章で説明する。

この提案手法では、通信遅延の抑制を目的としており、通信遅延の指標としてホップ数を用いている。ホップ数とは、転送・中継設備を経由した回数である。詳細については次の章で説明する。また、この研究において、2.2.1 節の研究の課題点で挙げられたものを解決していると言えるが、サーバの切り替え数については考慮されていない。自律分散的にサーバの割当てを行なっているため、サーバの切り替えを頻繁に行なっていると考えられる。そのため、この研究の課題点として、MEC サーバの切り替え頻度の高さが挙げられる。

## 2.2 過去研究

### 2.2.3 過去研究の課題点

文献 [8] では、この研究に対し、継続割当て手法を適用して評価を行なっている。評価では、継続割当てを適用した場合、平均ホップ数の抑制ができないという結果が得られている。理由として、継続割当てを行うことで、予測に正確性がなくなったことからホップ数の抑制に影響が出たと考えられている。

以上から、過去研究で考慮されていない部分について焦点を当て考える必要がある。過去研究では以下について考慮されていない。

- AP が複数種類ある場合
- AP の実行環境の起動時間

文献 [9] では、文献 [7] の集中アルゴリズム環境で AP を複数種類扱った場合について検討している。評価方法として、AP の種類によって必要なリソース量を設定し、AP 毎に利用できる AP 実行環境に制限を設けている。AP 毎に優先度の重みをつけ、平均ホップ数が抑制されるか検証した結果、優先度の有無に限らず平均ホップ数に差は出なかった。しかし、この検証によって、同じ集約局の通信範囲エリア内であれば、最大 3 ホップでの割当てが可能であることが明らかになった。

以上の結果を踏まえ、本研究では、過去研究が考慮していなかった AP 実行環境の起動時間に焦点を当て、平均ホップ数とサーバの起動回数を抑制するような手法の提案を行う。

## 第 3 章

# システムモデル

本章では，想定するシステムモデルについて説明する．想定するシステムは，データセンタによって MEC サーバ及びモバイル端末を一極管理しているものを想定している．それに伴い，以下では，ネットワーク構成と MEC サーバ，モバイル端末について説明する．

### 3.1 ネットワーク構成

まず，想定しているネットワークの構成について説明する．ネットワークは，基地局，集約局，中間局，データセンタの 4 層で構成としている．構成図を図 3.1 に示す．それぞれの転送・中継設備について以下で説明する．

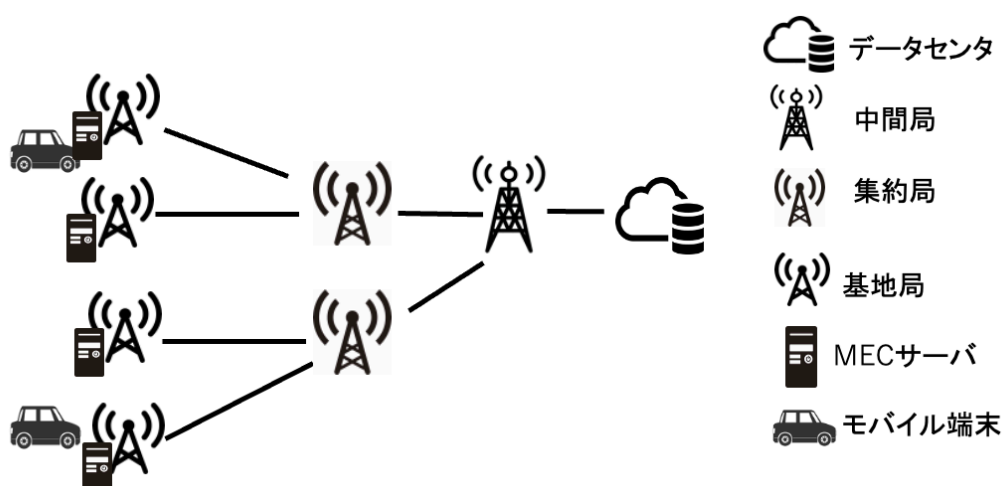


図 3.1 ネットワーク構成

## 3.2 MEC サーバ

### 3.1.1 転送・中継設備

- 基地局

ネットワークのエッジ部分。各基地局の通信範囲は半径  $n\text{km}$  以内であり、基地局同士の通信は、集約局を経由することで可能となる。

- 集約局・中間局

集約局は、ある通信範囲エリア内の基地局の情報を集約しており、中間局を経由してデータセンタと通信している。中間局は、各集約局からの情報をデータセンタへ経路するのに加え、集約局同士の通信の経路も行っている。

- データセンタ

データセンタでは、全ての基地局、MEC サーバ、モバイル端末の情報を管理している。また、MEC サーバのリソース割当てもデータセンタで管理されている。データセンタは今後 DC と略す。

## 3.2 MEC サーバ

MEC サーバは、ネットワークのエッジである基地局に配置されている。MEC サーバ間の通信は基地局のネットワークを利用する。また、MEC サーバが持つ情報を表 3.1 に示す。全ての MEC サーバには、割当て可能なリソース量が設定されている。実際には、記憶容量、CPU コア数、メモリなど複数種類のリソースが存在するが、今回はシンプルなモデルでの評価を行いたいため、MEC サーバで扱うリソースは一種類に限定する。ここで、MEC サーバを  $m_i$  とし、MEC サーバの所有リソース量を  $s$  とした時、MEC サーバの所有リソース量は  $S(m_i)$  で求められる。また、所有リソース量から、モバイル端末を割当てた際に用いたリソース量を引いた値を、割当て可能リソース量  $V(m_i)$  とする。MEC サーバの割当て可能リソース量  $R_i$  が枯渇しない限り、モバイル端末からの処理要求に対応することができる。最後に、MEC サーバ  $m_i$  の位置情報は、緯度経度を  $(x, y)$  とした時に  $L(m_i)$  で表す。

$$L(m_i) = (x, y)$$



### 3.3 モバイル端末

表 3.1 モバイル端末と MEC サーバがもつ情報

モバイル端末が持つ情報	MEC サーバが持つ情報
座標情報	座標情報
移動速度	配置されている集約局エリア
移動経路	基地局の通信可能範囲
実行対象 AP	時間当たりの利用可能リソース量 (AP の実行可能環境数)

### 3.3 モバイル端末

図 3.1 より、想定するシステムモデルでは、車両をモバイル端末として扱う。モバイル端末は道路ネットワーク上を移動する。移動経路は、カーナビゲーションのように事前に示された経路を移動し、示された経路以外は走行しないようなモデルを想定している。表 3.1 に示すモバイル端末が持つ情報は、モバイル端末自身で保有しており、その情報は DC へ共有している。モバイル端末が所有している AP は、モバイル端末自身では処理が困難であるものを利用している。ここで扱う AP はある種類の AP に限定しており、全てのモバイル端末は同じ AP の処理を要求しているものとする。

モバイル端末を  $d_j$  とした時、モバイル端末が MEC サーバへ要求するリソース量  $R(d_j)$  を求めることができる。また、モバイル端末は、図 3.2 のように、物理的に最も近い位置にある MEC サーバへ AP 処理を要求する。最後に、モバイル端末  $d_j$  の位置情報は、緯度経度を  $(x, y)$  とした時に  $Ld(d_i)$  で求めることができる。

$$Ld(d_i) = (x, y)$$

### 3.4 通信遅延

通信遅延の指標として、ホップ数を用いる。ホップ数を指標として扱う理由として、伝送時間やネットワークの効率は物理的距離よりも転送回数からの影響が大きいためである。ホップ数は、一つの転送・中継設備を経由するごとに 1 ホップずつ加算される。ホップ数の

### 3.4 通信遅延

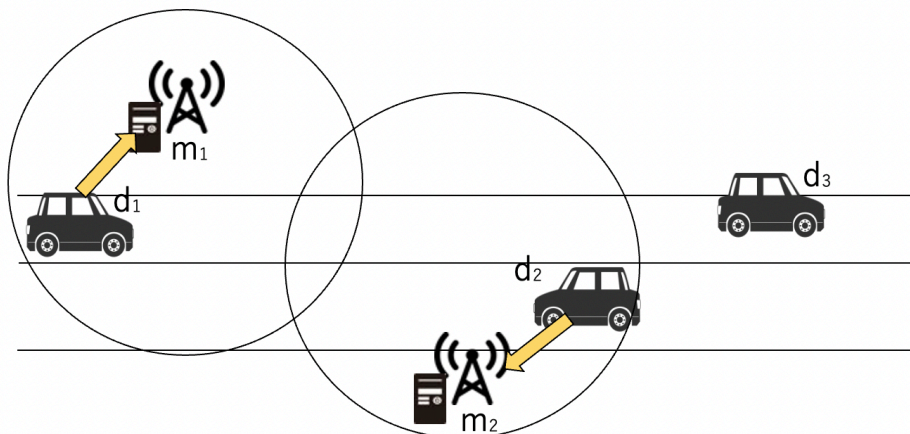


図 3.2 モバイル端末の走行

例を，図 3.1 を用いて説明する．

#### (1) モバイル端末-最も近い位置にある MEC サーバ

モバイル端末と，物理的に最も近い位置にある MEC サーバとの通信では，その MEC サーバの位置にある基地局のみを経由しているため，ホップ数は 1 となる．図 3.1 では，モバイル端末が滞在している MEC サーバが，この例に該当する．

#### (2) モバイル端末-同じ集約局エリア内にある MEC サーバ

同じ集約局の通信範囲エリア内にある基地局と通信を行う場合，モバイル端末と近い位置にある基地局から，そのエリアを集約している集約局を経由し，該当する基地局へ通信を行う．つまり，基地局-集約局-基地局という流れで通信を行っており，転送・中継設備を 3 つ経由しているため，この例でのホップ数は 3 となる．図 3.1 では，モバイル端末が滞在していない MEC サーバが，この例での通信対象となる MEC サーバを指す．

#### (3) モバイル端末-DC

DC までは，モバイル端末と近い位置にある基地局から，そのエリアの集約局，中間局を経由して DC へ通信を行う．基地局-集約局-中間局-DC の計 4 つの転送・中継設備を経由しているため，DC までの通信は 4 ホップ必要となる．

### 3.4 通信遅延

#### (4) モバイル端末-異なる集約局エリア内にある MEC サーバ

異なる集約局の通信範囲エリアにある基地局と通信を行う場合、モバイル端末に近い位置にある基地局からそのエリアの集約局、中間局を経由し、該当する基地局がある集約局から基地局へ通信を行う。つまり、基地局-集約局-中間局-通信対象エリアの集約局-通信対象の基地局という流れになり、転送・中継設備を 5 つ経由しているため、この例でのホップ数は 5 となる。図 3.1 では、図中の上段のモバイル端末が滞在している基地局から、下段のモバイル端末が滞在していない基地局へ通信を行うとすると、上段にある基地局から集約局、中間局を経由して、下段の集約局、基地局へと通信が行われる。

以上の例から、経由する中継・転送設備が多いほど通信距離が伸びていく。つまり、ホップ数が多いほど遅延が発生するものとする。

最後に、本章で説明した記号の定義を表 3.2 にまとめる。

表 3.2 システムで用いる記号の定義

記号	意味
$m_i$	MEC サーバ
$S(m_i)$	MEC サーバの所有リソース量
$V(m_i)$	MEC サーバの割当て可能リソース量
$L(m_i)$	MEC サーバの座標
$d_j$	モバイル端末
$R(d_j)$	モバイル端末の要求リソース量
$Ld(d_j)$	モバイル端末の座標

## 第 4 章

# 提案手法

モバイル端末が MEC サーバへ要求するリソースの割当てのアルゴリズムは、単位時間ごとに DC で実行されることによって、リソース割当てが実現している。本研究では、モバイル端末が AP 処理を要求する MEC サーバの選択方法として、AP 実行環境の起動時間を考慮し MEC サーバを選択するリソース割当て手法である「リソース予約手法」を提案する。本章では、リソース予約手法の概要と、そのアルゴリズムについて説明する。

### 4.1 概要

提案手法の目的はホップ数を抑えることである。ホップ数を抑えるためには、ホップ数 1 での割当て数が増えることが望ましい。そこで、モバイル端末の移動経路情報から、通過経路中にある MEC サーバへ AP 処理要求を行うことでホップ数の抑制に繋がると考えた。また、AP 実行環境の起動に時間を要することから、実行環境が起動してから割当てが行えるようにする必要がある。理由として、AP 処理要求を行う MEC サーバが決定しリソース割当てを行ったが、実行環境が起動する前に次の基地局の通信範囲に入ってしまった、異なる集約局エリアへ入ってしまったら、ホップ数が大きくなってしまふからである。そこで、本研究では、リソース割当て要求を受けた時刻から AP の実行環境が起動する時刻の MEC サーバのリソースを予約を行うことで、AP 実行環境の起動時間内に間に合いつつ、ホップ数の抑制をする。

## 4.2 予約アルゴリズム

提案する予約アルゴリズムについて説明していく．AP 処理の要求を受けた時刻を  $t$ ，AP 実行環境の起動時刻を  $\Delta t$  とする．本提案において，リソース割当て要求は，時刻  $t$  に  $\Delta t$  を加えた時刻  $(t + \Delta t)$  で行う．

### ステップ 1. 予約サーバの候補選出

最初のステップでは，モバイル端末が持つ移動経路情報から，リソース割当ての予約を行う MEC サーバ（以降予約サーバと呼ぶ）の選出を以下の手順により行う．

- (1) 時刻  $(t + \Delta t)$  でのモバイル端末の位置を求める．

モバイル端末  $d_j$  の時刻  $(t + \Delta t)$  での位置は，モバイル端末自身が持つ移動経路情報のうち，時刻  $(t + \Delta t)$  の座標情報より求めることができる．

- (2) 候補となる MEC サーバの情報を取得する．

時刻  $(t + \Delta t)$  から集約局エリアが変わるまでの時刻間で，モバイル端末  $d_j$  が通過する MEC サーバ  $m_i$  を求め，予約サーバの候補とする．モバイル端末自身が持つ移動経路情報より，時刻毎に通過する経路を取得することができる．

候補となるサーバ群を  $C$  とすると，

$$C(d_i) = (m_i, m_{i+1}, \dots, m_{i+n})$$

で表すことができる．移動経路による通過する MEC サーバ  $m_i$  の求め方としては，モバイル端末  $d_j$  の時刻毎の座標情報より，MEC サーバ  $m_i$  の配置されている座標情報を用いて，ユークリッド距離を用いて算出する．モバイル端末  $d_j$  の位置が MEC サーバ  $m_i$  の範囲内にあれば，その MEC サーバ  $m_i$  の通信範囲距離内を通過しているため，その MEC サーバ  $m_i$  は候補として取得できる事になる．ユークリッド距離を  $yl$  とすると， $yl$  は次式で求めることができる．

$$YL((x_1, y_1)(x_2, y_2)) = yl$$

モバイル端末  $d_j$  の座標  $Ld(d_j)$ ，MEC サーバの座標  $L(m_i)$  であるため，モバイル

## 4.2 予約アルゴリズム

端末  $d_j$  から MEC サーバ  $m_i$  のユークリッド距離を  $D_n$  は,

$$D_n = YL(L(m_i), Ld(d_j))$$

で求めることが可能である. ここで, 時刻  $t$  から最も近いサーバを優先順位を高く設定する.

### (3) 候補サーバ数の確認

(2) まですべて予約サーバの候補が得られなかった場合, 同じ集約局エリア内にある全ての MEC サーバを予約サーバの候補とする.

この時, 時刻  $t$  時点でモバイル端末  $d_j$  が割当てられているサーバ (以降割当てサーバと呼ぶ) が存在すれば, 優先的に予約サーバの候補として選出する. ない場合は (2) と同様に近いものから選出する. リソースの確保については次で説明する..

### ステップ 2. モバイル端末の割当て優先順序の決定

モバイル端末  $d_j$  の割当て順序は, ステップ 1 で選出した候補となる予約サーバの数によって決定する. 順序は, 候補が少ないモバイル端末順とする.

各モバイル端末の予約サーバの候補  $C(d_j)$  よりそのサイズを比較し, 順番を決定する.

### ステップ 3. 予約サーバの決定

候補となった MEC サーバのリソースが予約可能であるか求める. MEC サーバのリソースは, 時間毎に確保できるか確認する. この間の 1 秒でもリソースに不足が見られた場合は, リソースの完全な確保できないため, その MEC サーバは予約サーバから外される.  $t$  から  $(t + \Delta t)$  の間全てで完全にリソースの確保が可能であれば,  $t$  から  $(t + \Delta t)$  の間のリソースを確保し, AP 実行環境の起動を行う. AP 実行環境の起動は以下の通りである. ここで, 予約サーバが割当てサーバと同じ MEC サーバであった場合, サーバの切り替えは発生しない.

### ステップ 4. 予約サーバから割当てサーバへの切り替え

予約サーバのうち, AP 実行環境の起動が完了していれば, 現在の割当てサーバから予約サーバへ切り替える. ステップ 3 までを現在時刻で行い, ステップ 4 から時刻を一つ繰り上げた  $t + 1$  時で, 予約サーバの起動が完了していれば, 割当てサーバを予約サー

### 4.3 動作例

バへ切り替える。

以上で割当てステップの説明は終了である。

### 4.3 動作例

図 4.1 を用いながら、アルゴリズムの手順に沿って動作例を説明する。図 4.1 の例では、現在時刻を  $t$  とし、 $t$  時点の割当てサーバは M1 であり、このモバイル端末  $d$  は  $t+1$  秒ごとに M2, M5, M11 の順に走行する。また、例で扱う  $\Delta t$  は 2 とし、予約サーバは  $t+1$  で M2 を予約しているとする。

ステップ 1(1) より。モバイル端末の位置を求める。

$\Delta t = 2$  より、例では図 4.1 の  $t+2$  時点の位置を求める。 $t+2$  時点の位置は、M5 の通信範囲内に位置している。

ステップ 1(2) より。候補となる MEC サーバの情報を取得する。

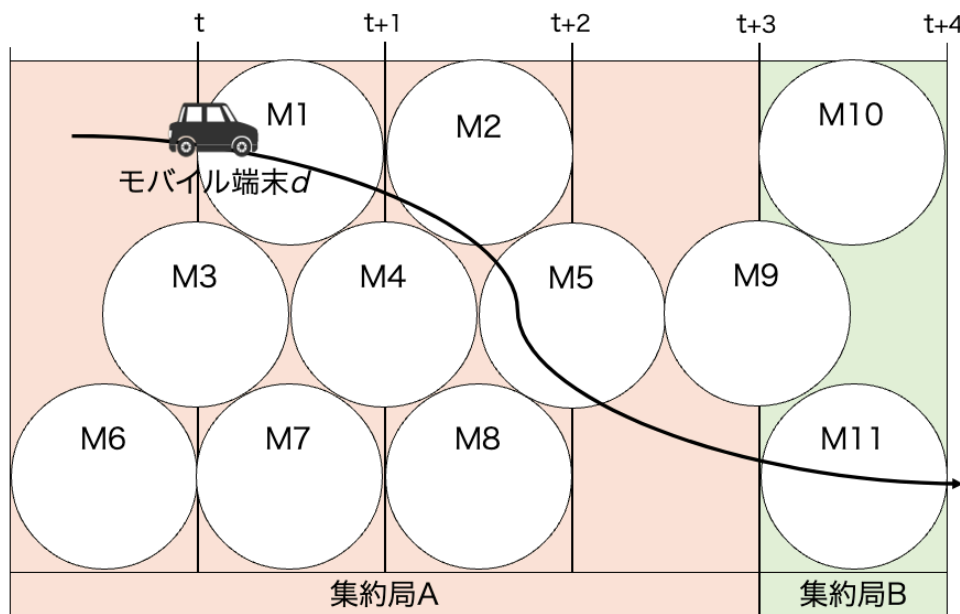


図 4.1 割当て例

### 4.3 動作例

$t+2$  から集約局が変わるまでの間で、滞在している集約局 A 内を通過する MEC サーバを候補として取得する。  $t+2$  では M5,  $t+3$  では集約局 B の通信エリア内にある M11 へ到着している。 よって、ここで取得した予約サーバの候補は M5 のみとなる。

#### ステップ 1(3) より。 予約サーバの候補となる MEC サーバ数の確認

例では (2) の時点で予約サーバの候補を取得しているため、この手順はスキップする。もし、候補が 0 であった場合は、集約局 A 内にある全ての MEC サーバを候補として取得する。つまり、M1 から M8 までの MEC サーバ全てが候補となる。候補の中に、現在時刻  $t$  で割当てサーバとなっている M1 が含まれているため、M1 を候補の中でも最も優先度が高い候補として並び替える

#### ステップ 2 より。 モバイル端末の割当て優先順位を決定する。

例では、モバイル端末  $d$  の予約サーバは M5 のみであるため、候補数は 1 となる。もし、ステップ 1 の (3) のように、集約局内にある全ての MEC サーバを候補として取得していた場合、候補数は 8 となる。

#### ステップ 3 より。 候補となっているサーバ群より、予約サーバの決定

予約サーバの候補となっている M5 の  $t$  から  $t+2$  まででリソースの確保が可能か確認を行う。M5 のリソースは確保できる状態にあるため、モバイル端末  $d$  は、 $t$  から  $t+2$  で M5 への割当て予約を行う。予約確定後、AP の実行環境を起動させ、予約サーバ決定までのフェーズは終了である。

#### ステップ 4 より。 割当てサーバから予約サーバへの切り替え

例では、 $t+1$  で M2 を予約サーバとしている。  $t+1$  で起動が完了しているため、割当てサーバを M1 から M2 へ切り替える。



# 第 5 章

## 評価

### 5.1 評価内容

提案手法の有効性を確かめるため，リソース割当てシミュレーションにより評価を行う．評価として，提案手法と，提案手法を適応せずにモバイル端末が AP 処理を要求する MEC サーバの選択をする方法より，提案手法と同様の方法で評価を行う．以下では，シミュレーションを行う上で必要なデータについての説明と，評価方法の説明を行う．

### 5.2 利用するデータ

シミュレーションを行う上で，モバイル端末及び基地局は，座標データを参照して位置を求めている．それに伴い，評価に使うシミュレーターには，実際に使われている道路ネットワークのデータと，基地局配置のデータを用いる．また，モバイル端末は，システムモデルで車両に置き換えて定義しているため，車両の走行シミュレーションデータを，走行データとして扱う．本節では，地図に関するデータ及び車両データについて説明を行う．

#### 5.2.1 道路ネットワークデータ及び基地局の配置データ

道路ネットワークデータは，先行研究の評価でも利用されていた岡山駅周辺のデータを利用する．シミュレーションで利用するデータは，OpenStreetMap より取得した [10]．取得した通りネットワークデータの図 5.1 に，座標を表 5.1 に示す．

基地局の配置データは，道路ネットワークのデータとして取得した，表 5.1 を元に，世界

## 5.2 利用するデータ

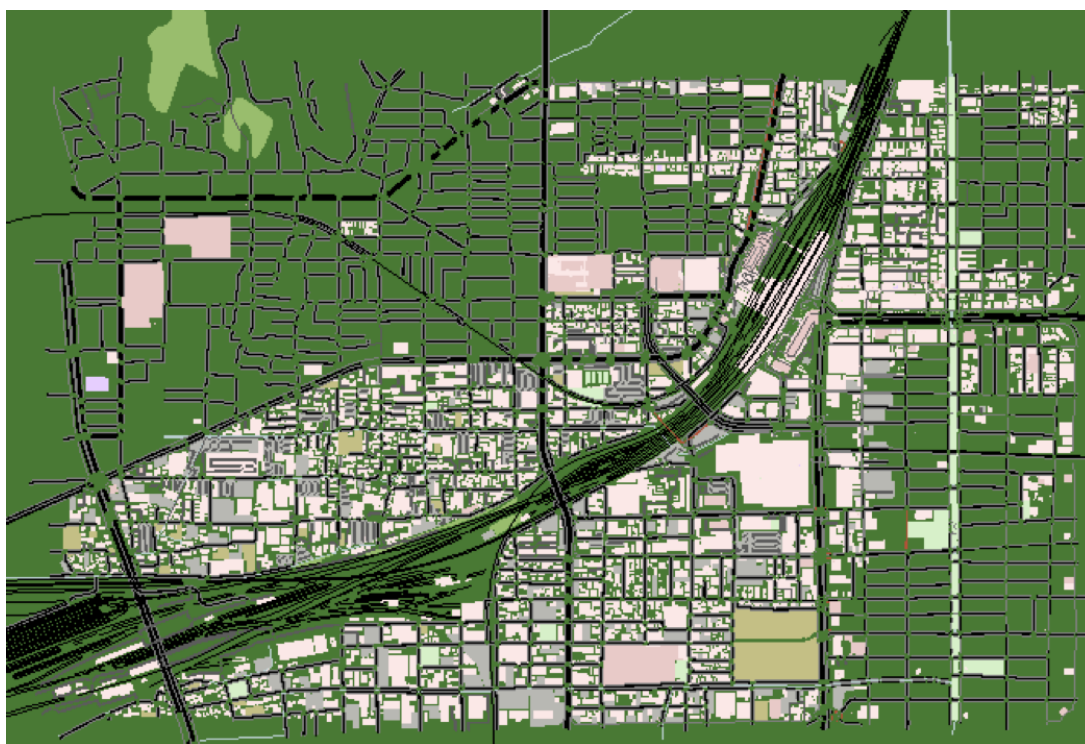


図 5.1 岡山駅周辺の道路ネットワーク

表 5.1 岡山駅周辺の座標

緯度	経度
34.673946	133.879051
34.673946	133.932266
34.629747	133.879051
34.629747	133.932266

中にある携帯電話基地局の位置情報を扱っている OpenCellID より取得する．[11] 取得した基地局のデータは，表 5.1 の座標位置内にあるもので，KDDI の LTE 回線を扱う基地局のみの 548 台の配置データを取得した．取得した道路ネットワークと基地局の配置の様子を図 5.2 に示す．図 5.2 の青のピンは基地局配置を示し，緑のピンは表 5.1 の座標地点を指している．

次に，道路ネットワークデータと基地局配置データをもとに，集約局の配置を行う．評価

## 5.2 利用するデータ

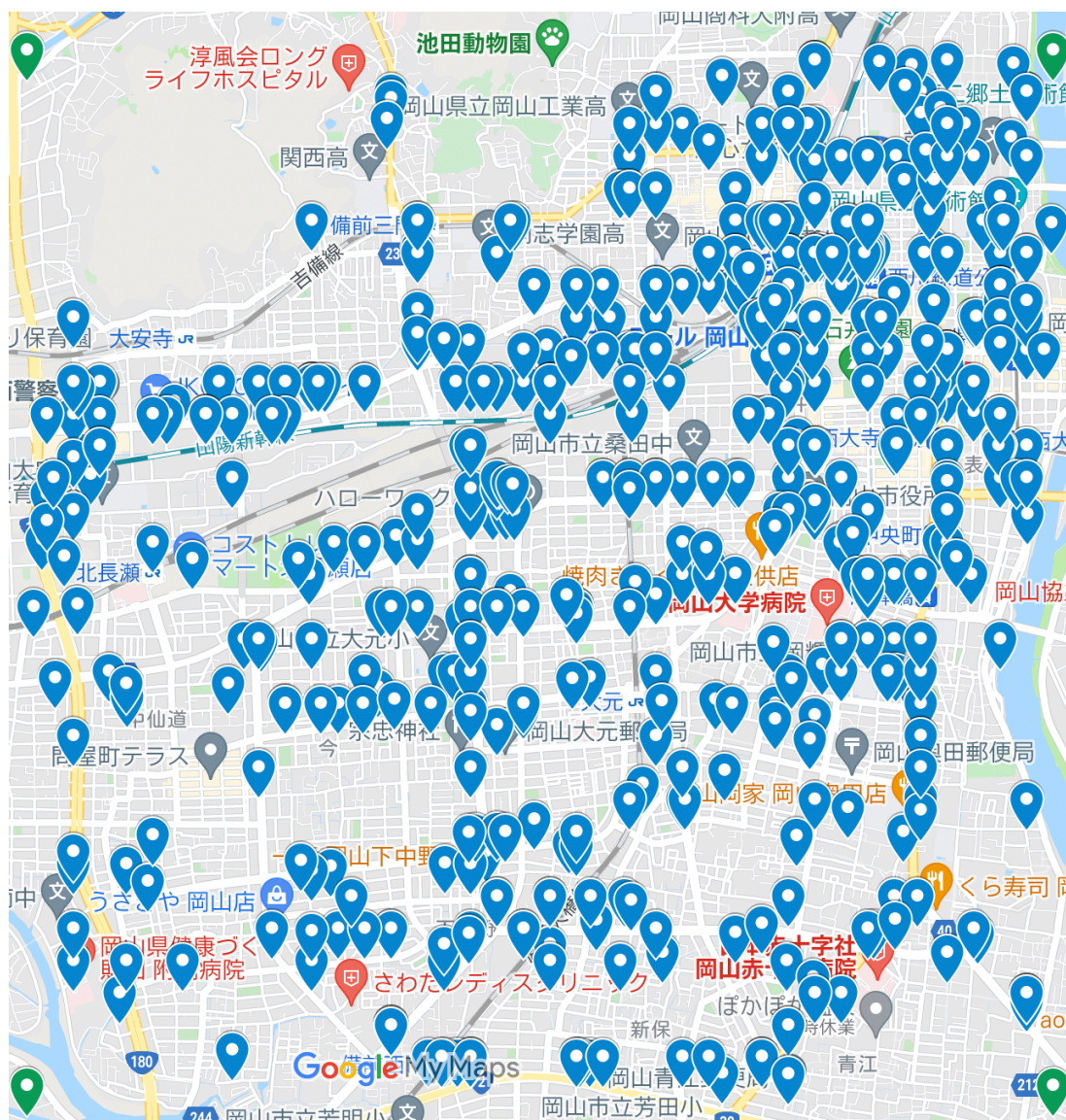


図 5.2 岡山駅周辺の基地局の配置図

では、図 5.2 を  $3 \times 3$  の格子状にわけ、各セルに一つづつに集約局を配置した。集約局の配置図を図 5.3 に示す。

また、各集約局エリア内にある MEC サーバの数を表 5.2 に示す。シミュレーションでは、モバイル端末が今回設定した集約局のどのエリアにも属さない位置へ移動した場合、エリア内に戻ってくるまでの間は割当て処理は行わない。



## 5.2 利用するデータ

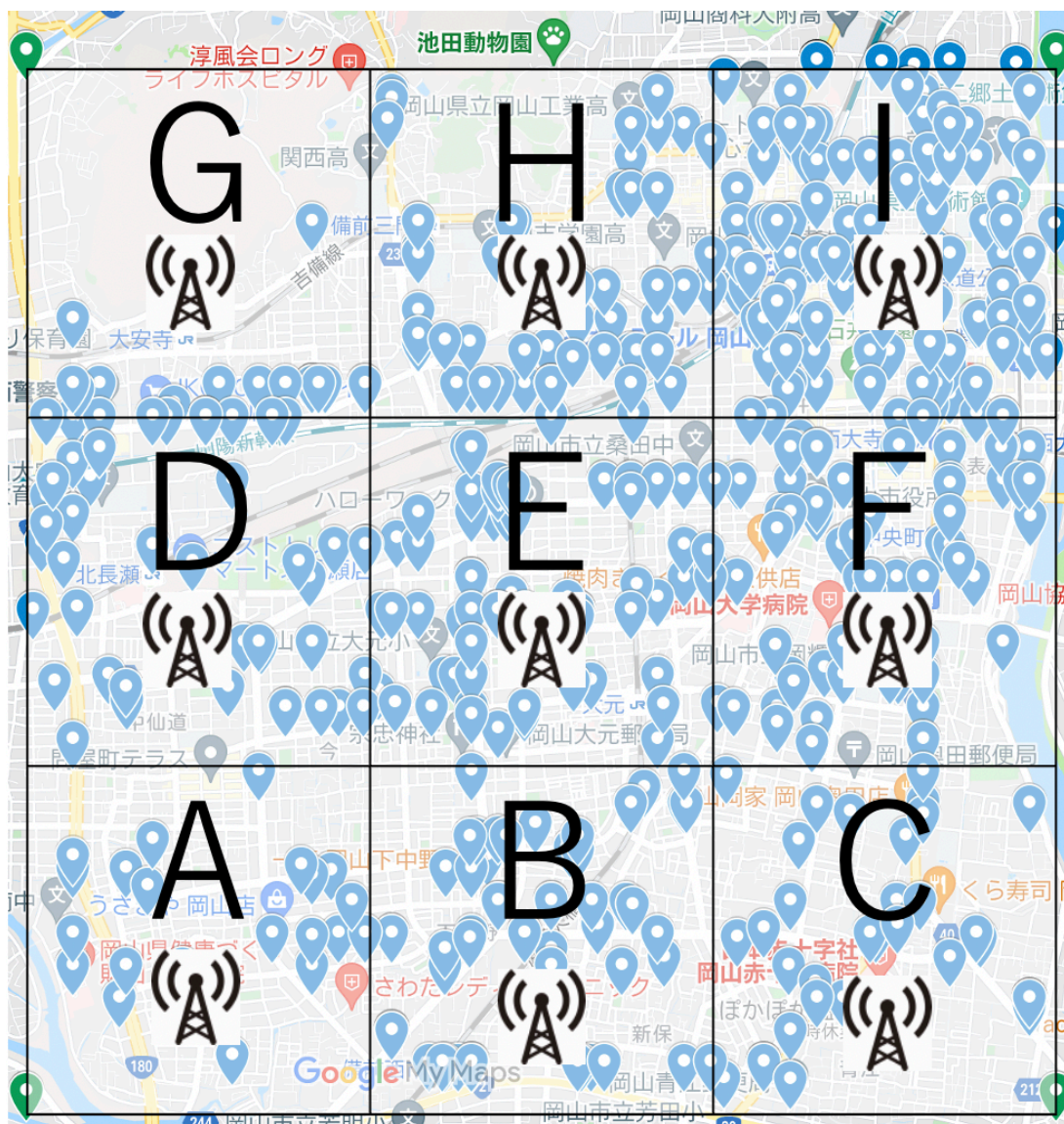


図 5.3 集約局の配置図

### 5.2.2 車両データ

車両データは、道路交通シミュレーションである SUMO を用いて取得する [12]。取得した道路ネットワークのデータを使って、岡山駅周辺の車両走行データを取得する。また、取得した車両走行データを用いて、シミュレーション上で走行車両のデータを複製することで、意図的な渋滞の発生や、走行車両数の増減を行なっている。

### 5.3 評価方法

表 5.2 各集約局通信エリア内の基地局数

集約局エリア名	基地局数
A	22
B	58
C	36
D	50
E	69
F	86
G	16
H	60
I	151

### 5.3 評価方法

評価では、提案手法と 2 つのモバイル端末が AP 処理を要求する MEC サーバの選択方法より割当てシミュレーションを行い、その結果を比較することで行う。比較対象としている二つの MEC サーバ選択方法は以下の通りである。

#### (1) ランダム選出

現在地点の集約局エリアからランダムで予約サーバの候補を選出する方法である。この場合優先順位はつけず、ランダムで選出した順番に従うものとする。

#### (2) リソース順選出

現在地点の集約局エリアにある MEC サーバのうち、時刻  $(t + \Delta t)$  の平均リソース残量が多いものから順に予約サーバの候補を選出する方法である。

二つの比較対象と提案手法をそれぞれリソース割当てのシミュレーションを行い、平均ホップ数及び、ホップ数が 1 となる MEC サーバを選んだ回数を比較することで評価する。

### 5.3 評価方法

表 5.3 実験パラメータ

MEC サーバの割当て可能リソース量	5, 10
MEC サーバの通信可能範囲	500
モバイル端末数	500, 1250, 2000, 2750
モバイル端末の割当て要求リソース量	1
MEC サーバ数	548
AP 実行環境起動時間	30s, 60s, 90s, 120s, 150s

平均ホップ数は、全てのホップした回数に対し、それぞれのホップ数の割当て総回数で割ったものとする。なお、平均ホップ数を計測する上で、DC からモバイル端末、MEC サーバへの割当て情報の際に発生する通信は評価対象外とする。

また、実験パラメータを表 5.3 に示す。7.2 節で取得した基地局配置データを、MEC サーバの配置位置データとして用いる。また、MEC サーバの通信可能範囲は 300m, 500m として設定する。500m という設定は、文献 [13] に基づいて設定している。MEC サーバのリソースの設定は、AP 実行環境の数、すなわち、モバイル端末割当て可能数可能数として、5 台までの時と 10 台までの時で実験を行う。それに伴い、モバイル端末が要求する AP リソースは実行したい数として扱うため、リソース量 1 とする。モバイル端末は、SUMO より取得した車両データ 823 台を元に、500 台、1250 台、2000 台、2750 台それぞれの場合で実験を行う。

上記の条件で、シミュレーションによるリソース割当て実験を行なう。また、シミュレーションは 1000s 間行い、結果と考察を次の節で述べる。

## 5.4 結果

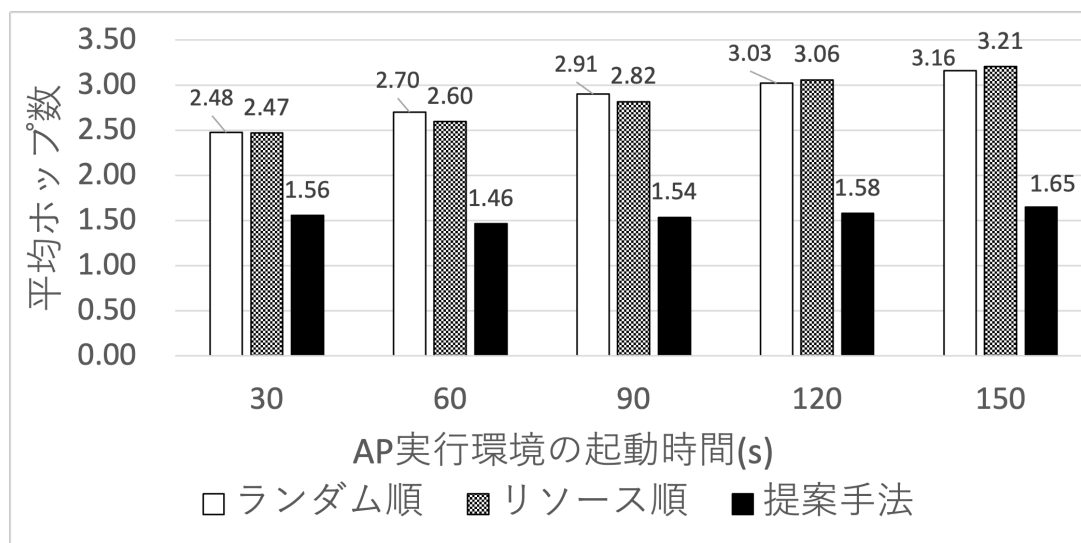


図 5.4 MEC リソース 5, モバイル端末 500 台の時の平均ホップ数

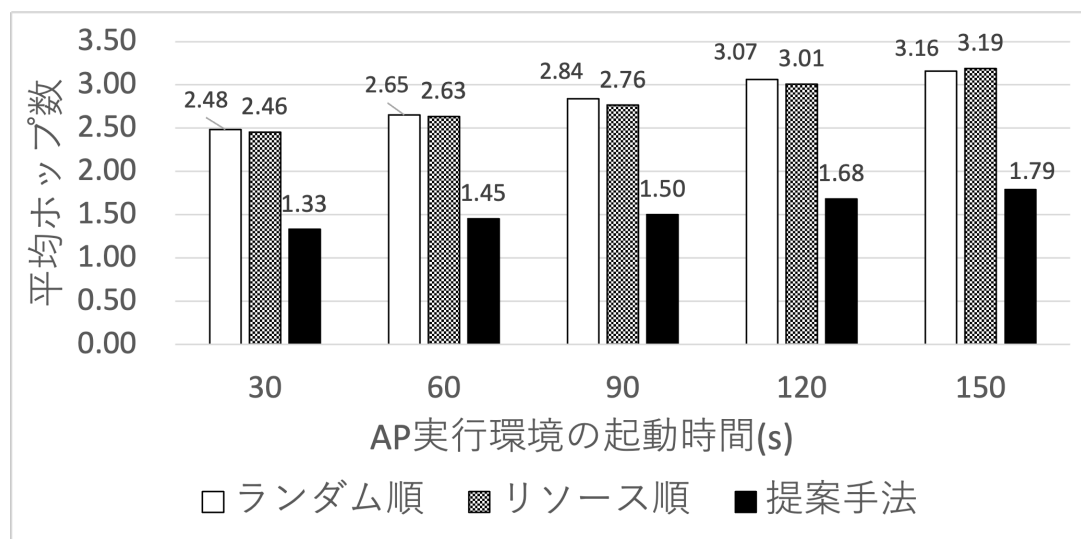


図 5.5 MEC リソース 10, モバイル端末 500 台の時の平均ホップ数

## 5.4 結果

シミュレーション結果と考察について述べる.

## 5.4 結果

### 5.4.1 MEC リソース量を変化した時の結果

まずは、デバイス数を 500 台に固定し、MEC リソース量を変化させた場合の結果から見ていく。MEC リソースが 5 の平均ホップ数を比較したグラフを図 5.4 に、MEC リソース 10 の平均ホップ数を比較したグラフを図 5.5 に示す。

どちらの結果も AP 実行環境の起動時間数が増えていくにつれ、全体なホップ数の平均も上がっている。MEC サーバ選出順別に見ていくと、ランダム順、リソース順はどちらの場合も最低で平均およそ 2.4 ホップとなり、提案手法の平均ホップ数はどちらの場合においても 2 を超えていない。また、この二つの結果は MEC サーバのリソース量がそれぞれ異なっている。リソース量で比較した際、提案手法では、AP 起動時間が 30s での MEC リソース 5 と 10 では 0.23 ホップの差があるのに対し、AP 起動時間が長くなるに連れて、差がなくなり、最終的には MEC リソース 5 の時の方が 0.14 ホップ少ないという結果が得られた。理由として、AP 実行環境の起動時間が大きくなるほど、通過する MEC サーバへの割当てができなかった場合ホップ数 3 以上になる割当てが増える。ホップ数が 3 以上になる割当てになった場合、MEC リソースに余裕がある場所へ割当てられる場合が多くなるため、実行環境の起動時間が大きい場合においては元のリソース量の大小はあまり重要でないと考えられる。

### 5.4.2 モバイル端末数変化を増やした時の結果

次に、MEC リソース量を 5 に固定し、モバイル端末数を変化させた場合の結果を見ていく。図 5.6 に 1250 台の結果、図 5.7 に 2000 台の結果、図 5.8 に 2750 台の結果を示す。

モバイル端末数を増やした結果として、全体的に大きな変化が見られないことがわかる。AP 起動時間が 30s の時の各モバイル端末の平均ホップ数も大きな差は見られない。このことから、モバイル端末の数が増えても、割当てに影響はないと考えられる。

以上二つの検証より、モバイル端末数が増えても割当て自体には影響は出ないが、AP 実行環境の起動時間が大きくなるにつれ全体的に平均ホップ数が大きくなることがわかった。



## 5.4 結果

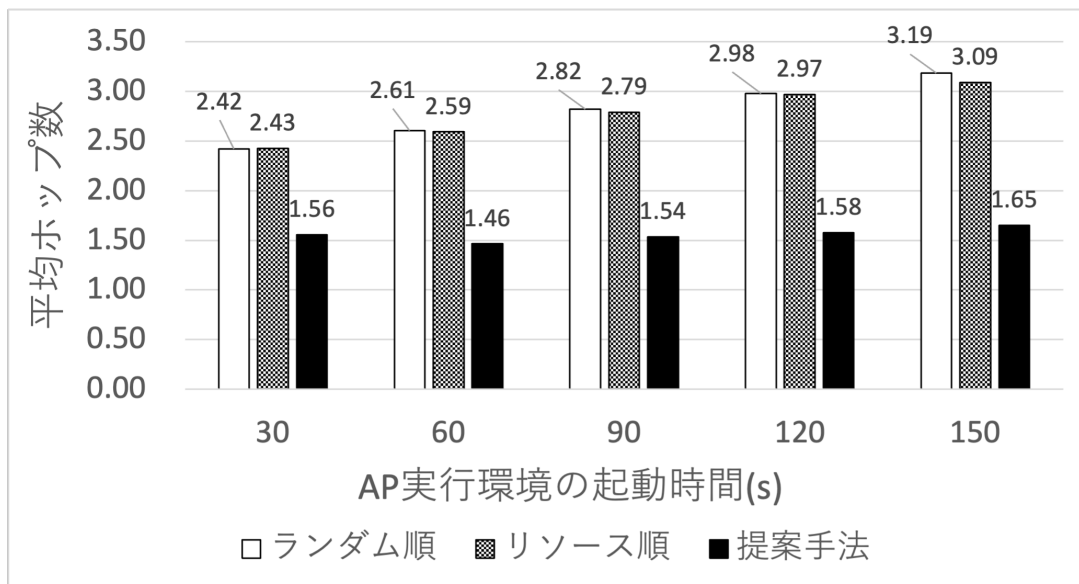


図 5.6 MEC リソース 5, モバイル端末 1250 台の時の平均ホップ数

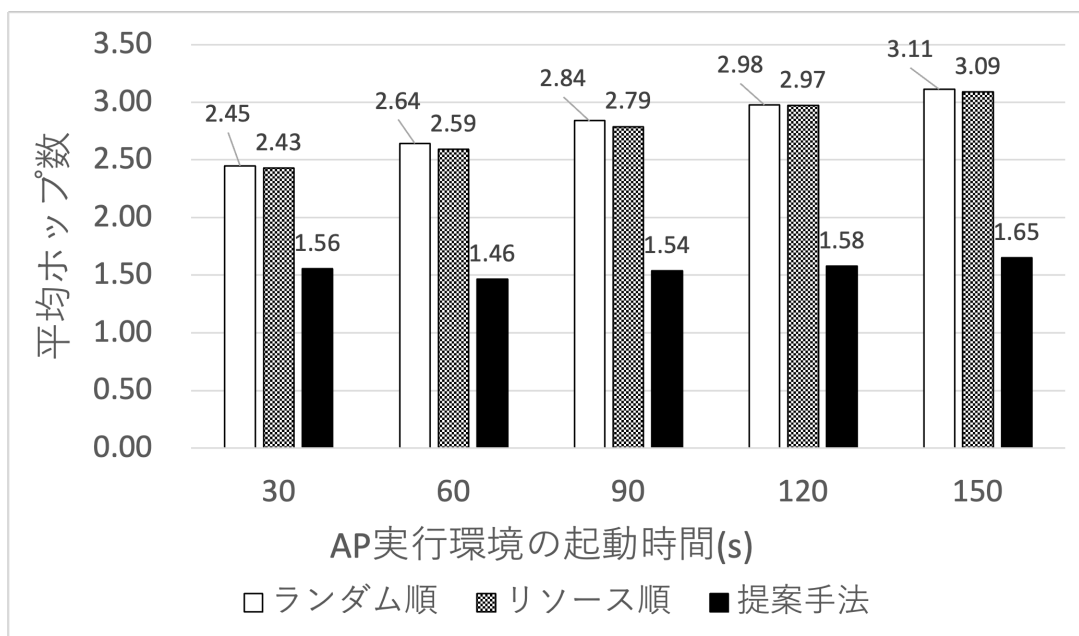


図 5.7 MEC リソース 5, モバイル端末 2000 台の時の平均ホップ数

## 5.4 結果

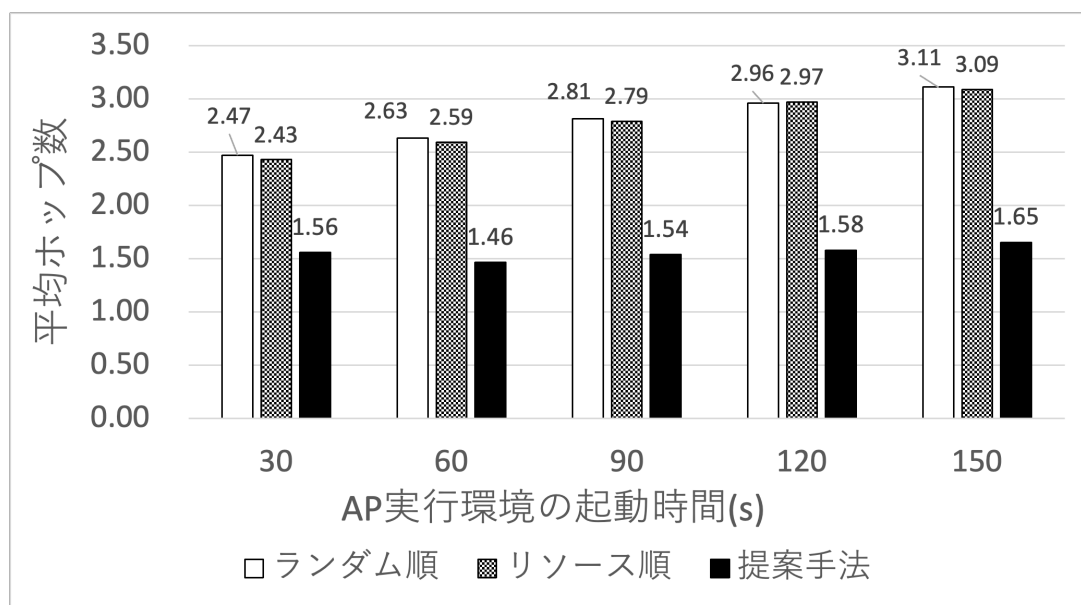


図 5.8 MEC リソース 5, モバイル端末 2750 台の時の平均ホップ数

理由として、AP 実行環境の起動時間を考慮した割当てを行っていることから、起動時間が大きくなると、起動が完了する前に次の集約局エリアへ移動してしまうケースが増えてしまい、これによって、新しい割当て先でなく、元の割当てサーバに、MEC リソースが許す限り継続して割当てを行い、その結果、ホップ数 3 となる割当てが増えてしまうと考えられる。そんな中、提案手法では、どのグラフ内でも平均ホップ数が 2 を超えることなく割当てができています。

それぞれの選出方法別にホップ数 1 の選択回数の平均から選択率を算出した結果を、MEC リソースが 5 の時の結果を図 5.9 に、MEC リソースが 10 の時の結果を図 5.10 に示す。このグラフからも、2 つの MEC サーバ選出順と比べ、提案手法はホップ数 1 となる通過 MEC サーバを選択して割当てを行っていることがわかる。また、MEC リソース数 10 の時は AP 実行環境の起動時間が大きくなるにつれ、ホップ数 1 の割当てとなる MEC サーバを選択している回数が減少している様子がわかる。一方、MEC リソース数 5 の時は、AP 実行環境起動時間毎のグラフの変動に大きな差は無いが、AP 実行環境の起動時間が増えるにつれて徐々に選択回数が減ってきている。この結果の考察として、AP 実行環境の起動時間が 30s の時は、MEC リソース数 5 と比べて MEC リソース数 10 の時の方がホップ数 1 となる選

## 5.4 結果

択回数が多い。これに対し、AP 実行環境の起動時間が 150s の時の選択回数は両者はほぼ変わらない結果になっている。この結果は 5.4.1 節でも述べているが、AP 実行環境の起動数が大きいほど、ホップ数 1 となる割当てを行える回数に限りが生まれるため、APk 実行環境の起動時間が大きくなるにつれて、元のリソース数の大小では割当てに対して影響しなくなると考えられる。

これらの結果より、提案手法による MEC サーバの選択方法を用いることによって、平均ホップ数を抑えた割当てが行えていると言える。また、その結果には、AP 実行環境の起動時間が大きく影響しており、起動時間が大きくなるとその分選択回数は減少するものの、ランダムやリソース残量が多いものから選ぶよりも、提案手法を用いた方が効率のいい割当てができると言える。

## 5.4 結果

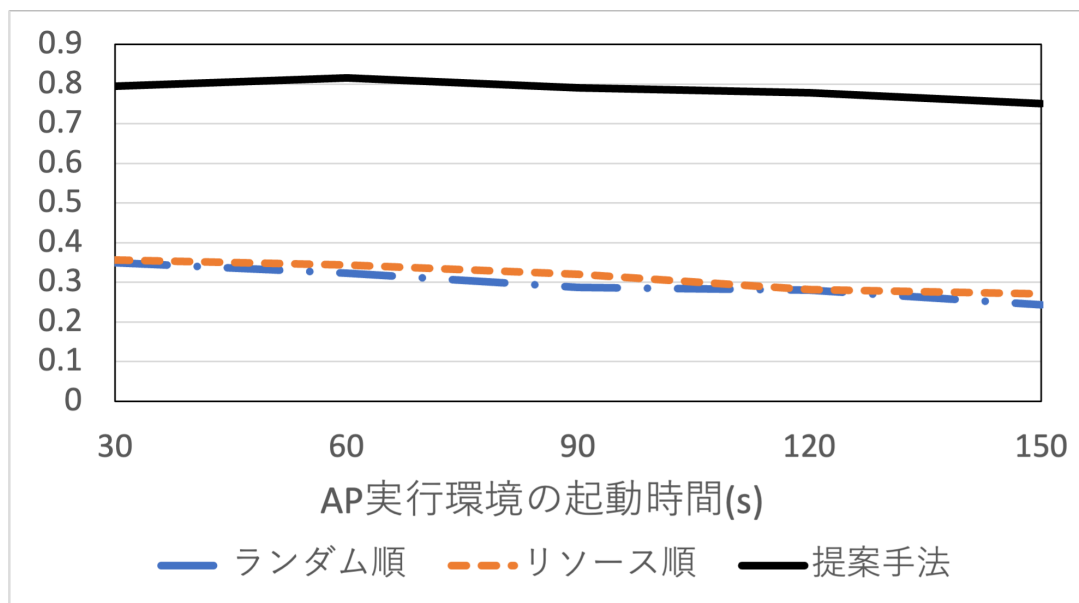


図 5.9 ホップ数1となる MEC サーバの選択率 (MEC リソース 5)

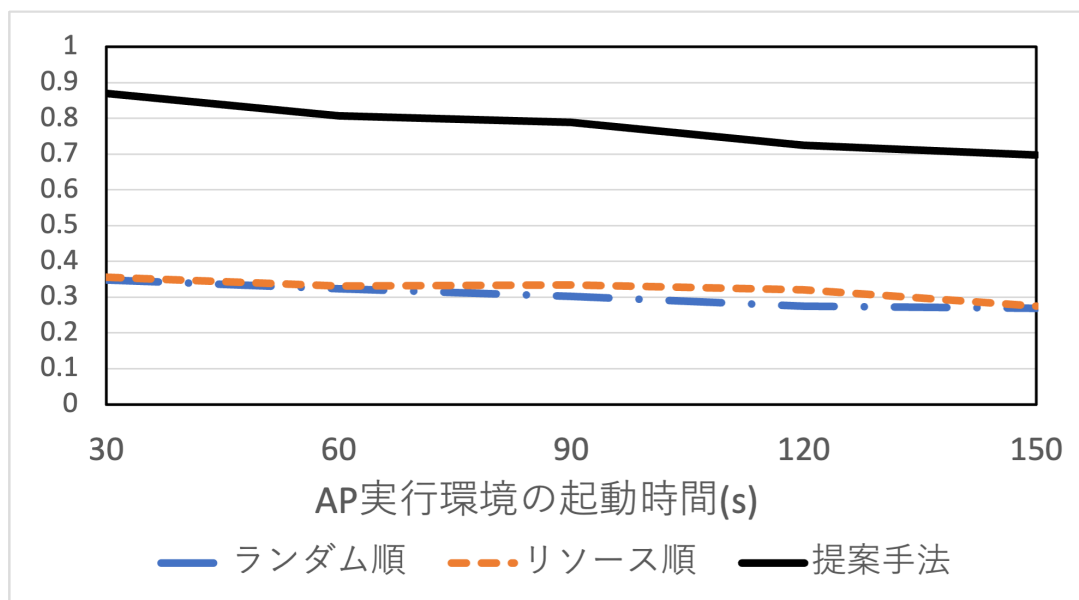


図 5.10 ホップ数1となる MEC サーバの選択率 (MEC リソース 10)

## 第 6 章

# おわりに

### 6.1 おわりに

近年，モバイル端末の利用は増加しており，それに伴い端末内ではなくクラウドへの処理要求も増加している．アクセス集中による負荷や，データセンタの位置の関係でサービス処理中の遅延などといった問題に対し，モバイルエッジコンピューティングでは，ネットワークのエッジに位置していることからユーザに近い場所で処理を行うことができる．しかし，MEC にはデータセンタのサーバマシンのような豊富なリソースがないことから，モバイル端末の効率的な割当てをする必要がある．

本研究では，過去研究で考慮されていなかった AP 実行環境の起動時間を考慮した割当て手法を提案した．この手法では，実行時間から AP 実行環境の起動時間までの間のリソースが確実に確保できるか検証し，その間のリソースを予約確保することで，ホップ数が 1 となるような MEC サーバへの割当てを行っている．評価では，提案手法と，リソース予約を行う MEC サーバの選択方法を，ランダムに決定するランダム順選出と，リソースの平均残量が多いものから選出を行うリソース順選出の 3 つの手法から，平均ホップ数，ホップ数 1 の選択回数の比較によって提案手法が有効であることを示した．

### 6.2 今後の課題

本研究で扱った，一極集中型のモデルは，過去研究の課題として指摘されていたように，現実問題に置き換えた場合，実用的であるとは言えない．そのため，実際の環境に近づける

## 6.2 今後の課題

ために、自律分散的に割当てを行うような分散型アルゴリズムの適応をすることがまず課題点としてあげられる。そして、過去研究含め、本研究では、単位時間毎に同期的に割当て実行を行っていた。これも現実問題に置き換えた場合、モバイル端末は非同期的に処理を要求しているため、非同期的な要求を考慮した環境に対応した改良が必要になってくると考えている。

# 謝辞

本論文の作成にあたり，多くの方々にご指導ご鞭撻を賜りました．指導教員の横山和俊教授には終始適切なご指導を賜りました．ここに深謝の意を表します．またお忙しい中，本研究の副査をお引き受け戴きました高田喜朗准教授，植田和憲講師には，副査として多くのご助言を戴きました．感謝申し上げます．最後に，谷玲治さんを始めとする研究室の皆様に支えられ研究をすることができましたことを心より感謝申し上げます．

# 参考文献

- [1] 総務省 安心してインターネットを使うために国民のための情報セキュリティサイト, ”クラウドサービスとは?” [https://www.soumu.go.jp/main\\_sosiki/joho\\_tsusin/security/basic/service/13.html](https://www.soumu.go.jp/main_sosiki/joho_tsusin/security/basic/service/13.html), 閲覧日 2022 年 1 月 31 日.
- [2] Red Hat, ”エッジコンピューティングについて理解する” <https://www.redhat.com/ja/topics/edge-computing>, 閲覧日 2022 年 1 月 31 日.
- [3] 株式会社モバイルテクノ, ”技術動向:5G 要件を実現するエッジコンピューティング技術” <https://www.fujitsu.com/jp/group/mtc/technology/trend/g5-edge/>, 閲覧日 2022 年 1 月 31 日.
- [4] Xiang Sun, Nirwan Ansari, ”PRIMAL: PRofit Maximization Avatar pLacement for mobile edge computing”, 2016 IEEE International Conference on Communications (ICC), 2016.
- [5] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, Kun Yang, ”Toward Cloud-Based Vehicular Networks with Efficient Resource Management”, IEEE Network( Volume: 27, Issue: 5, September-October 2013), pp48-55(2013)
- [6] 大崎康平, 福永昂輝, 横山和俊, ”Cloudlet 環境における移動経路計画を用いたリソース割当て手法の検討”, 情報処理学会研究報告, Vol.2017-DPS-173, No.8, pp.1-8(2018).
- [7] 福永昂輝, 横山和俊, ”モバイルエッジコンピューティング環境におけるモバイル端末の走行履歴を用いたリソース割当て手法の検討”, 第 82 回全国大会講演論文集, 第 3 分冊, pp147-148(2020).
- [8] 谷玲治, 福永昂輝, 杉村侑起, 横山和俊, ”エッジコンピューティング環境における局所的情報を用いたモバイル端末へのリソース割当て手法”, 情報処理学会研究報告, 第 28 回マルチメディア通信と分散処理ワークショップ論文集, pp.219-221(2020).
- [9] 白神智生, ”エッジコンピューティング環境における AP の動作環境を考慮したリソー



## 参考文献

- ス割当て手法の評価”, 高知工科大学学士論文集 (2020).
- [10] OpenStreetMap, <https://www.openstreetmap.org/about>, 閲覧日 2022 年 1 月 31 日.
- [11] OpenCellID. ' What is OpenCellID ', <https://www.opencellid.org/#zoom=16&lat=37.77889&lon=-122.41942>, 閲覧日 2022 年 1 月 31 日.
- [12] SUMO. ' Simulation of Urban Mobility ', <https://www.eclipse.org/sumo/>, 閲覧日 2022 年 1 月 31 日.
- [13] Shiqiang Wang, Rahul Urgaonkar, Ting He, Murtaza Zafer, Kevin Chan, Kin K.Leung, “ Mobility-Induced Service Migration in Mobile Micro-clouds”, 2014 IEEE Military Communications Conference, pp.835—840(2014).