

令和 4 年度
修士学位論文

RSU を用いたエッジコンピューティング 環境におけるリソース予約状況を考慮した VM 再配置手法

A VM Relocation Method
Considering Resource Reservation Status
in Edge Computing Environment Using RSUs

1255113 永元 陽一

指導教員 横山 和俊

2023 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

RSU を用いたエッジコンピューティング 環境におけるリソース予約状況を考慮した VM 再配置手法

永元 陽一

近年，自動運転などの先進的な運転支援システムの需要が高まっている．それに伴って車両で処理するアプリケーションが高負荷化し，処理の遅延が問題となっている．しかし，車両の計算資源には計算能力や消費電力に限りがある．このため，エンドユーザの近くに計算資源を配置することでアプリケーションの遅延軽減や負荷軽減ができるエッジコンピューティングが注目されている．道路に沿って設置した計算資源と車両が通信することで負荷分散を目指す取り組みとして，RSU(Roadside Unit) を用いたエッジコンピューティングが研究されている．この環境では，車両の移動を考慮してアプリケーションの実行先を再配置する必要があり，RSU と車両の通信切断時刻に再配置を開始する手法が提案されている．しかし，既存手法には，次の RSU の通信可能範囲への移動時間が再配置時間よりも小さい場合に再配置が間に合わないという問題がある．本研究では，前倒してリソース予約することで，通信開始時刻に再配置を完了させる手法を提案し有用性を評価する．

キーワード エッジコンピューティング，ロードサイドユニット

Abstract

A VM Relocation Method Considering Resource Reservation Status in Edge Computing Environment Using RSUs

Yoichi Nagamoto

In recent years, demand for advanced driver assistance systems such as automatic driving has been increasing. As a result, the applications processed in the vehicle become overloaded, and processing delays are becoming a problem. However, the vehicle's computing resources have limited computing power and power consumption. For this reason, edge computing is attracting attention because it can reduce application latency and load by placing computational resources close to end-users. Edge computing using roadside units (RSUs) has been studied as an approach that aims to distribute the load by allowing vehicles to communicate with computational resources installed along the road. In this environment, it is necessary to relocate the application execution destination in consideration of vehicle movement, and a method is proposed to start relocation at the time of disconnection of communication between the RSU and the vehicle. However, the proposed method has the problem that relocation cannot be performed in time if the travel time to the communication range of the next RSU is smaller than the relocation time. In this study, we propose a method to complete relocation at the communication start time by reserving resources ahead of time, and evaluate its usefulness.

key words Edge Computing, RoadSide Unit

目次

第 1 章	はじめに	1
第 2 章	関連技術	3
2.1	エッジコンピューティング	3
2.2	RSU(RoadSide Unit)	4
第 3 章	関連研究	6
3.1	MEC 環境における効率的なリソース割り当てを 目的とした研究	6
3.2	RSU を用いたエッジコンピューティング環境における 再配置手法の研究	7
3.3	関連研究の問題点	8
第 4 章	環境モデル	10
4.1	ネットワーク構成	10
4.1.1	通信設備	10
(1)	RSU	10
(2)	集約局・中間局	10
(3)	データセンタ	11
4.1.2	通信遅延	11
4.1.3	車両の移動および車両と RSU の通信時間	12
4.2	計算資源とアプリケーション	13
4.3	リソース管理・リソース予約	14
第 5 章	提案手法	16

目次

5.1	制約条件と目的関数	16
	(1) 制約条件	16
	(2) 目的関数	17
5.2	再配置の前倒し	17
5.3	再配置先選択指標	17
5.3.1	混雑度を用いた優先順位	18
5.3.2	再配置先選択アルゴリズム	19
5.3.3	アルゴリズムの具体例	20
第 6 章	評価	23
6.1	評価方法	23
6.2	評価指標	25
	(1) 平均処理時間	25
	(2) 平均開始遅延時間	25
	(3) 平均リソース使用率, 平均リソース超過数	25
6.2.1	比較手法	25
	(1) 既存手法	25
	(2) ランダム選択手法	25
6.3	結果と考察	26
6.3.1	平均開始遅延	26
6.3.2	平均処理時間	26
6.3.3	平均リソース使用率	27
6.3.4	要求リソース超過数	28
第 7 章	おわりに	30
7.1	まとめ	30
7.2	今後の課題	30

目次

謝辭 32

参考文献 33

目次

2.1	エッジコンピューティング	4
2.2	RSU を用いたエッジコンピューティング	5
3.1	混雑度による優先度	7
3.2	再配置	8
4.1	環境モデル	11
4.2	通信遅延	12
4.3	経路情報	13
4.4	リソース予約	15
5.1	時刻 t_{s1} から時刻 t_{s2} までの経路と到達時間	18
5.2	混雑度	19
5.3	時刻 t におけるリソース予約状況	22
6.1	シミュレーション環境	24
6.2	平均開始遅延	26
6.3	平均処理時間	27
6.4	平均リソース使用率	28
6.5	平均リソース超過数	29

表目次

4.1	RSU が持つ情報	14
6.1	シミュレーションのパラメータ	24

第 1 章

はじめに

近年、自動ブレーキや自動運転などの先進的な運転支援システムの需要が高まっており、それに伴って車両で処理するアプリケーションが高負荷化している。また、これら運転支援システムはリアルタイム性が求められるため、遅延に対する要求が厳しくなっている [1]。しかし、車両の計算資源には計算能力に限りがあるため、車両以外の計算資源を用いてアプリケーションを処理するエッジコンピューティングが注目されている。エッジコンピューティングは、エンドユーザの近くに計算資源を配置しアプリケーションを処理するため、データセンタの計算資源で処理するクラウドコンピューティングに比べ遅延軽減が期待できる。道路に沿って設置した計算資源と車両が通信することで遅延軽減を目指す取り組みとして、RSU(Roadside Unit) を用いたエッジコンピューティングがある。RSU は、車両の計算資源よりも高い計算能力を持つため、RSU でアプリケーションを処理することで遅延を軽減できる。しかし、RSU はデータセンタにあるサーバマシンほど豊富な計算資源を持たないことや、無線通信のため通信可能範囲に制限があるなどの制約がある。そのため、RSU を用いたエッジコンピューティングでは、車両の移動を考慮してアプリケーションの実行先を再配置する必要がある。この再配置手法として、コンテナ応答時間を削減する再配置手法が提案されている [2]。この手法では、RSU との通信切断時刻に再配置を開始し、通信切断時刻の負荷状況から再配置先を決定する。しかし、次の RSU の通信可能範囲への移動時間が再配置時間よりも小さい場合に再配置が間に合わないという問題がある。また、他の車両の再配置状況によって RSU と通信中の負荷が変化するため、通信中は高負荷となる可能性がある。そこで本研究では、リソース予約状況を考慮した再配置を前倒して開始することで、通信開始時間に再配置を完了させる手法を提案する。

本稿の構成を以下に示す．2 章で関連技術を述べ，3 章で関連研究の内容と問題点を示す．4 章では本研究が想定する環境モデルを説明し，5 章では提案手法を述べる．6 章では提案手法の有用性を確かめるための評価方法と評価項目，結果と評価を示し，7 章でまとめと今後の課題を述べる．

第 2 章

関連技術

2.1 エッジコンピューティング

エッジコンピューティングとは、エンドユーザの近くにサーバを分散配置するネットワークアーキテクチャである。このサーバはエッジサーバと呼ばれ、データ処理をエッジサーバで行うことによりアプリケーションの遅延軽減、ネットワーク負荷軽減、セキュリティ強化、BCP 対策、GDPR と親和性の高いシステム構築に活用できるといった多くの利点がある [3]。近年、スマートフォンや IoT 機器の普及が進んだことで、多くのデバイスがインターネットに接続されることにより、データトラフィックが増大している。このような状況において、エッジコンピューティングの必要性は高まっており、研究が進められている。スマートフォンなどの移動体を対象としたエッジコンピューティングは、モバイルエッジコンピューティング (MEC) と呼ばれ、図 2.1 のようにエンドユーザに近い位置にある計算資源を使うことで、クラウドへの負荷軽減やアプリケーションの遅延軽減を目的として研究が行われている。さらに、この MEC に車両ネットワークと後述する路側インフラ RSU を含めたものは Vehicular Edge Computing (VEC) と呼ばれ、車両ネットワークにおける情報収集範囲の拡大や、自動運転などの高度な運転支援システムへの応用を目的として研究が進められている。

エッジコンピューティングに似たアーキテクチャとして、米シスコが提唱するフォグコンピューティングがある。このアーキテクチャは、エンドユーザとクラウドの間にフォグと呼ばれるサーバ群を設けることで、エッジコンピューティングと同じように分散処理を行うものである。

2.2 RSU(RoadSide Unit)

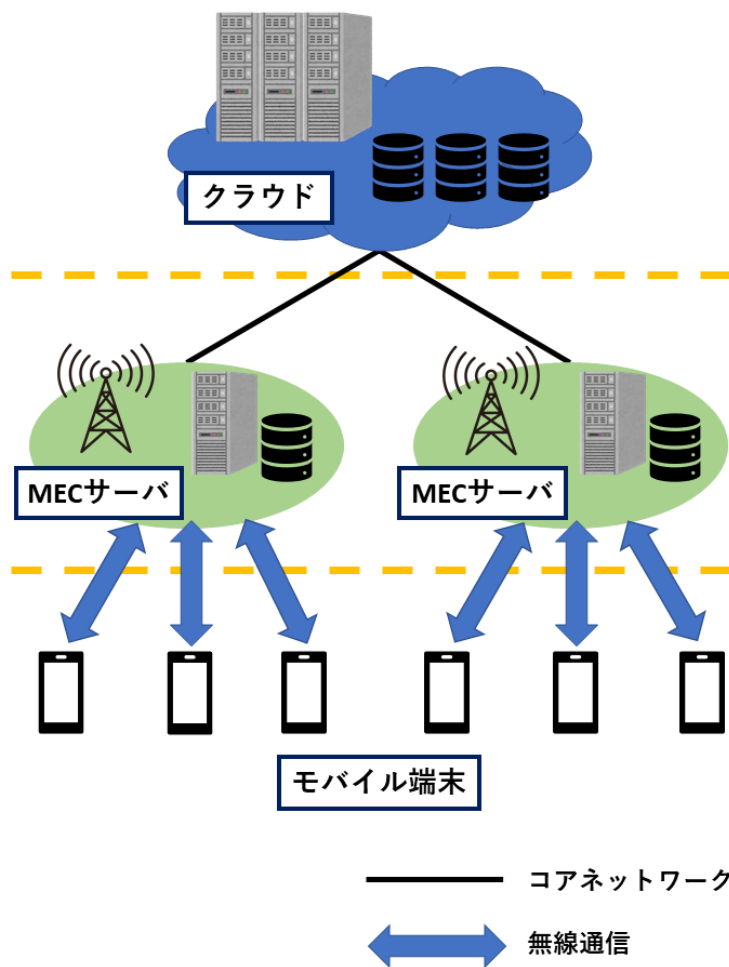


図 2.1 エッジコンピューティング

2.2 RSU(RoadSide Unit)

RSU とは、無線通信可能な計算資源を道路に沿って設置し、車両の認識範囲拡大や車両の計算能力を超えた処理を目指す路側インフラである。RSU は、車両の計算資源よりも強力な計算資源というだけでなく、周囲の車両の情報を収集・伝達することで車両 1 台の認識範囲を超えた制御ができる。図 2.2 のように RSU を利用したエッジコンピューティングは、高度な安全運転支援や自動運転に活用できる計算資源として研究が進められている [1]。

2.2 RSU(RoadSide Unit)

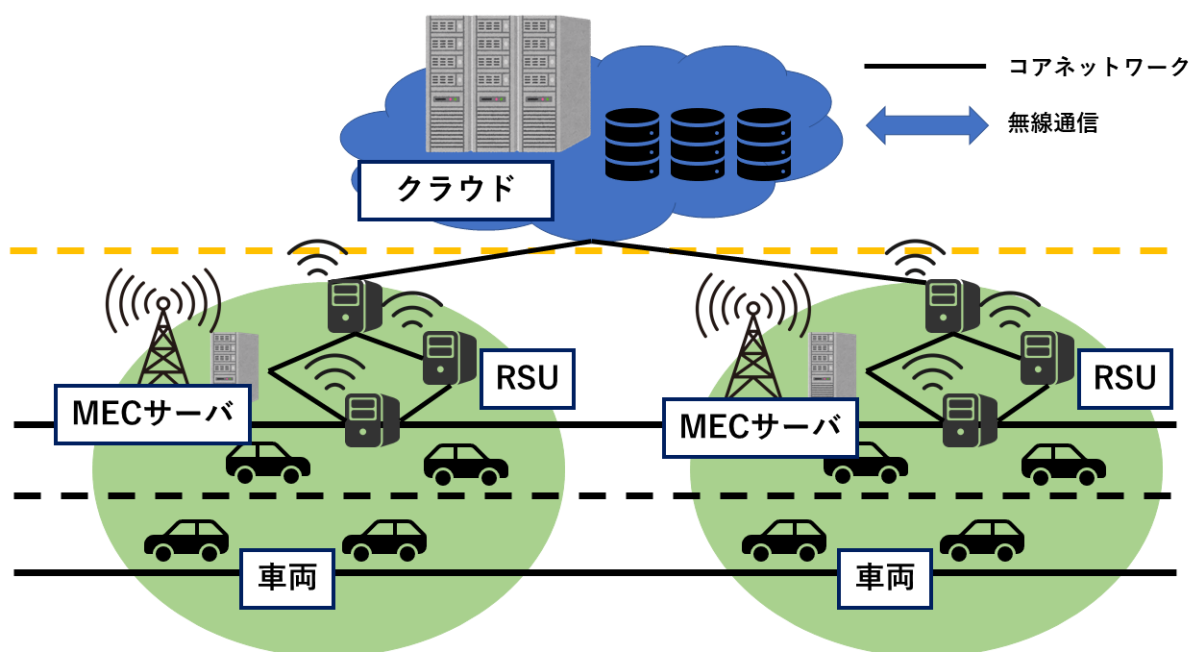


図 2.2 RSU を用いたエッジコンピューティング

第 3 章

関連研究

関連研究として、MEC 環境における効率的なリソース割り当てを目的とした研究と、RSU を用いたエッジコンピューティング環境における再配置手法の研究を説明する。

3.1 MEC 環境における効率的なリソース割り当てを 目的とした研究

スマートフォンや IoT 機器の利用増加に伴ってクラウドで処理を行うアプリケーションが増加している。しかし、クラウドは遠隔地に建てられたものが多く、リアルタイム性が求められる処理や大量のデータによるネットワーク負荷の問題があるため、エンドユーザの近くに計算資源を配置するエッジコンピューティングが注目されている。MEC 環境では、モバイル端末からアプリケーションの処理を要求された際に、モバイル端末の移動や MEC の負荷を考慮して最適な配置先を決める必要がある。この MEC 環境における最適な配置の研究として、[4] と [5] がある。[4] では、通信遅延の抑制と割り当て先の切り替え頻度抑制を目的として、混雑度を用いた車両の優先度と継続割り当てを用いたリソース配置手法を提案している。混雑度は、図 3.1 に示すように要求リソース量の合計が多い順として優先順位を決めており、混雑度の高い MEC と通信する車両から割り当て先を決めることで、通信遅延の評価指標であるホップ数の増加を抑制できることを確認している。[5] では、通信遅延の抑制を目的として、アプリケーション実行環境の起動時間を考慮したリソース配置手法を提案している。この手法では、移動経路情報を用いて位置の先読みし、移動先でユークリッド距離が小さい計算資源に配置することにより通信遅延の抑制している。また、計算資源のへ割

G : 混雑度

R : 要求リソース量

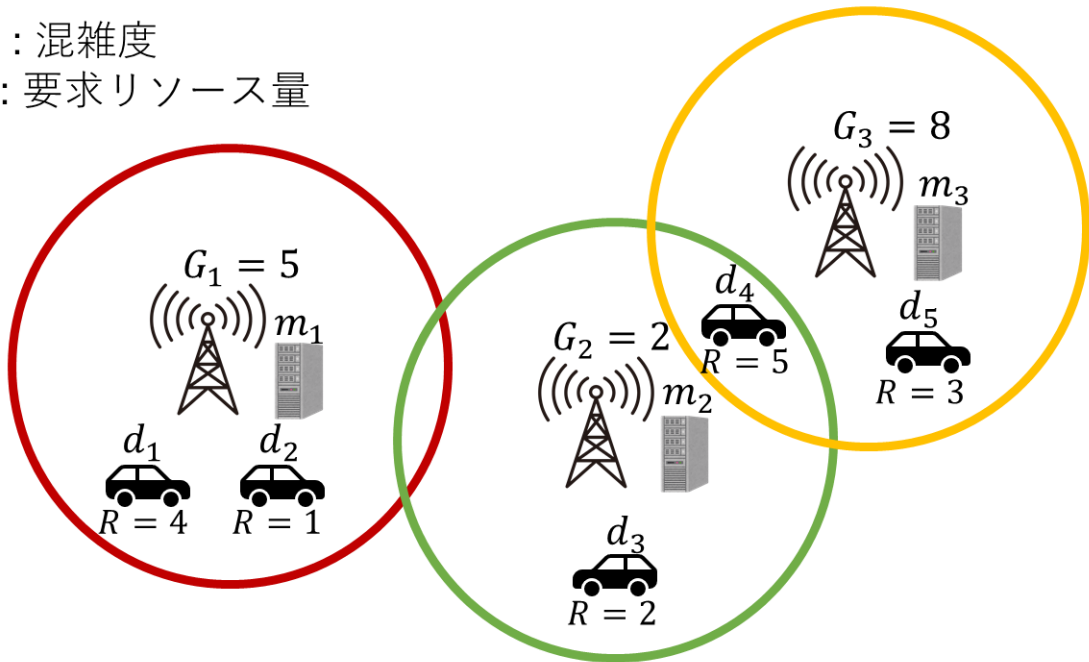


図 3.1 混雑度による優先度

り当てリソース量が最大リソース量を超過することを防ぐため、アプリケーション実行環境の起動時間中のリソースが確保できる計算資源に配置している。シミュレーションを用いて通信遅延の指標であるホップ数を評価しており、ホップ数の増加を抑制できることを確認している。[4], [5] より、移動経路情報を用いて車両の未来の位置を予測することで通信遅延の増加を抑制することができると考えられるため、本研究でも移動経路情報を用いる。

3.2 RSU を用いたエッジコンピューティング環境における 再配置手法の研究

車両ネットワークに MEC を導入した VEC は、リアルタイム性が高く車両では高負荷となるようなアプリケーションを処理することを目的として研究が行われている [1]。図 3.2 に示すように RSU は通信可能範囲が限られているため、RSU を用いたエッジコンピューティング環境で低遅延なアプリケーションを実行するには通信の切断や車両の移動を考慮してアプリケーションの実行場所を再配置する必要がある。この RSU を用いたエッジコンピュー

3.3 関連研究の問題点

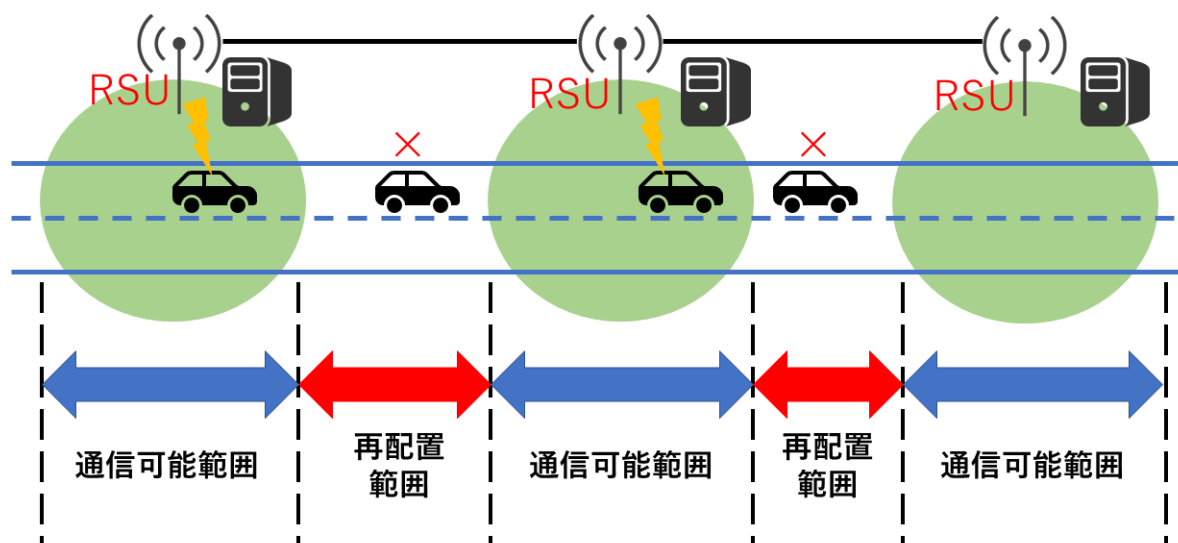


図 3.2 再配置

ティング環境における再配置の研究として、Vehicular Edge Computing におけるコンテナ応答時間を削減するためのコンテナ再配置手法 [2] がある。この研究では、車両の移動方向を考慮し予測した通信切断時刻に各計算資源の CPU 使用率、通信遅延などの指標を収集し、最適な再配置先を決める手法を提案している。コンテナの処理時間の推定を T_{comm} 、コンテナの伝送遅延 T_{trans} と通信遅延は T_{dis} とし、これら 3 つの合計が最小となる計算資源に再配置を行うことにより、コンテナとの切断時間を抑えつつアプリケーションの応答時間を削減できている。

3.3 関連研究の問題点

以下に関連研究の問題点を説明する。

(問題 1) [4], [5] では、ホップ数の増加を抑制できているものの、配置先の選択ではユークリッド距離を用いて近い計算資源に配置しているため、計算資源の負荷状況を考慮していない。そのため、配置した計算資源の負荷が高い場合にアプリケーション実行時間

3.3 関連研究の問題点

が大きくなる可能性がある。

(問題 2) [2] では、RSU との通信切断時に再配置を開始するが、次の RSU と通信開始するまでの時間がコンテナの再配置時間よりも小さい場合、再配置が間に合わずサービスが受けられない。

(問題 3) [2] では、再配置先の計算におけるコンテナの処理時間の推定には、RSU との通信が切断した時刻 t の CPU 使用率 $\delta_{t,k}$ と要求計算時間 COM_{k,C_n} を用いて式 (3.1) から求めており、再配置による負荷や負荷の変化を考慮していない。そのため、サービス提供前のコンテナが多く存在する場合や再配置が同じ RSU に集中した場合に、負荷が増加しアプリケーションの処理時間が増加する可能性がある。

$$T_{comm} = COM_{k,C_n} / (1 - \delta_{t,k}) \quad (3.1)$$

そこで本研究では、次に通信する RSU との通信開始時刻を予測することによる再配置の前倒しと、再配置にかかる負荷を含めたリソース予約状況を用いて通信時間中の負荷を予測することによって、アプリケーションの処理時間を短くする再配置手法を提案する。

第 4 章

環境モデル

4.1 ネットワーク構成

本研究が想定するネットワーク構成について説明する。ネットワークは、図 4.1 のように RSU、中間局、集約局、データセンタの 4 層構成とする。それぞれの設備について以下で説明する。

4.1.1 通信設備

(1) RSU

ネットワークのエッジ部分であり、通信機器と計算資源が合わせて設置されており、無線を用いて車両と通信ができ、かつ有線を用いて他の RSU やクラウドサーバと通信することができる。車両と物理的な距離が近いので高速な通信が可能ではあるが、通信できない範囲が存在する。

(2) 集約局・中間局

集約局は、一定範囲内の RSU を管理しており、中間局は複数の集約局をまとめたエリア情報を管理している。全ての RSU は所属するエリア管理する集約局に接続しており、集約局はマップ全体を管理する中間局に接続している。

4.1 ネットワーク構成

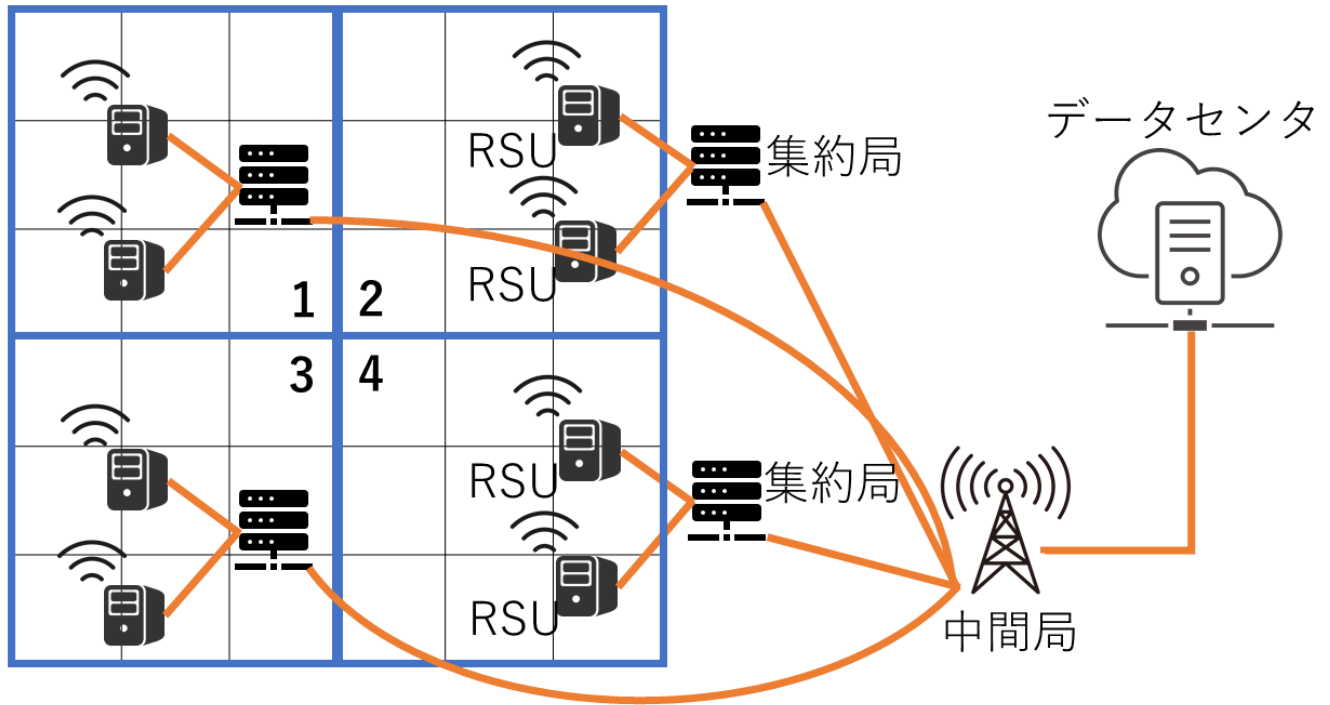


図 4.1 環境モデル

(3) データセンタ

データセンタには豊富な計算資源が設置されている。また、データセンタでは、全ての中間局の情報を管理している。

4.1.2 通信遅延

RSU 同士の通信には集約局や中間局を経由して通信を行う必要がある。具体的な通信遅延を図 4.2 を用いて説明する。車両 v が通信する RSU_{r_i} と同じ集約局と接続している RSU_{r_j} でアプリケーションを処理する場合は、1つの集約局を経由するため2ホップ、 RSU_{r_k} でアプリケーションを処理する場合は、2つの集約局と1つの中間局を経由するため4ホップとなる。RSU から集約局、集約局から中間局までの通信遅延は等しいとして1ホップの通信遅延を D_{hop} とすると、 RSU_{r_i} から別の RSU_{r_j} への通信遅延を D_{r_i, r_j} は以下のようになる。

4.1 ネットワーク構成

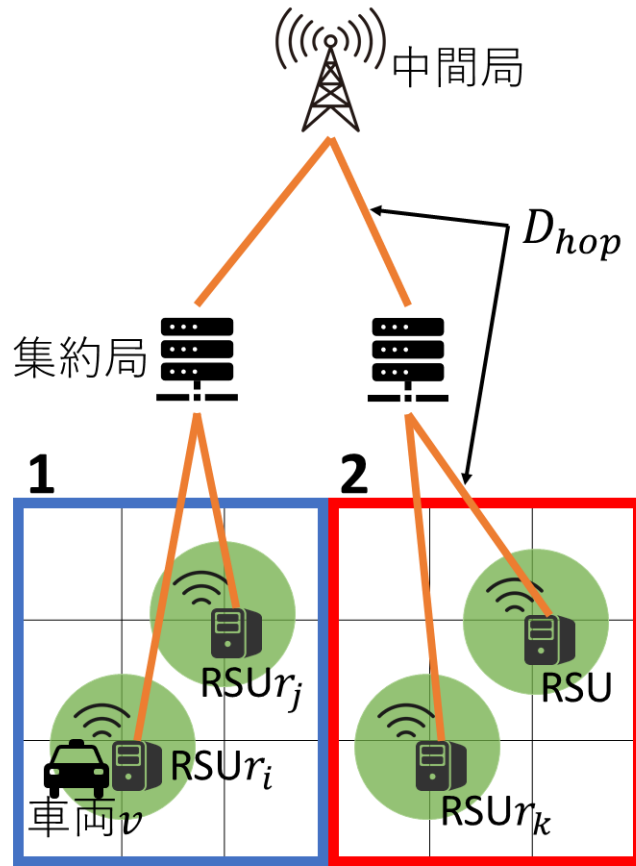


図 4.2 通信遅延

同じ集約局の場合

$$D_{r_i, r_j} = 2D_{hop} \quad (4.1)$$

異なる集約局の場合

$$D_{r_i, r_j} = 4D_{hop} \quad (4.2)$$

4.1.3 車両の移動および車両と RSU の通信時間

図 4.3 に車両 v_k の移動経路を示す．車両 v_k はある時刻 t_{in} でマップ端の地点 A に出現し，別の端地点 B まで移動する．車両は，カーナビゲーションなどに従って決められた経路を移動し，決められた移動経路以外が走行しないようなモデルを想定している．この移動

4.2 計算資源とアプリケーション

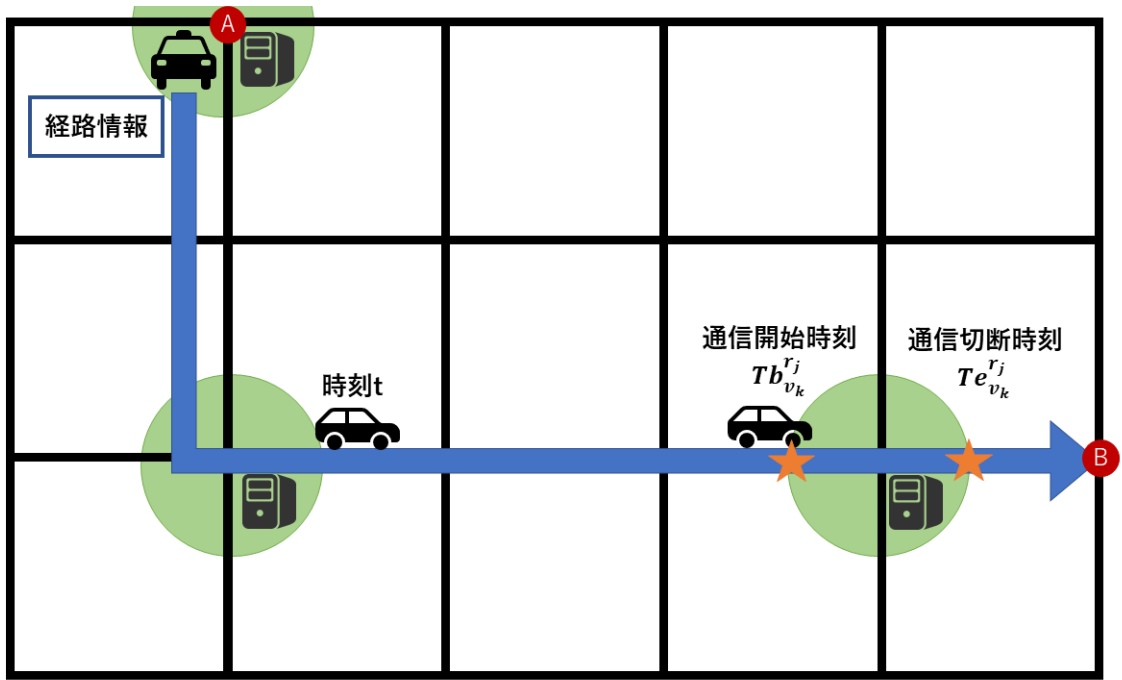


図 4.3 経路情報

経路情報と RSU の通信可能範囲を用いて、車両と各 RSU との通信開始時刻と通信切断時刻を求めることができる。時間 t における車両 v_k の次に通信する RSU_{r_j} の通信開始時間 $Tb_{v_k}^{r_j}$ と通信終了時間 $Te_{v_k}^{r_j}$ は、 $T_{comm}(t, v_k)$ で求まるとする。

$$T_{comm}(t, v_k) = (Tb_{v_k}^{r_j}, Te_{v_k}^{r_j})$$

4.2 計算資源とアプリケーション

RSU が持つ計算資源の情報を表 4.2 に示す。本研究では、RSU とデータセンタに計算資源が配置されていることを想定している。ただし、データセンタの計算資源はリアルタイム性が満たせないものとして通常時は使用せず、RSU の計算資源が枯渇した際に一時的に利用するものとする。

本研究が想定するアプリケーションは、1 種類のアプリケーションであり単独の RSU で処理可能なものとする。本来、アプリケーションを実行する際は CPU の処理能力、コア数、

4.3 リソース管理・リソース予約

表 4.1 RSU が持つ情報

RSU が持つ情報	形式
座標	(x, y)
最大リソース量	P_{cap}
AP リソース	P_{res}
単位アプリケーション処理時間	T_{ap}
集約局	接続している集約局
通信車両	現在通信している車両情報
リソース予約情報	$U_{r_i,t}$

使用メモリ量など様々なリソースが存在するが，シンプルなモデルで評価するため本研究におけるアプリケーションのリソースは単一のものとする．RSU が持つ計算能力を P_{cap} ，RSU の計算能力を 100% 使用した場合のアプリケーション処理時間 (以降，単位 AP 処理時間と呼ぶ) を T_{ap} ，アプリケーションの実行に必要なリソース量を P_{res} する．

4.3 リソース管理・リソース予約

各 RSU は時間ごとの使用リソース量を管理する．図 4.4 に時刻 t_b から時刻 t_e までのリソース予約量を示す．図 4.4 のように， RSU_{r_i} のリソース予約量の集合を U_{r_i} とし，時刻 t における RSU_{r_i} のリソース予約量を $U_{r_i,t}$ として時間ごとの予約リソース量を管理する． $U_{r_i,t}$ の初期値は 0 であり，リソース予約を行う場合は要求リソース量をこの値に加える．車両が RSU の通信可能範囲内にあり，かつ再配置が完了してアプリケーションが実行可能な場合は P_{res} ，VM 起動中は $P_{res}/2$ を必要なリソースとして予約する．

4.3 リソース管理・リソース予約

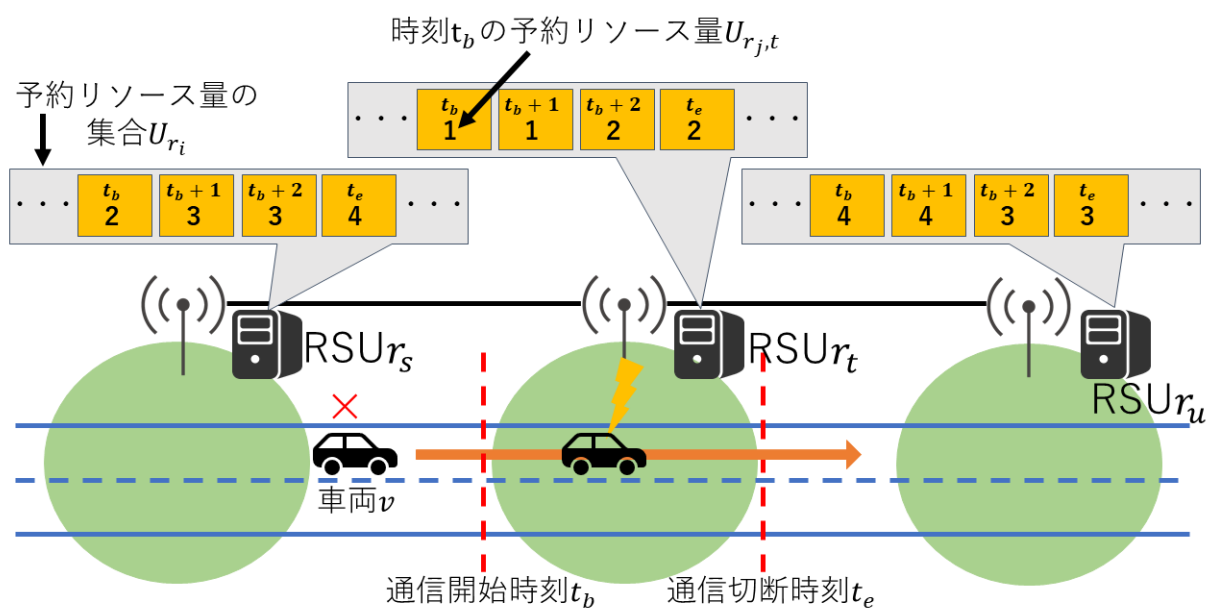


図 4.4 リソース予約

第 5 章

提案手法

5.1 制約条件と目的関数

車両 v_k がマップ内で RSU と t_{total} 秒通信した時、時刻 t におけるアプリケーション実行時間と通信遅延を合わせたアプリケーション処理時間 (以降は AP 処理時間と呼ぶ) は、再配置されたリソース量が RSU の最大リソース量 P_{cap} 未満の場合は式 (5.1), P_{cap} 以上の場合は RSU_{r_i} の時刻 t における超過した分のリソース量 $U_{r_i,t}^{over}$ を用いて式 (5.2) で求まるとする。また、車両 v_k の t_{total} 秒分の平均処理時間 $R^{ave}(v_k, t_{total})$ は式 (5.3) で求まるとする。以下に提案手法の制約条件と目的関数を示す。

$$f(r_i, v_k, t) = T_{ap} * P_{cap} / (P_{cap} - U_{r_i,t}) + D_{r_i,r_j} \quad (5.1)$$

$$f(r_i, v_k, t) = T_{ap} * (P_{cap} + U_{r_i,t}^{over}) / P_{res} + D_{r_i,r_j} \quad (5.2)$$

(1) 制約条件

制約条件は、RSU と通信開始からアプリケーションが処理できるまでの遅延時間 (以降は開始遅延) である D_{wait} が 0 とする。

5.2 再配置の前倒し

(2) 目的関数

目的関数は、式 (5.3) から求まる AP 処理時間 $R^{ave}(v_k, t_{total})$ を小さくすることを目指す。

$$R^{ave}(v_k, t_{total}) = \sum_{t=1}^{t_{total}} f(r_i, v_k, t) / t_{total} \quad (5.3)$$

5.2 再配置の前倒し

提案手法では、移動経路情報を用いて各 RSU との通信開始時刻を取得することで、車両と RSU の通信切断時間が再配置時間より小さい場合でも通信開始に間に合うよう再配置を前倒す。このことを、図 5.1 を用いて説明する。図 2 では、車両 v_k は、RSU r_j に時刻 Tb_{v_k, r_j} に到着する。VM の移送に要する時間と VM の起動時間の和である再配置時間を T_m としたとき、RSU r_j への再配置が時刻 Tb_{v_k, r_j} までに完了するには、再配置時間 T_m 前倒して時刻 $(Tb_{v_k, r_j} - T_m)$ に再配置を開始することで、RSU r_j と通信開始した際の開始遅延 D_{wait} を 0 にできる。

5.3 再配置先選択指標

RSU のリソース使用状況は他の再配置によって変化するため、再配置を行う時刻でリソースに余裕があったとしても、その後リソースが逼迫する可能性がある。そのため、提案手法では通信時間の変化を考慮した平均負荷率を再配置指標として使用する。この平均負荷率から推定平均 AP 実行時間を求め、この値と通信遅延を合わせた推定平均実行時間が最も小さい RSU に再配置を行う。ある車両 v_k が時刻 t_b から時刻 t_e において RSU r_i と通信する場合の RSU r_j の平均負荷率 (5.4)、推定平均 AP 実行時間 (5.5)、推定平均処理時間 (5.6) は以下のように定義する。また、 n 個の RSU の平均負荷率の集合を (5.7) とする。

$$L_{r_j}(t_b, t_e) = \frac{\sum_{t=t_b}^{t_e} (P_{cap} - U_{r_j, t}) / P_{cap}}{t_e - t_b + 1} \quad (5.4)$$

5.3 再配置先選択指標

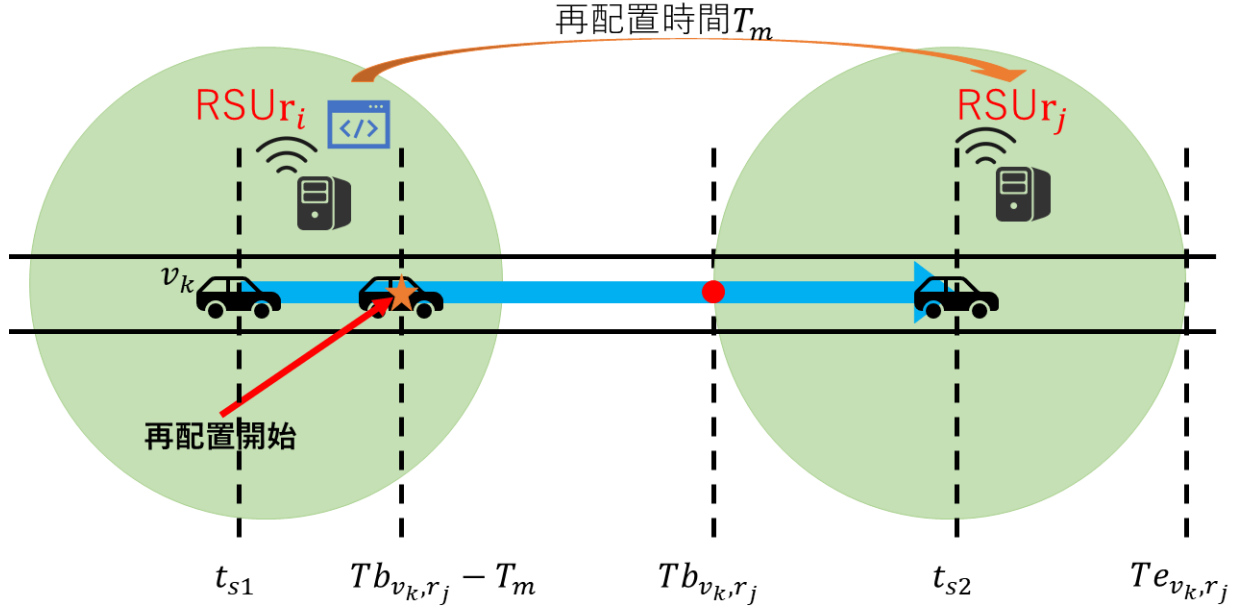


図 5.1 時刻 t_{s1} から時刻 t_{s2} までの経路と到達時間

$$R_{r_j}^{ap} = T_{ap} / L_{r_j}(t_b, t_e) \quad (5.5)$$

$$R_{r_j}^{run} = R_{r_j}^{ap} + D_{r_i, r_j} \quad (5.6)$$

$$L(t_b, t_e) = \{L_{r_0}(t_b, t_e), \dots, L_{r_{n-1}}(t_b, t_e)\} \quad (5.7)$$

5.3.1 混雑度を用いた優先順位

提案手法は、一定時間ごとに車両の情報を収集し再配置計算を行う集中型アルゴリズムのため、同時に複数の車両の再配置が必要な場合がある。どの車両から再配置を行うかの優先順位として、[4] を参考に混雑度を用いる。車両が通信する RSU が現在通信している車両台数を混雑度の指標とし、通信する車両が多い順に再配置を行う。また、時刻 t における混

5.3 再配置先選択指標

雑度は、ある $RSU r_i$ の混雑度は、 $J(t, r_i)$ で求まるとする。図 5.2 のような場合は、 $RSU r_s$ が 3 台、 $RSU r_t$ が 5 台、 $RSU r_u$ が 2 台の車両と通信しているため、混雑度による優先度は $RSU r_t > r_s > r_u$ となる。

5.3.2 再配置先選択アルゴリズム

再配置先選択アルゴリズムを以下で説明する。ここでは、時刻 t における車両 v_k が現在通信している RSU を $r_{v_k, t}^{now}$ 、次に通信する RSU を $r_{v_k, t}^{next}$ とする。時刻 t において $(t + T_m) = Tb_{v_k, r_j}$ となる車両を再配置対象として、各車両が通信するの混雑度が高い順に再配置を行う。

<ステップ 1> 再配置車両の並べ替え

$(t + T_m) = Tb_{v_k, r_j}$ となる車両 v_k が各車両が現在通信している $RSU r_{v_k, t}^{now}$ の混雑度

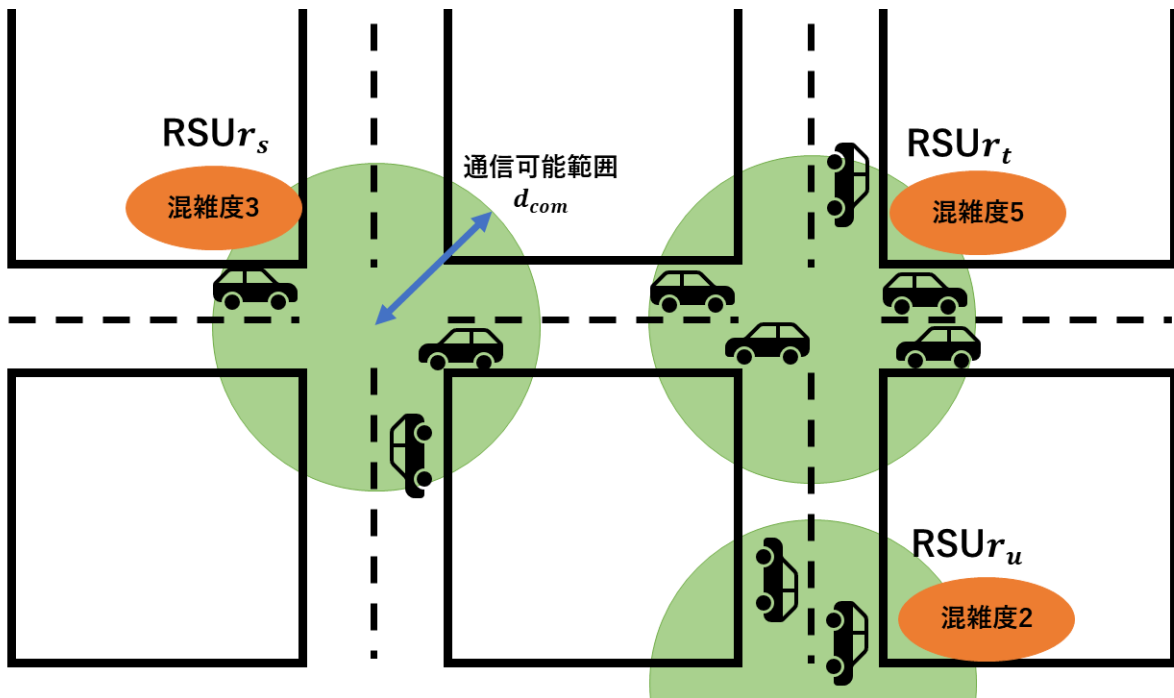


図 5.2 混雑度

5.3 再配置先選択指標

$J(t, r_k)$ を収集し、この混雑度が高い順に再配置が必要な車両を並び替える。優先度が高い車両から以下のステップ 2～ステップ 4 を繰り返す。

<ステップ 2> 再配置先計算に必要な指標の収集

[ステップ 2-1] 各 $RSU r_n$ の時刻 $Tb_{v_k, r_i}, Te_{v_k, r_i}$ 間の平均負荷率を収集し、推定 AP 実行時間を計算する。

[ステップ 2-2] $r_{v_k, t}^{next}$ から各 RSU への通信遅延を収集する。

<ステップ 3> 再配置先 RSU の選出

各 RSU の推定平均 AP 実行時間と通信遅延を足し推定平均処理時間を計算する。全ての RSU の中で推定実行時間が最も小さい RSU を再配置先とする。

<ステップ 4> リソース予約

再配置先 RSU に対して、時刻 t から時刻 $(Tb_{v_k, r_i} - 1)$ までは VM 起動中として要求リソース量 $P_{res}/2$ 、時刻 Tb_{v_k, r_i} から Te_{v_k, r_i} までは要求リソース量 P_{res} でリソース予約を行う。

5.3.3 アルゴリズムの具体例

再配置アルゴリズムの具体例として、図 5.3 を用いて説明する。時刻 t において次の $RSU r_j$ との通信開始時刻を $Tb_{v_k, r_j} = t_b$ 、通信切断時刻 $Te_{v_k, r_j} = t_b + 5$ とし、 $(t + T_m) = t_b$ となる車両 v_k の再配置先を決める。再配置先の対象は、 $RSU r_s$ 、 $RSU r_t$ 、 $RSU r_u$ とし、 $RSU r_u$ は別の集約局が管理しているとする。また、RSU の最大リソース量を 10、アプリケーションの実行に必要なリソース量を 2、計算資源を 100% 使えるときのアプリケーション実行時間を 0.01、1 ホップの遅延を 0.01 とする。

まず、 t_b から $t_b + 5$ までのリソース予約状況を収集し、式 (5.4) をから各 RSU の平均負

5.3 再配置先選択指標

荷率を計算する．例として RSU_{r_s} の平均負荷率は (5.8) のように 0.5 となり，各 RSU の平均負荷率 $L(t_b, t_b + 5)$ は (5.9) のようになる．

$$\begin{aligned} L_{r_s}(t_b, t_b + 5) &= \frac{(4 + 4 + 5 + 5 + 6 + 6)/10}{t_b + 5 - t_b + 1} \\ &= \frac{3}{6} = 0.5 \end{aligned} \quad (5.8)$$

$$L(t_b, t_b + 5) = \{0.5, 0.25, 0.3\} \quad (5.9)$$

次に，平均負荷率を用いて (5.5) から推定平均 AP 実行時間を計算する．先ほど求めた平均負荷率から空きリソースを求め、単位 AP 実行時間 T_{ap} を空きリソースで割ること求める． RSU_{r_s} の場合は (5.10) のように 0.2 となり，各 RSU の推定平均 AP 実行時間は (5.11) となる．

$$\begin{aligned} R_{r_s}^{ap} &= 0.01/(1 - 0.5) \\ &= 0.02 \end{aligned} \quad (5.10)$$

$$R^{ap} = \{0.02, 0.013, 0.014\} \quad (5.11)$$

この推定平均 AP 実行時間に各 RSU までの遅延を足し，推定平均処理時間を求める． RSU_{r_s} は 2 ホップ， RSU_{r_u} なのでそれぞれの通信遅延を足すと以下のようなになる．

$$R^{all} = \{0.04, 0.013, 0.044\} \quad (5.12)$$

最も推定平均処理時間が小さいのは RSU_{r_s} なので， RSU_{r_s} に対して時刻 t から $t_b - 1$ までは要求リソース量 1，時刻 t_b から時刻 $t_b + 5$ までは要求リソース量 2 でリソース予約を行う．

5.3 再配置先選択指標

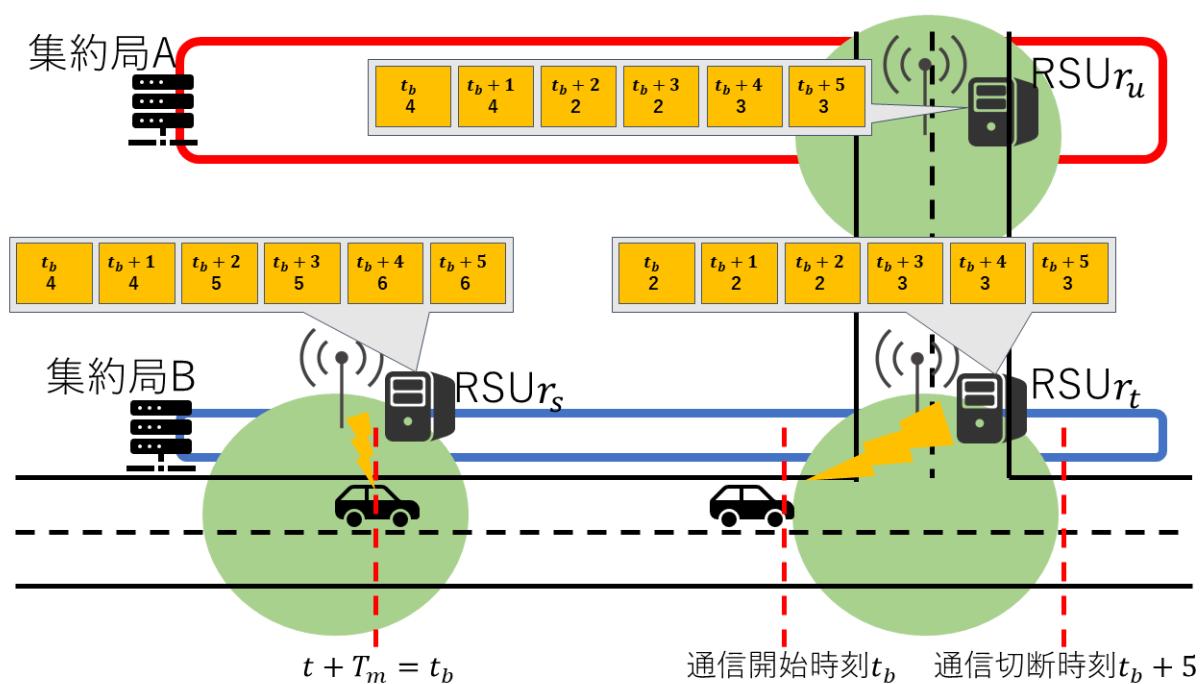


図 5.3 時刻 t におけるリソース予約状況

第 6 章

評価

6.1 評価方法

提案手法および比較手法のアルゴリズムを実装し、再配置のシミュレーションを行う。図 6.1 の様に、格子状のマップに通信範囲が重ならないように RSU をランダムに配置し、走行シミュレータ SUMO[6] を用いて車両の移動や RSU と車両の通信のシミュレーションを行う。車両は、図 6.1 の赤線で囲んだ範囲からランダムに出現位置と退出位置を選び走行させる。シミュレーションのパラメータは表 6.1 の通りである。シミュレーションでは、車両のマップ出現時に RSU が近すぎる場合、再配置が間に合わず開始遅延が発生する可能性があるため、結果を収集する際は各車両がマップ出現から 20 秒後のデータを使用する。また、マップ内の車両が少なすぎる場合は比較手法との差が小さくなり平均を計算した際に差が出てくるため、結果を収集は RSU と通信している車両数が 50 台以上の場合とする。

評価項目は、平均処理時間、平均開始遅延時間、平均リソース使用率、平均リソース超過数を使用し、比較手法は、既存手法を参考に通信切断時刻から再配置を行う手法（以降は既存手法）、ランダム選択手法を使用する。

6.1 評価方法

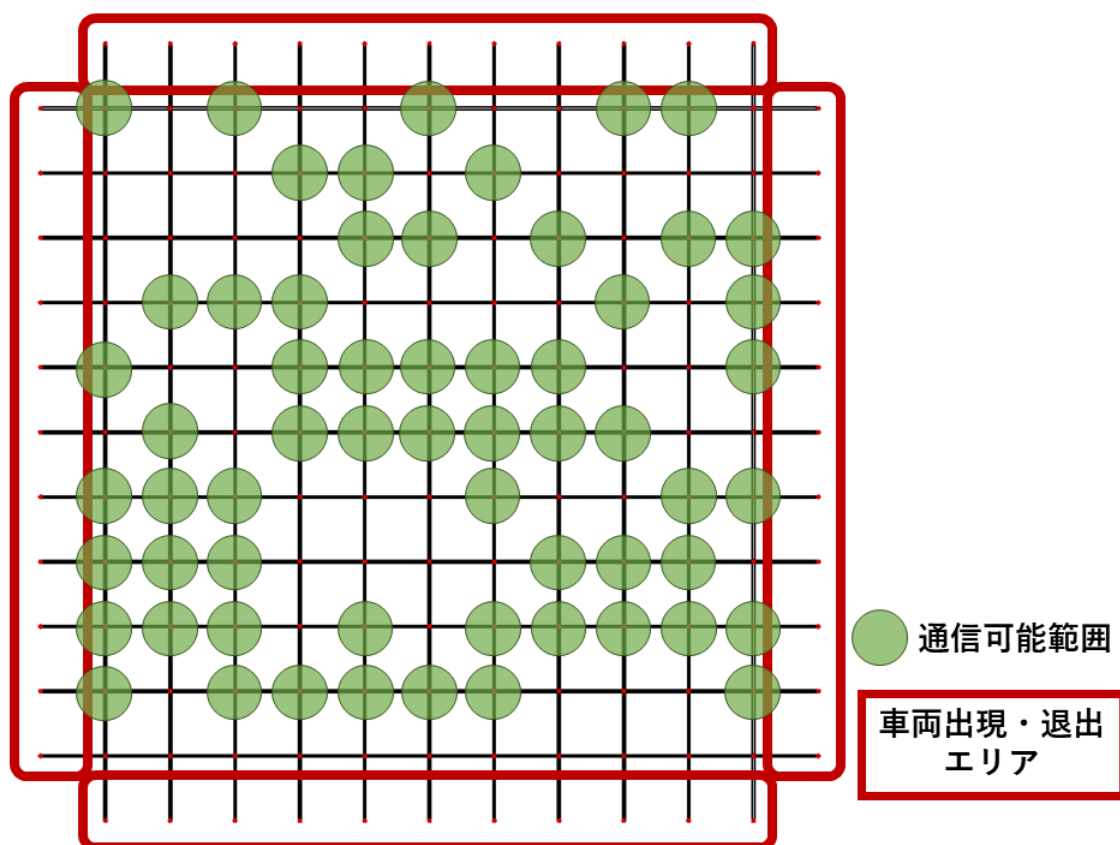


図 6.1 シミュレーション環境

表 6.1 シミュレーションのパラメータ

マップサイズ	2km x 2km
道路	11 x 11 の格子状 (200m x 10)
RSU のリソース量	車両 10 台分
再配置にかかる時間	10 秒
計算能力 100%の時の ap 実行時間	0.01 秒
RSU の通信可能範囲	半径 80m
配置する RSU の数	60 台
車両数	500, 800, 1000 台

6.2 評価指標

(1) 平均処理時間

式 (3.1) を用いて各車両の RSU との通信時間における処理時間を収集し平均を求める。RSU のリソースを超過していた場合は、式 (3.2) の様に RSU の最大リソースを超過したリソース量分、RSU の最大リソース量を増加させた場合の処理時間とする。

(2) 平均開始遅延時間

RSU と通信開始から何秒後に再配置が完了しアプリケーションが処理できる状態になるか全ての車両の平均を求める。

(3) 平均リソース使用率、平均リソース超過数

各時間における RSU のリソース使用率と再配置リソースが超過している RSU の台数を収集し平均を求める。

6.2.1 比較手法

(1) 既存手法

既存手法を元に、通信切断時刻から再配置を開始する手法。本研究ではシミュレーションを用いた評価のため、既存手法も混雑度による車両の優先度を用いて再配置先を決める。

(2) ランダム選択手法

再配置先をランダムに選択する手法。提案手法と同様に再配置の前倒しと混雑度の高い車両から再配置先を計算するが、再配置先はランダムに選択する。

6.3 結果と考察

6.3.1 平均開始遅延

図 6.2 に平均開始遅延を示す。提案手法は、開始遅延が発生しておらず、既存手法は 500 台のとき約 5.18 秒、800 台の時約 3.81 秒、1000 台の時 4.76 秒開始遅延が発生していた。よって、提案手法は再配置の前倒しによって遅延なくアプリケーションを実行することが確認できた。

6.3.2 平均処理時間

図 6.3 に平均処理時間を示す。平均処理時間は、車両が 500 台のとき提案手法が約 24.7 ミリ秒、既存手法が約 25 ミリ秒と差が小さいものの、800 台の時は提案手法が 32.6 ミリ秒、提案手法が 34.4 ミリ秒、1000 台の時は約 42.5 ミリ秒、約 46.3 ミリ秒と車両数が増えるにつれて差が大きくなり、提案手法が最も短いことが確認できる。既存手法が通信切断時刻の負荷状況で再配置先を決めるのに対して、提案手法は通信時間中の負荷の変化を考慮して再

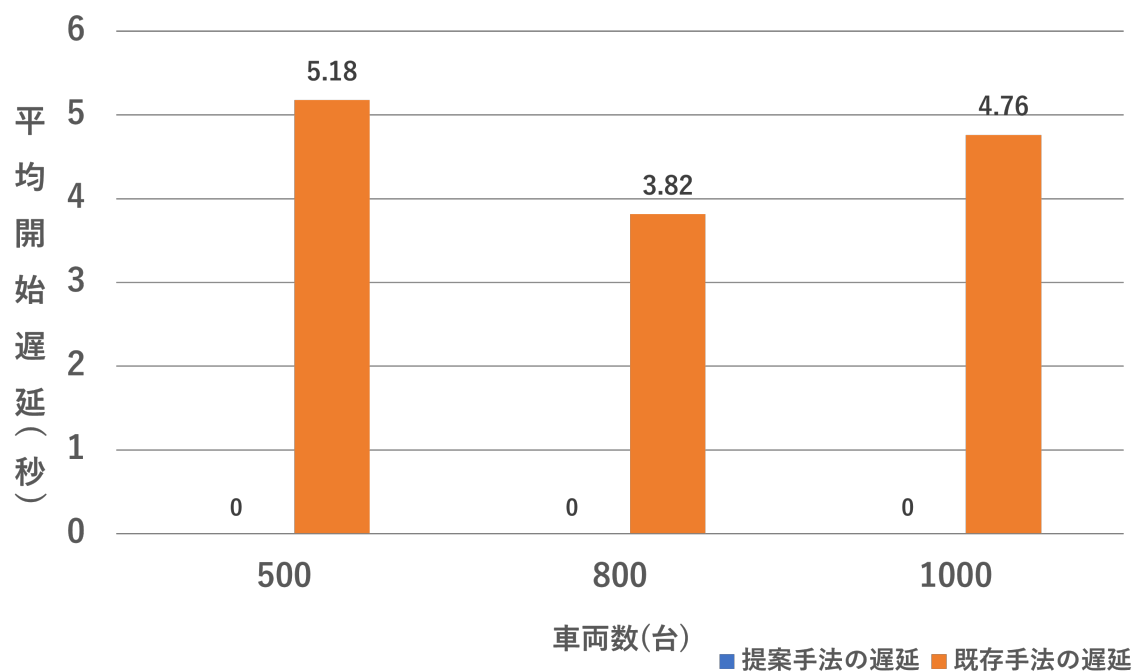


図 6.2 平均開始遅延

6.3 結果と考察

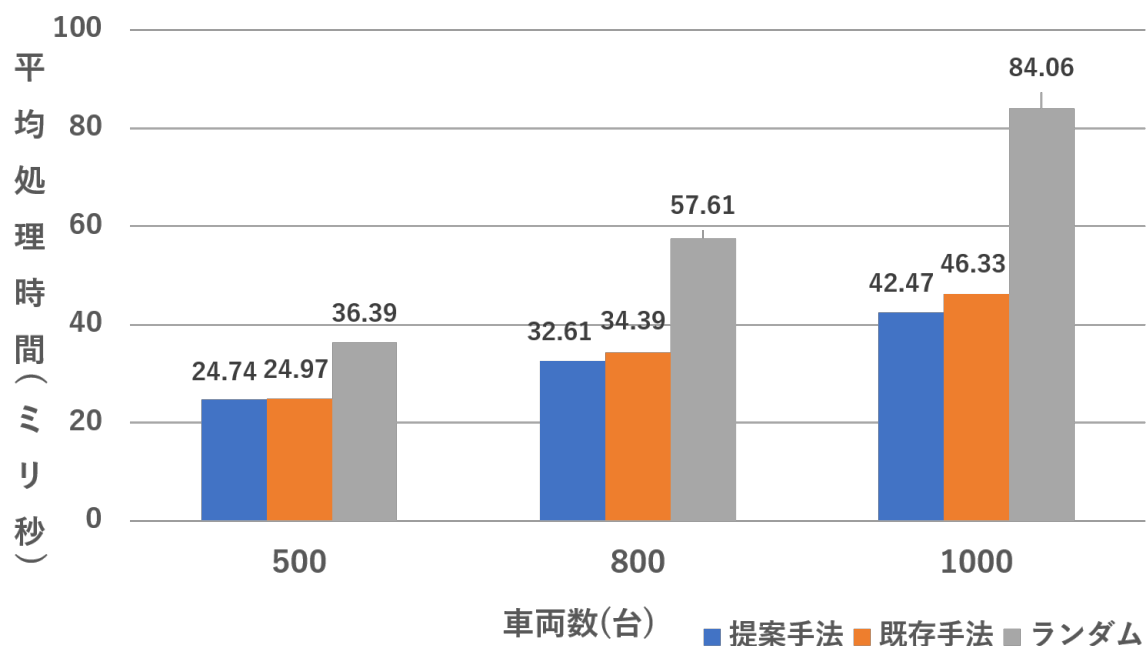


図 6.3 平均処理時間

配置先を決めるため、通信時間全体で見たときに最も効率が良い計算資源に再配置することができる。車両台数が増えるにつれて再配置途中の車両が増えるため、切断時刻の負荷と通信時間中の負荷の差が大きくなることにより、提案手法の方が既存手法に比べ平均処理時間が短くなったと考えられる。

6.3.3 平均リソース使用率

図 6.4 に平均リソース使用率を示す。図 6.4 から既存手法に比べ、提案手法の方が全体的にリソース使用率が高いことが確認できる。図 6.2 にあるように、既存手法は通信開始に再配置が間に合わない場合があるため、リソースを予約していてもアプリケーションが実行できない場合がある。提案手法は、再配置を前倒しすることにより通信開始時刻からアプリケーションを実行できるため、既存手法に比べ提案手法の方がリソースを効率的に利用できることが確認できた。また、提案手法に比べ既存手法の方が平均リソース使用率が低いものの、既存手法の方が平均処理時間が長いことから、提案手法の方がより効率が良い計算資源に再配置できると考えられる。

6.3 結果と考察

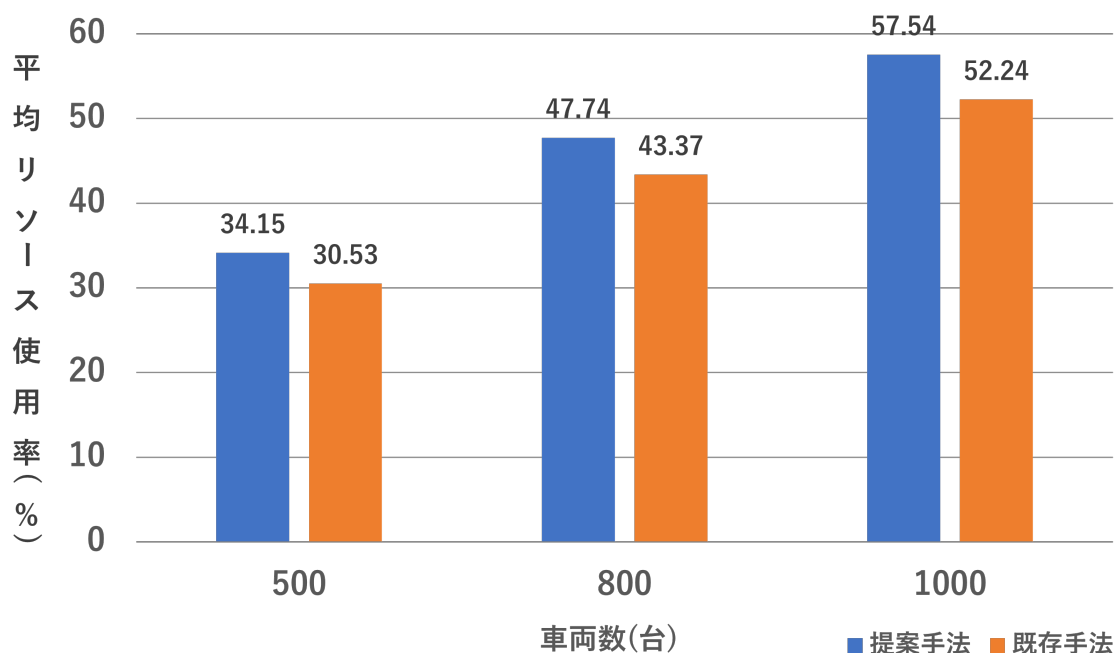


図 6.4 平均リソース使用率

6.3.4 要求リソース超過数

図 6.5 に要求リソース超過数を示す。図 6.5 から、既存手法やランダム選択手法ではリソースが偏り、一部の RSU で再配置リソースが超過していることが確認できる。これは、既存手法が通信切断時刻の負荷で再配置先を決めるため、一部の RSU に再配置が偏り再配置リソースの超過が発生していると考えられる。提案手法では、リソース予約状況から通信時間中の負荷を予測し超過する計算資源には再配置しない。しかし、既存手法では、再配置途中のアプリケーションが多い計算資源に再配置した場合、通信時間中の負荷が増加し要求リソースの超過が発生する可能性がある。リソースの超過により平均処理時間が増加するため、既存手法は提案手法に比べ平均リソース使用率が低いにもかかわらず、平均処理時間が長くなったと考えられる。よって、提案手法は再配置による要求リソースの超過が発生せず、リソースを効率的に利用できていることが確認できた。

6.3 結果と考察

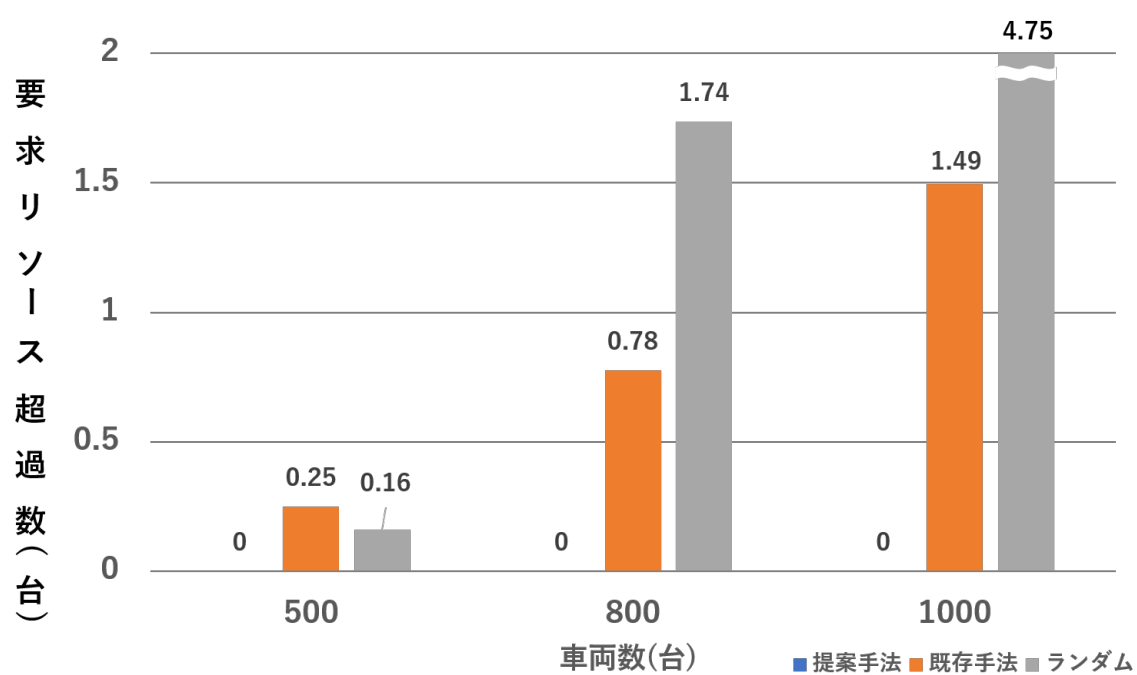


図 6.5 平均リソース超過数

第 7 章

おわりに

7.1 まとめ

近年、先進的な運転支援システムの需要が高まっており、それに伴って車両で処理するアプリケーションが高負荷化し、遅延に対する要求が厳しくなっている。しかし、車両の計算資源には計算能力や消費電力に限りがあるため、車両以外の計算資源を用いてアプリケーションを処理するエッジコンピューティングが注目されている。道路に沿って設置した計算資源と車両が通信することで負荷分散を目指す取り組みとして、RSU(Roadside Unit)を用いたエッジコンピューティング環境において、コンテナ応答時間を削減する再配置手法が提案されている。既存研究は、次の RSU の通信可能範囲への移動時間が再配置時間よりも小さい場合に再配置が間に合わない問題があり、RSU と車両の通信時間中の負荷の変化を考慮していない。そのため、リソース予約状況を考慮した再配置を前倒して開始することで、通信開始時間に再配置を完了させる手法を提案した。評価では、提案手法、既存手法と同様に通信切断時に再配置を開始する手法、ランダム選択手法を平均処理時間、平均開始遅延、平均リソース使用率、平均リソース超過数で比較し、提案手法が有効であることを示した。

7.2 今後の課題

本研究で扱った一極集中型のモデルは、同期的に再配置先を計算する。しかし、現実の交通状況では車両は非同期に通信範囲を移動するため、同期する時刻まで再配置されない問題がある。他にも、対象とする計算資源が多くなった場合に、指標の収集時間や配置先の計算

7.2 今後の課題

時間が膨大になる問題があるため、本手法をそのまま実際の車両ネットワークに適用することはできない。そのため、非同期的かつ自律分散的に再配置先を決めるアルゴリズムを適用することが課題である。また、本研究では格子状のマップに RSU をランダムに配置してシミュレーションを行ったが、実際の道路では規制速度や交通量を考慮して RSU の配置を決める必要があり、最適な RSU 配置を決める手法が研究されている [7]。そのため、より現実に即したモデルで比較し有用性を確認する必要がある。

謝辞

本研究を進めるにあたり，最後まで研究の適切な指導をしていただいた指導教員の横山和俊教授，本研究の副査である敷田幹文教授，高田喜朗教授に心より感謝します．また，研究室の皆様に支えられ研究を続けることができ，心より感謝いたします．

参考文献

- [1] Raza, S., Wang, S., Ahmed, M. and Anwar, M.R.: A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions, *Wireless Communications and Mobile Computing*, Vol.2019, 19 pages (Feb. 2019).
- [2] 豊田睦, 佐竹颯太, 武藤晟, 重野寛: Vehicular Edge Computing におけるコンテナ応答時間を削減するためのコンテナ再配置手法, *情報処理学会論文誌* Vol.63, No2, pp.588-596 (2022).
- [3] エヌ・ティ・ティ・コミュニケーションズ: エッジコンピューティングとクラウドがつくりだす新しい世界, <https://www.ntt.com/business/services/network/m2m-remote-access/bmobile/mvno/edge.html>, (参照 2023 年 2 月 1 日).
- [4] 杉村侑起, 横山和俊: エッジコンピューティング環境での移動計画を用いたリソース割り当て手法の評価, 令和元年年度学士学位論文.
- [5] 有蘭里奈, 横山和俊: 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, 令和 3 年度修士学位論文.
- [6] German Aerospace Center (DLR) :Simulation of Urban MObility (SUMO), <https://sumo.dlr.de/docs/index.html>, (参照 2023 年 2 月 1 日).
- [7] Abderrahim Guerna, Salim Bitam, Carlos T. Calafate.: Roadside Unit Deployment in Internet of Vehicles Systems: A Survey, <https://www.mdpi.com/1424-8220/22/9/3190>, (参照 2023 年 2 月 1 日).