Vol.2017-ITS-68 No.10
2017/3/1

冬道情報サービス構築のための エッジサーバを用いた分散処理フレームワークの検討

市居遼平[†1]　棟朝雅晴[†2]

概要：現在，北海道をはじめとする積雪地帯を対象とする高度道路交通サービスの実現には，多くの課題を抱えている．冬道の路面状況は常に大きく変化しており，高頻度かつ高精度なデータ収集と解析が必要である．しかし，現在広く利用されているクラウドコンピューティングを用いたデータ解析では，多数の車両による大容量データの通信によって，通信帯域の逼迫という課題を抱えることが考えられる．本論文では，クラウドコンピューティングに代わり，ネットワークのエッジにサーバを分散配備して構成される，エッジコンピューティングによる冬道情報サービスを実現するフレームワーク考察する．エッジサーバのリソース状況およびネットワーク状況を評価するアルゴリズムを利用し，エッジネットワーク内に分散された処理リソースと端末側である自動車とのコネクションをダイナミックに切り替えることで，移動体である自動車とエッジサーバとのコネクションによる帯域の使用を軽減する手法について提案する．

キーワード：エッジコンピューティング，エッジサーバ，分散処理，冬道

1. はじめに

現在，北海道をはじめとする積雪地帯における自動車交通の高度なサービス提供は多くの課題を抱えている．冬道の路面状況は常に大きく変化するために，正確な情報提供のためには，高頻度かつ高精度なデータ収集と解析が必要である．しかし，現在広く利用されているクラウドコンピューティングを用いたデータ解析では，多数の車両による大容量データの通信によって，通信帯域の逼迫という課題を抱えることになる．

クラウドコンピューティングとは異なり，ネットワークのエッジにサーバを分散配備して構成されるエッジコンピューティングによって，これらの問題は解決できる．しかし，エッジサーバの限られた処理性能と，端末側である自動車が常に移動し続けている点から，各エッジサーバの処理性能を維持しながら，通信帯域の逼迫を抑える適切な接続環境を維持することが求められている．

本研究の目的は，冬道情報サービスを例とした自動車-エッジサーバ間での分散処理において，処理性能を維持しつつ，通信帯域の逼迫を抑えるフレームワークを提案することである．自動車の数や移動の状況に合わせ変化する，エッジサーバのリソース状況および自動車-エッジサーバ間のネットワーク状況を一元的に評価するアルゴリズムを考案し，ゲーム理論を基にしたセッション切り替えのルールを適用することによって，課題を解決する．また，冬道の情報サービスの一つとして，車載スマートフォンのセンサーを利用した路面凹凸の検出および情報提供を想定し，実際のデータ処理やセッションの切り替えについて実験および考察を行う．

2. エッジコンピューティング

2.1 エッジコンピューティングの概要

エッジコンピューティングとは，ネットワーク上の末端部に近い周縁（エッジ）に接続されたエッジサーバなどのリソースを用いて，サービスを提供あるいは利用する形態のことである．利用者側の端末とエッジサーバは，ネットワーク上において他のサーバと比べ近距離に存在し，ネットワーク遅延の比較的小さな環境下でサービスを提供できる利点を持つ．

エッジコンピューティングは，ネットワーク上の資源をサービス提供の基盤として利用するクラウドコンピューティングと共通点は多いが，次の点において両者は異なる．

- 資源のネットワーク上における相対的な位置：クラウドコンピューティングでは，クラウドサーバをネットワーク上において，通信や情報処理の要求度が高い中心部に配置することが多い．一方でエッジコンピューティングでは，エッジサーバをより利用者に近いエッジに配置する．
- 資源の性能：クラウドコンピューティングでは，大規模データセンタにサーバを集中配備することによって，豊富な資源を柔軟に提供することを可能としている．一方でエッジコンピューティングでは，クラウドサーバに比べて小型のサーバを建物の一角や屋外に分散して配備することを前提としており，資源は各サーバで一定の限度がある．

2.2 冬道情報サービスへの適用における課題

エッジサーバ上で冬道情報サービスを実現するためには，以下の課題が考えられる．

†1 北海道大学　大学院情報科学研究科
Graduate School of Information Science and Technology Hokkaido University
†2 北海道大学　情報基盤センター
Information Initiative Center Hokkaido University

ⓒ 2017 Information Processing Society of Japan

- エッジサーバの配置：エッジサーバはネットワーク上で分散されて配備される．自動車との通信を確立しつつ，十分な計算リソースを提供する為には，要求リソースの予測と，車両に対するリソースの適切な配分が求められるため，エッジサーバを負荷状況に応じて，地域内に分散して配置する必要がある．
- 移動体である車両との接続：自動車は移動体であるため，通信経路は常時変化する．エッジサーバ上で処理を行う際は通信帯域を逼迫しないよう，ダイナミックに適切なサーバへ接続を切り替える必要がある．
- エッジサーバの限定的なリソースの利用：エッジサーバはクラウドサーバに比べ小型で，計算リソースが限定的であるため，各サーバあたりの上限に合わせて，接続する車両の利用リソース量を制限する必要がある．

3. 関連研究

従来のクラウドコンピューティング基盤における，モバイル端末とクラウドとの間の分散処理に関する研究は，これまでに数多く存在する．

Lei Yang ら[1]は，モバイル端末とクラウドサーバ間における，遺伝的アルゴリズムを用いたストリームデータ処理の最適化手法を提案している．アプリケーションをモバイルとクラウドの 2 点で分割して動作するプログラムのためのフレームワークにより，モバイル単体では処理できないストリーミングデータを扱えるようにしたもの．目的はストリームデータ処理のスループット最大化としている．比較的軽量のアプリケーションにおいては高いスループットにて処理が可能である．一方で，遺伝的アルゴリズムを用いていることによって，オンラインでの評価ができないことと，オフロード先をクラウドサーバとしており，他のモバイル端末が利用することによるサーバの負担が考慮されていないという課題がある．

Istvan Ketyko ら[2]は，複数ユーザーが利用する環境下でのナップザック問題を利用したオフローディング手法を提案している．ユーザーが多数存在する環境において，Moblile Edge Computing Server(MECS)に対してモバイル端末上のデータ処理をオフローディングする事により，処理の遅延を最小化することを目的としている．従来のクラウドサーバに対するオフローディングの手法と比べて，オフロード先のサーバの計算リソースを限定してある点が異なる．限られたリソースを多数のモバイル端末間で分割して利用するため，ナップザック問題を適用して MECS のリソースを分配する手法を提案している．この手法は，ゲーム理論に基づくものであり，必ずしも環境全体での最適解を選択するものではないが，端末数の増減などに対し柔軟に適用できるという利点がある．一方で，モバイル端末数に対する MECS の数が多くなるほど，ナップザック問題での解決に時間を要するようになる．そのため，モバイル端末側が自動車などの移動体のときは，端末数の増減が非常に大きいために計算時間の面において問題があると考えられる．

4. 提案手法

この章では，本研究の提案手法である「エッジサーバにおけるスコアリングアルゴリズムを利用したダイナミックなリソース分配」について説明する．2.2 にて説明した課題を解決するために，本研究では，ネットワークの帯域負荷の最小化を目的とした手法を提案する．この手法は「エッジサーバにおけるデータの多段階処理」，「スコアリング」および「エッジサーバによるスコアのランク付け」の 3 点に分けられる．

4.1 エッジにおけるストリーミング処理の全体像

本研究におけるエッジサーバでの処理は，多段階に分割して行うことを前提としている．冬道情報サービスにおいて利用されるデータはその性質上，最終的には各道路の地点毎に評価されるようになっており，解析データをいずれかのエッジサーバで統合する必要がある．そこで，自動車から送信されたセンサーなどの大容量データを，ネットワーク帯域負荷が小さなエッジにて中間処理をした上で，解析データを特定のエッジで統合させるフレームワークを考案した．この方法により，大容量データがネットワーク上で長距離の通信を行うことなく処理ができると考えられる．

まず，センサーによって得られたデータを車両内部でノイズ除去などの基本的な処理を行う．その後，データは中間処理エッジに送信され，解析が行われる．処理が済んだデータは，各地域に存在するデータ統合エッジで最終的な解析が行われ，解析結果がサービス提供される．

冬道情報サービスとして提供されるアプリケーションは，あらかじめ分割して動作することを前提として構築される必要がある．また，分割されたアプリケーションのモジュール間では，受け渡しされるデータ量をあらかじめ決めておく．このことによってモバイル端末，中間処理エッジ，データ統合エッジでそれぞれ必要なリソース量と，各エッジおよびモバイル端末間で通信に要するデータ量をモデル化することができる．

4.2 スコアリング

エッジサーバへのデータ送信のためには，送信するエッジサーバを選択する必要がある．本研究では，多段階でのデータ処理を行うために，通信区間も複数区間に分割される．そのために，モバイル端末と中間処理エッジ間および中間処理エッジ間とデータ統合エッジ間でそれぞれサーバ選択について評価をする必要がある．また，車両側である

ⓒ 2017 Information Processing Society of Japan

モバイル端末は，最終的にデータを送信するエッジサーバが決められている以上，選択できるエッジサーバにも限りがある．そのため，データの最終送信先であるデータ統合エッジが，モバイル端末の接続先候補を提示し，モバイル端末はその候補の中から接続先エッジを選択する方法を用いることとした．この方法によって，適切なサーバを選択するための評価を短時間に行うことができると考えられる．

4.3 エッジサーバによるスコアのランク付け

モバイル端末は，接続先候補の中間処理エッジの中から，最もスコアの高いエッジを選択して接続する．各中間処理エッジでは，自身に接続しているモバイル端末とのセッションのスコアを保持し，これらをスコアの高い順にランク付けする．ランクの上位から計算リソースを割り当てていき，計算リソースをすべて割り当てた場合は，ランク下位にあるセッションに接続先の切り替えを促す．このことによって，リソース量の限度内でスコアが高く通信帯域逼迫の少ないセッションのみが接続されるようになる．

上記のように，アプリケーションを分割可能な状態で構築した上で，スコアリングとランク付けを一定間隔で行うことにより，モバイル端末とエッジサーバ間での接続をダイナミックに切り替える．本フレームワークを用いることで，本研究の目的である帯域使用の逼迫の低減を達成する．

5. スコアリングアルゴリズム

5.1 スコアリングアルゴリズムの概要

スコアリングアルゴリズムは，モバイル端末とエッジサーバとの接続（セッション）をスコアで表し，他のセッションと比較することでダイナミックにセッションの切り替えを出来るアルゴリズムである．

スコアリングアルゴリズムの動作は以下の通りである．

- モバイル端末が，現在位置に対応するデータ統合エッジサーバを問い合わせる．位置情報と対応するデータ統合エッジは一意に決定される．問い合わせ先はクラウドなどのサービスを集中管理するサーバとする．接続先のデータ統合エッジが確定すると，モバイル端末が直接データ統合エッジに接続する．
- 接続されたデータ統合エッジは，候補選択アルゴリズムを用いて，データ統合エッジ-中間処理エッジ間のスコアを算出する．条件に合致するエッジを選択して，選択先候補としてモバイル端末へ応答する．
- モバイル端末は，スコアリングアルゴリズムを用いて提示された候補を 1 つずつ評価する．
- 候補があらかじめ保持していたデータ統合エッジ-中間処理エッジ間のスコアと，モバイル端末側で評価したスコアを，加重和によって最終的なスコアを算出する．
- 最終的なスコアがもっとも高い中間処理エッジを選択して接続する．

このアルゴリズムでは，データ統合エッジにて接続先のエッジサーバの候補をあらかじめ選択しておくことで，モバイル端末側での計算による負荷を軽減し，また選択の高速化を可能としている．

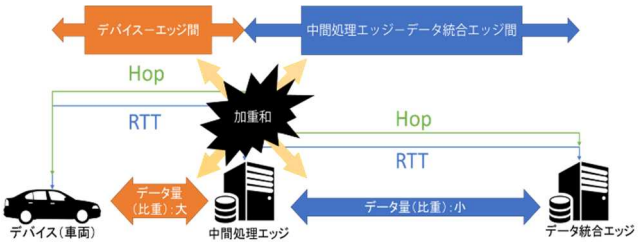

図 1　スコアリングアルゴリズムのイメージ

5.2 候補選択アルゴリズム

データ統合エッジにおいて接続先エッジの候補を選択するに当たり，候補選択アルゴリズムを適用する．候補選択アルゴリズムは，以下の式で表される．

$$\mathrm{Score}_{ij} = \frac{2}{Hop_{ij}} \times \frac{\min_{i \in PE, j \in CE}(RTTij)}{RTT_{ij}}$$

ここでiはデータ統合エッジの ID(i=1,2…PE}であり，j は中間処理エッジの ID(j=1,2…CE)である．Hop_{ij}はデータ統合エッジ i と中間処理エッジ j との間の Hop 数を示し，RTT_{ij}は i と j との間の RTT(Round Trip Time)を示す．ここで，Hop 数を含む項の分子が 2 となっているのは，ネットワークの構造上 Hop 数が 2 より小さくなることはない（アドホックで接続されることはない）という前提のもと設定をしている．よって，Hop 数を含む項は最も Hop 数が小さくなる時(2 の時)に 1 を示す．一方で，RTT を含む項の分子がすべての RTT 候補のうちの最小値をとるようになっているのは，最も小さな RTT の候補の時に項が 1 となるようにするためである．よって Hop 数，RTT のいずれの項も値が大きくなる，すなわち帯域が狭い，または遅延が大きくなる経路をたどる場合にはスコアが小さくなるようになっている．最も Hop 数が小さく，かつ RTT も小さい経路の場合，スコアが 1 になる．なお，Hop 数については大きくなるとスコアが急激に低くなるのは，Hop 数と通信経路とは比例の関係にあり，RTT が小さい場合でも長い経路を通った場合には通信帯域の逼迫に繋がると考えたためである．

5.3 モバイル端末-中間処理エッジ間のスコアリングアルゴリズム

データ統合エッジより送信されてきた候補となるエッジサーバの ID とスコアを受信したモバイル端末は，それらの候補から自機と中間処理エッジの間のスコアを算出する．

スコアリングアルゴリズムは以下の式にて表される．

$$\mathrm{Score}_{kj} = \frac{2}{Hop_{kj}} \times \frac{\min_{k \in D, j \in CE}(RTTkj)}{RTT_{kj}} \times \mathrm{Area}_{ik} \times Avail_{sj} \times Stop_j$$

ここで k はデバイスの ID(k=1,2…D)を表す．Hop_{kj}は，モバイル端末 k と中間処理エッジ j との間の Hop 数を示し，RTT_{kj}は k と j との間の RTT を示す．また，モバイル端末側である車両は移動し続けていることを考慮して，データ統合エッジが存在するエリアと現在地のエリアが異なる場合は，スコアが小さくなる（Area_{ik}）．スコアは，現在地のエリアとデータ統合エッジのエリア差を分母，1 を分子とする．よって，隣のエリアの場合はスコアが 2 分の 1 に，2 エリア離れるとスコアが 4 分の 1 になる．$Avail_{sj}$は，利用するサービス s が，中間処理エッジ j で利用可能であれば 1 を，利用不可能であれば 0 を示す．これは，利用不可能なエッジサーバへの接続を防ぐためである．$Stop_j$は，中間処理エッジ j が新たな接続を一時停止している場合には 0 を，受け入れている場合には 1 を示す．これは，セッションの切り替え待ちをしているエッジが多く，新たなセッションを受け入れられない場合を考慮している．

5.4 最終スコアの算出

候補選択アルゴリズムと，モバイル端末-中間処理エッジ間のスコアリングアルゴリズムによって，2 種類のスコア Score_{ij}，Score_{kj}が算出された．これらのスコアは単独で算出されているため，加重和を用いることによって，モバイル端末-中間処理エッジ-データ統合エッジ間のセッションのスコアを算出する．最終スコアは以下の式で表される．

$$\mathrm{Score}_{ijk} = a \times \mathrm{Score}_{ij} + b \times \mathrm{Score}_{kj}$$

ここで，a と b は，データ通信量の比を表す．モバイル端末 k から中間処理エッジ j に対して送信したデータ量 b と j からデータ統合エッジ i に対して送信したデータ量 a の比$\frac{a}{b}$を，加重和の比とする．$a \geq 0$，$b \geq 0$，$a + b = 1$である．Score_{ijk}は$0 \leq \mathrm{Score}_{ijk} \leq 1$で表されるスコアとなり，セッションの帯域逼迫と遅延を示す．モバイル端末上では，各候補についてスコアを算出し，最もスコアの高いエッジに対して接続を行う．

ここで導出したスコアは，Hop 数と RTT を含んでおり，車両の移動や通信帯域の状況によって大きく変化する．よって，スコアの精度が低下することのないよう，一定間隔でスコアを更新する必要がある．一方で，接続中のすべての車両がスコアリングを頻繁に行うと，そのコストが大きくなることが考えられるため，車両の移動による Hop 数と RTT の変化が大きいモバイル端末-中間処理エッジ間のスコアのみ更新することとする．しかし，車両の移動が長時間に及ぶと，最適なエッジが変化することが考えられ，スコアの更新だけでは最も適切ではないエッジサーバに接続し続けてしまう可能性がある．よって，モバイル端末-中間処理エッジ間スコアの更新よりも長い間隔で，スコアリングを繰り返すこととする．このことによって，一定間隔でモバイル端末側からより適切なエッジサーバに移動することができるようになる．

6. エッジサーバにおけるランク付け

6.1 ランク付けの概要

スコアはセッションの帯域逼迫の程度を示す値であり，スコアが低ければ帯域逼迫が大きいことを意味する．また，エッジサーバの計算リソースには限りがあるため，接続できるモバイル端末の数は限られている．よって，よりスコアの高いセッションを優先して各エッジサーバに接続することで，ネットワーク全体の帯域逼迫を小さくすることができると考えられる．以下にランク付けの方法について説明する．

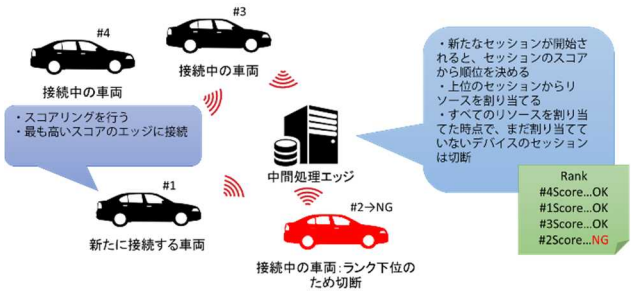

図 2 エッジサーバでのランキングとセッション切り替えのイメージ

6.2 中間処理サーバにおけるランキング

本研究においては，すべてのモバイル端末-エッジサーバ間の接続（セッション）が中間処理サーバを経由している．また，接続先のデータ統合サーバは，モバイル端末の現在地によって一意に決まるため，ダイナミックにセッションが切り替えられるのはモバイル端末-中間処理エッジ間である．中間処理エッジは計算リソースに限りがあるため，接続しようとするモバイル端末が要求するリソース量の合計が，提供できるリソース量を上回る場合には，リソースを割り当てるセッションを選択する必要がある．よって中間処理サーバにおいて，セッションのスコアをランク付けすることによってセッションの選択およびダイナミックな切り替えを行う．

各中間処理エッジサーバは，提供するそれぞれのサービスについて同時に処理をすることができる台数があらかじめ定め，上限を設定しておく必要がある．新たなセッションが発生すると，中間処理エッジは内部に持つランキングより，セッションの順位をつける．上位のセッションから順にエッジのリソースを割り当てて行き，すべてのリソースを割り当てた時点で，未割り当ての下位のセッションは

ⓒ 2017 Information Processing Society of Japan

保持できなくなる．セッションを切断された車両は，新たな接続先を見つけるためにスコアリングを行い，その時点で最も高いスコアの中間処理エッジへと再接続する．

6.3 セッションの切替

各車両は，それぞれのサイクルより適切なエッジへ切り替えるためのスコアリングを行うため，ランキングの上位にあってもセッションを切り替えることがある．中間処理エッジはランキングを保持し，維持できないセッションに対して切り替えを指示するため，常に最大限のリソースを利用可能な状態にしつつ，スコアの高い，すなわち帯域逼迫が比較的小さいと判断されるセッションのみ保持することが可能となる．

7. 実験と考察

7.1 シミュレーションによる帯域負荷の実験

提案手法による負荷軽減の効果を検証するために，実際に自動車を走行させるのと同等の環境をシミュレートすることによる実験を行った．

エッジコンピューティングは，実用化が始まった段階であり，本研究で行ったようなセッション切り替えに関するフレームワークに関する研究は非常に少ない．また，現在 ITS として実用化されているものは，クラウドコンピューティング基盤を用いているものが多い．そのため，本実験では，以下の 2 つの手法と比較し，実験する．

- クラウドコンピューティング基盤の利用 : 冬道情報サービスを，クラウドコンピューティング基盤にて適用する．実験環境においてはクラウドサーバを 1 台のみ設置する．提案手法では，中間処理エッジとデータ統合エッジに分割して処理を行っていたが，車両内部での処理を除くすべての処理をクラウドサーバ上で行うとする．
- 最短距離にあるエッジサーバの利用 : 提案手法であるスコアリングアルゴリズムおよびセッションのダイナミック切り替えを使用せず，車両に最も近い場所に設置されたエッジサーバに接続する．RTT の最も小さなサーバへの接続を試みるが，既に中間処理エッジのリソースの空きが無い状態であれば，次に近い中間処理エッジに接続する．同じエリアに接続できるエッジが無かった場合には，隣接するエッジを探索する．それでもエッジが見つからない場合は，接続しない状態で次のタイミングに再試行する．セッションの切替は，一定の間隔またはエリアが切り替わった場合に行う．

本実験においては，自動車の走行と同じ環境をシミュレートする．提案手法では，エッジサーバの設置のためにエリアを設定する必要がある．各エリアには 1 つのデータ統合エッジが存在しており，データ統合エッジがエリア内の道路状況を解析した結果のデータを保持している．

中間処理エッジは，エリアの分割とは関係なく配置される．本実験では全エリアを対象に均一な密度で配置することとする．また，中間処理エッジの性能はすべて同一としたため，1 台のエッジサーバで処理できるセッション数も定数で与えられる．中間処理エッジにごく近い距離に無線接続用のアンテナがあり，アンテナ-中間処理エッジ間のネットワークについては十分な帯域を持つとする．また，本実験では 1 エリア当たりのエッジサーバの台数を 20 台とした．

試験空間は，2km 四方の区画を 1 つのエリアとし，南北 10 エリア，東西 10 エリアの合計 100 エリアを持つ400km^2 の空間を設定した．自動車はあらかじめ指定された台数が走行し，実験中に台数の増減はしない．各車両は，およそ 20km/h～60km/h の速度でランダムに指定されたスタート地点からゴール地点までを走行する．ゴールにたどり着いた車両は，また新たなランダム地点をスタート地点として，再度走行する．また，車両は目的地へ向かって走行するが，最短距離で進むのでは無く，ランダムな確率によって多少遠回りしながら目的地まで走行する．

比較手法であるクラウドコンピューティングを用いる場合は，1 カ所にのみ存在するクラウドサーバを用いる．車両からクラウドサーバまでの接続には，中間処理エッジを利用する場合のエッジの設置位置と同じ位置にアンテナが存在する．データはこのアンテナを経由し，アンテナから先は有線接続にてクラウドサーバまで接続される．

シミュレートにあたり，各エッジサーバおよびモバイル機器の間にある Hop 数と RTT を設定する．RTT や Hop 数は，実際の距離にある程度比例して大きくなることが明らかであり[3]¥cite{rtt}，本実験においても遠距離での接続では RTT および Hop 数が大きくなるようにする一方で，必ずしも比例することのないよう，また外的要因による影響も考慮し，乱数による調整を行っている．

7.2 実験結果

7.2.1 車両の台数を変化させたときの負荷の変化

この実験では，エッジサーバ 1 台あたりのセッション保持可能数を 60 台とし，またスコアリングおよびエッジ切り替えのサイクルを 10 分，車両の平均移動速度 40km/h，中間処理によるデータ量減少の比率を 1000 倍に設定した上で，シミュレータ内の時間で 60 分間試験を行った結果である．

比較するための数値として，車両 1 台 1 分あたりの帯域使用量を求めた．帯域使用量は，データ量と通信距離の積とした．提案手法では，車両台数 80000 台まではクラウドよりも良い結果を示しているが，90000 台前後からクラウドよりも劣る結果となった．これは，付近で利用できるエ

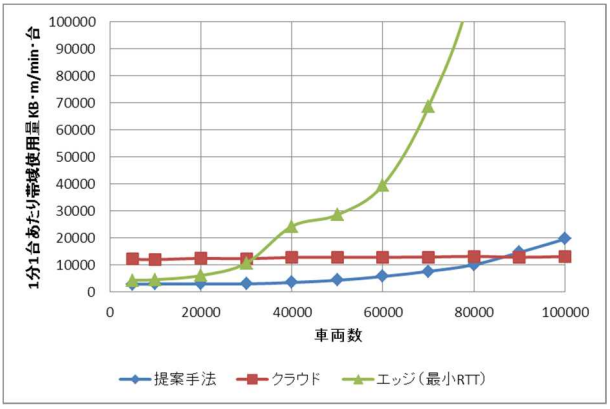

図 3　車両の台数を変化させた時の負荷の変化

ッジサーバのリソースが不足し始めた結果，遠方のサーバに接続することになった結果と思われる．また，最小 RTT による選択をする手法では，常に提案手法よりも帯域使用量の多い結果となった．また，30000 台前後からはクラウドよりも帯域を多く使用する結果となった上に，台数が増えるにつれて非常に大きな値をとるようになった．これは，付近のエッジのリソースがすべて占有されているだけで無く，エッジの切り替えのサイクルがあるために，帯域を使用するセッションでも接続されたままになっていることが原因と考えられる．また，クラウド環境の結果が常に変化していないのは，車両の走行がエリア内で均等に行われている点から，全車両の平均で見た場合には常に安定した数値となると考えられる．

7.2.2 データ量の比率による負荷状況の変化

中間処理エッジにおけるデータ処理によって，車両-中間処理エッジ間に比べ，中間処理エッジ-データ統合サーバでは通信データ量が減少する．しかし，どの程度減少するかは通過した車両の数あるいは，同じ地点や同じ時刻に走行していた車両の数に依存するため，常に変化すると考えられる．この実験では，データの減少度合いと帯域使用量の関係について検証した．車両数は 20000 台，エッジサーバ 1 台あたりのセッション保持可能数を 60 台，スコアリングサイクルを 10 分として 60 分間シミュレートした．

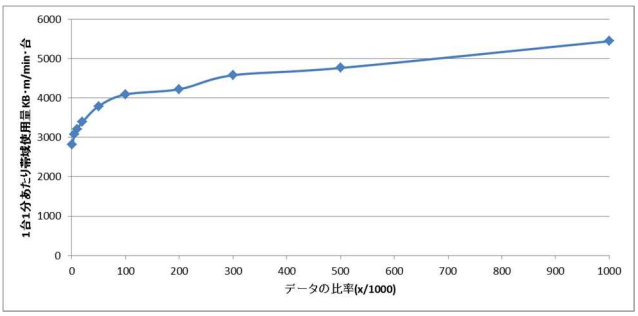

図 4　データ量の比率を変化させたときの負荷の変化

データ量の比が 1000 倍を超える（データ量が 1000 分の 1 まで減少する）場合には帯域使用量は小さくなるが，比率が小さくなる（データ量の減少が抑えられる）と帯域使用量は増加して行き，全くデータ量の減らない時には 5500KB・m/min 前後まで増加した．中間処理エッジでのデータ処理によって，通信量は大幅に減少すると考えられるが，その比率ほど通信帯域使用量が減らなかったのは，車両-中間処理エッジ間の距離が変化するために，実際の通信経路も車両-中間処理エッジ間が中間処理エッジ-データ統合エッジよりも長いということが考えられる．

7.2.3 スコアリング間隔の変化による負荷状況の変化

5.4 で説明したように，スコアリングは計算コストなどがかかるために一定の間隔で実施する．この間隔の変化による通信帯域使用量の変化について検証した．車両数は 60000 台，エッジサーバ 1 台あたりのセッション保持可能数を 60 台，データ比率を 1000 倍として，120 分間シミュレートした．

スコアリングの間隔が短いと，セッションを切り替える頻度が非常に高くなり，適切なエッジに接続することが可能な時間が長くなるため，帯域使用量が小さくなる．しかし，スコアリング間隔が 1～3 分の間は帯域使用量がほとんど変化していない．これは，スコアリングを行っても最適なエッジが変化しないことによるものと考えられる．スコアリング間隔が長くなると，帯域使用量が大きくなる．スコアリング間隔が 20 分になると帯域使用量は 11500KB・m/min 近くなるが，間隔が 30 分までの間には増加が緩やかになっている．これは，中間処理エッジでのランク付けによる切り替えが多く発生することによって増加が抑えられているため，または目的地に到着する車両が増えているためと考えられる．

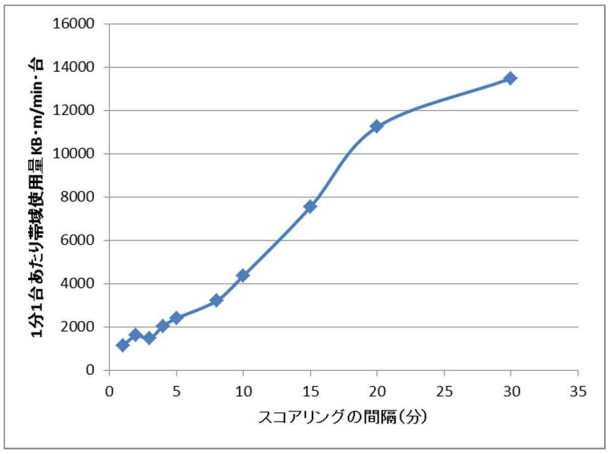

図 5　スコアリングの間隔を変化させた時の負荷の変化

7.3　通信帯域使用量削減効果についての考察

実験を通して，提案手法の効果について検証した．車両台数の変化による負荷状況の変化の検証では，最小 RTT に

よる選択をする手法に比べて帯域使用量が大幅に少なくなることが明らかとなった．これは，ランク付けによるセッションの切り替えが行われたことと，最小 RTT による選択では遠くのエッジを選択するリスクが大きいことが原因と考えられる．一方で，車両の台数の増加によって，提案手法でも帯域使用量が増加し，90000 台で実験を行った際には，クラウドよりも帯域使用量が大きい結果となった．これは，クラウドの場合は最も通信距離が短くなるように頻繁にアンテナを切り替えているのに対し，提案手法では，スコアリングの間隔によって一定時間同じエッジで接続していることが多い点と，車両の台数の増加によって最適なエッジを選択できなくなっていることが原因と考えられる．この問題は，エッジサーバの数を増やすことで解決ができると考えられる．

データ量の比率を変化させた実験では，比率と帯域使用量に一定の相関があることを確認できたが，期待していたほどの帯域量の差は発生しなかった．車両-中間処理エッジ間の距離が大きいことで，中間処理エッジでデータ量が小さくなっていても全体の帯域使用量には大きな影響を与えなかったことが原因と考えられる．ただし，実際にデータ処理をする際には，この比率は一定ではなく，データ収集の内容によって大きく変化するため，比率も含めシミュレートを可能にすることで，より実際の運用時に近い結果を得られると考えられる．

スコアリング間隔の変化に関する実験では，スコアリング間隔が長くなるにつれて，帯域使用量が大きくなることを確認した．テストで使用した車両の台数が 60000 台であったことから，ランク付けをした場合でもリソース量の限度を超えないエッジサーバが比較的多かったと考えられる．一方で 30 分では帯域使用量の増加が緩やかになっていることから，ランキングによるセッション切り替えが多数発生した可能性が考えられる．結果より，スコアリングの間隔を短くするとより帯域使用量が減らせることがわかるが，本実験ではスコアリングおよびセッションの切り替え自体に掛かるコストを考慮していない．そのため，常時スコアリングやセッション切り替えを行うことで，冬道情報サービスやアプリケーションのサービス品質自体が低下する可能性があることを考慮する必要がある．

実験全体を通して，通信帯域使用量を削減したことを確認した．一方で，データ比率の実験やスコアリング間隔の実験からは，帯域使用量があまり削減されない，あるいは帯域使用量が大きく増えることが分かった．スコアリングアルゴリズムによって選択されるエッジサーバは，車両の現在位置を利用しているが，移動先の予測を加味してアルゴリズムを利用することによって，現在より長時間帯域使用量を抑えたまま，同じセッションを続けることができると考えられる．また，実際の使用環境下では通勤ラッシュやイベントなどによって車両が局所的に集中することがあるため，特殊な環境下での本研究のフレームワークの性能についても検証する必要があると考えている．また，関連研究では分散処理環境下におけるサーバの消費電力削減について検証しているものがあり，本研究のような分散環境においても検討が必要であると考えられる．

8. おわりに

本稿では，高度な冬道情報サービスを実現するための帯域使用量を抑えたフレームワークについて提案した．これまでの ITS はクラウドコンピューティング基盤を用いることがほとんどであったが，より高度なサービスを可能にするための，大容量かつ高頻度のデータ通信を可能にするエッジコンピューティングを利用したフレームワークを構築した．エッジサーバ上でのサービス構築のために，「多段階に処理を分割」，「スコアリングアルゴリズムによる通信帯域利用量の数値化」，「ランキングによるダイナミックなセッションの切り替え」という 3 つの方法について提案し，この方法を適用したシミュレーションによる実験を行った．実験では，帯域利用量の削減について確認することができたが，同時により効果を高めるための課題についても新たに発見することができた．

本研究では，シミュレータ上で性能を検証したが，より実際の使用状況に近づけて検証を行う必要があると考えている．実験では，車両の移動はランダムに指定されていたが，特定の箇所に集中させる，局所的に車両の数が変化するなどのイレギュラーな状態下でも実験を行うことで，より実際に近い状況を試験したいと考えている．

参考文献

1 Lei Yang, Jiannong Cao, Yin Yuan, Tao Li, Andy Han, and Alvin Chan(2013) "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing" Performance Evaluation Review, Vol.40, No.4 pp.23-32
2 Istvan Ketyko, Laszlo Kecskes, Csaba Nemes, Lorant Farkas(2016) "Multi-User Computation Offloading as Multiple Knapsack Problem for 5G Mobile Edge Computing" European Conference on Networks and Communications
3 高木 圭一(2009), "教科書には載っていない ネットワークエンジニアの実践技術"
¥¥ "http://gihyo.jp/admin/serial/01/net¥_prac¥_tech/0001"

ⓒ 2017 Information Processing Society of Japan