大阪大学

リアルタイムオペレーティングシステムを用いたエッジコンピューティング環境におけるITS アプリケーションの遅延特性の評価

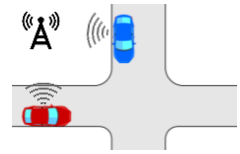
大阪大学基礎工学部情報科学科

村田研究室

八千古嶋 龍

研究背景

- **Multi-access Edge Computing (MEC) への期待**
  - モバイル基地局などユーザに近い位置にコンピューティングリソースを配置しサービスを提供するコンピューティング方式
  - クライアント・サーバ間の伝送遅延を低減し、サービスを低遅延化
- **5G 時代に向けた Intelligent Transport Systems (ITS) への MEC 応用**
  - ITS: 通信技術を用いて道路交通に関する問題を解決するシステム
  - 衝突検知アプリケーションが代表例
    - 処理・伝送の遅延が衝突という致命的な結果に
    - 時間制約が厳しい
- **MEC に処理が集中すると、処理にかかる遅延が悪化**
  - 高負荷な状況下でも一定の応答性能を確保することが課題

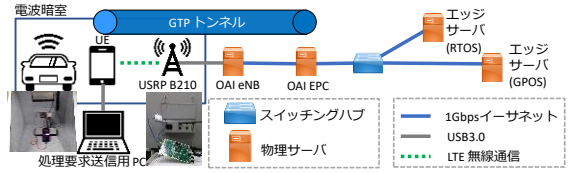


本研究におけるアプローチと研究目的

- **安定した応答性能を確保するためのアプローチ**
  - リアルタイムオペレーティングシステム (RTOS) に着目
    - 高い応答性能、優先度やデッドラインベースのスケジューラを持つOS
    - 処理の時間制約を保証
- **研究の目的**
  - MEC を利用する ITS アプリケーションに対するRTOS 導入効果を実機を用いて測定
- **研究手順**
  1. ITS アプリケーションの一例として、衝突検知アプリケーションを実装
  2. サーバ単体で、 RTOS 導入による処理遅延低減効果を測定
  3. LTE 通信を組み入れた MEC 環境を構築し、ユーザからの衝突検知要求に対する応答遅延を測定

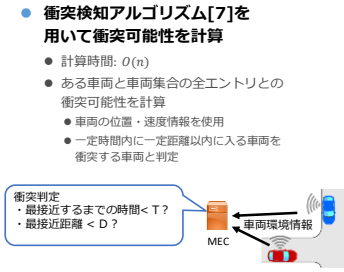
構築した実験環境

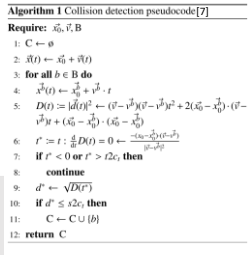
- **UE・エッジ間で通信を行う LTE 通信システムを構築**
  - UE: LTE 通信端末（スマートフォン）と処理要求送信用の PC で構成
  - エッジサーバ 2 台
    - RTOS を導入したものと対照比較のために GPOS を導入したもの
  - LTE 環境
    - モバイル基地局、モバイルコアには、OSS コミュニティ OpenAirInterface (OAI) で実装されたソフトウェアを使用
    - 無線処理にはソフトウェア無線 USRP B210 を使用



衝突検知アプリケーション

- **MEC で車両環境情報を収集・解析し、衝突しそうな車両に警告メッセージを発するアプリケーション**
- **衝突検知アルゴリズム[7]を用いて衝突可能性を計算**
  - 計算時間: $O(n)$
  - ある車両と車両集合の全エントリとの衝突可能性を計算
    - 車両の位置・速度情報を使用
    - 一定時間内に一定距離以内に入る車両を衝突する車両と判定





Algorithm 1 Collision detection pseudocode[7]

Require: $\vec{x_0}, \vec{v}, \mathrm{B}$

1: $\mathrm{C} \leftarrow \emptyset$
2: $\vec{x}(t) \leftarrow \vec{x_0} + \vec{v}(t)$
3: **for all** $b \in \mathrm{B}$ **do**
4: $\quad \vec{x^b}(t) \leftarrow \vec{x_0^b} + \vec{v^b} \cdot t$
5: $\quad D(t) := |\vec{d}(t)|^2 \leftarrow (\vec{v}-\vec{v^b})(\vec{v}-\vec{v^b})t^2 + 2(\vec{x_0}-\vec{x_0^b})\cdot(\vec{v}-\vec{v^b})t + (\vec{x_0}-\vec{x_0^b})\cdot(\vec{x_0}-\vec{x_0^b})$
6: $\quad t^* := t : \frac{d}{dt}D(t) = 0 \leftarrow \frac{-(x_0-x_0^b)\cdot(v-v^b)}{|v-v^b|^2}$
7: $\quad$ **if** $t^* < 0$ **or** $t^* > t2c_i$ **then**
8: $\quad\quad$ **continue**
9: $\quad d^* \leftarrow \sqrt{D(t^*)}$
10: $\quad$ **if** $d^* \leq s2c_i$ **then**
11: $\quad\quad \mathrm{C} \leftarrow \mathrm{C} \cup \{b\}$
12: **return** C

[7] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina,"Performance analysis of C-V2I-based automotive collision avoidance," in *Proceedings of 2018 IEEE 19th International Symposium on A World of Wireless, Mobile and MultimediaNetworks (WoWMoM)*, pp. 1–9, June 2018.

実装した衝突検知アプリケーションの処理の流れ

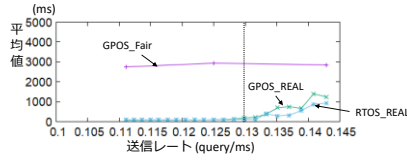
1. **UE から MEC へ処理要求を一定間隔で送信**
   - 処理要求: 車両環境情報、リクエスト ID
   - UE 側で送信時刻をリクエスト ID に紐づけて保存
2. **MEC が処理要求を受信、衝突検知処理及び応答の送信**
   - データベースを更新: 受信した情報の追加、最も古いエントリを削除
   - 受信した車両環境情報とデータベースの全エントリとの衝突可能性計算
   - 処理要求毎にスレッドを作成し衝突検知処理を並列化
3. **UE が応答を受信、応答遅延を記録**
   - 受信した応答に含まれるリクエスト ID から送信時刻を参照し、受信時刻との差を記録

応答遅延の評価 7

- **評価指標**
  - 処理要求のバースト的な発生に対する応答遅延
    - RTOS の効果であるリアルタイムタスク処理能力の上昇により処理集中時の応答遅延がどれほど低減されるかを検証
- **応答遅延の計測方法**
  - 車両台数 50,000 に固定
  - 送信間隔を 7ms ~ 9ms まで 0.1ms ずつ変更し、各送信間隔毎に 4,000 回処理要求送信
    - 短時間に集中する処理要求を再現
  - リソースの競合をバックグラウンド負荷により再現
    - CPU負荷、メモリ割り当て解放負荷、メモリ入出力負荷、ディスク負荷
  - 3 種類のポリシーで遅延を計測
    1) GPOS の通常スケジューリングポリシー、
    2) GPOS のリアルタイムスケジューリングポリシー、
    3) RTOS のリアルタイムスケジューリングポリシー

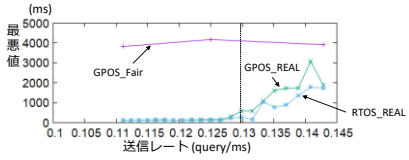
応答遅延の測定結果：平均値 8

- **GPOS の通常スケジューリングポリシーと GPOS のリアルタイムスケジューリングポリシーを比較**
  - リアルタイムスケジューリングの効果を検証
  - 低負荷時高負荷時共に差は大
- **GPOS のリアルタイムスケジューリングポリシーと RTOS のリアルタイムスケジューリングポリシーを比較**
  - RTOS の高い応答性能の効果を検証
  - 低負荷時はさほど差がないが、高負荷時は差が顕著に現れる
  - 0.129 [query/ms] に注目すると、平均値は192.1ms から 129.7ms に低減



応答遅延の測定結果：最悪値 9

- **GPOS の通常スケジューリングポリシーと GPOS のリアルタイムスケジューリングポリシーを比較**
  - リアルタイムスケジューリングの効果を検証
  - 低負荷時高負荷時共に差は大
- **GPOS のリアルタイムスケジューリングポリシーと RTOS のリアルタイムスケジューリングポリシーを比較**
  - RTOS の高い応答性能の効果を検証
  - 低負荷時はさほど差がないが、高負荷時は差が顕著に現れる
  - 0.129 [query/ms] に注目すると、最悪値が 592.5ms から 185ms に低減



まとめと今後の課題 10

- **まとめ**
  - RTOS により最悪時の処理遅延が大幅に削減（研究手順2）
    - 100 台の衝突検知計算の遅延を測定
    - 14.05 ms（ GPOS 通常スケジューリング）から0.02msに削減
  - RTOS により高負荷状況下での応答遅延が低減
    - GPOS と RTOS のリアルタイムスケジューリングを比較
    - 応答遅延の最悪値が 400ms 程度低減
- **今後の課題**
  - 5G システムの構築
    - 伝送遅延の小さい次世代モバイルネットワークでの実機実験
  - エッジ間オフローディング機構の実装・評価
    - RTOS を導入したエッジサーバであっても処理能力には限界がある
    - 複数のエッジサーバ、別拠点のエッジ間での処理連携を図る