大阪大学

4G セルラーシステムを用いた V2X プラットフォームの遅延特性の評価

八千古嶋龍*, 荒川伸一*, 荻野長生†, 北原武†, 長谷川剛††, 村田正幸*

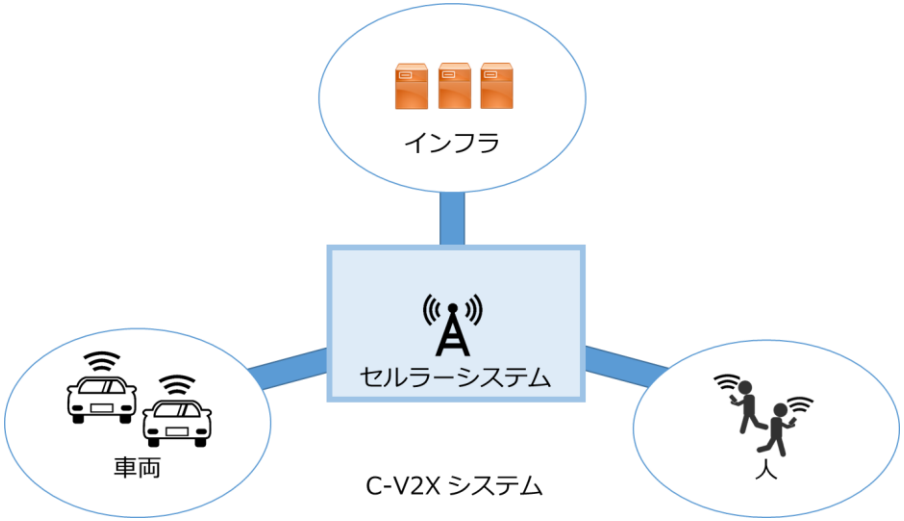
* 大阪大学大学院 情報科学研究科

† KDDI 総合研究所

†† 東北大学 電気通信研究所

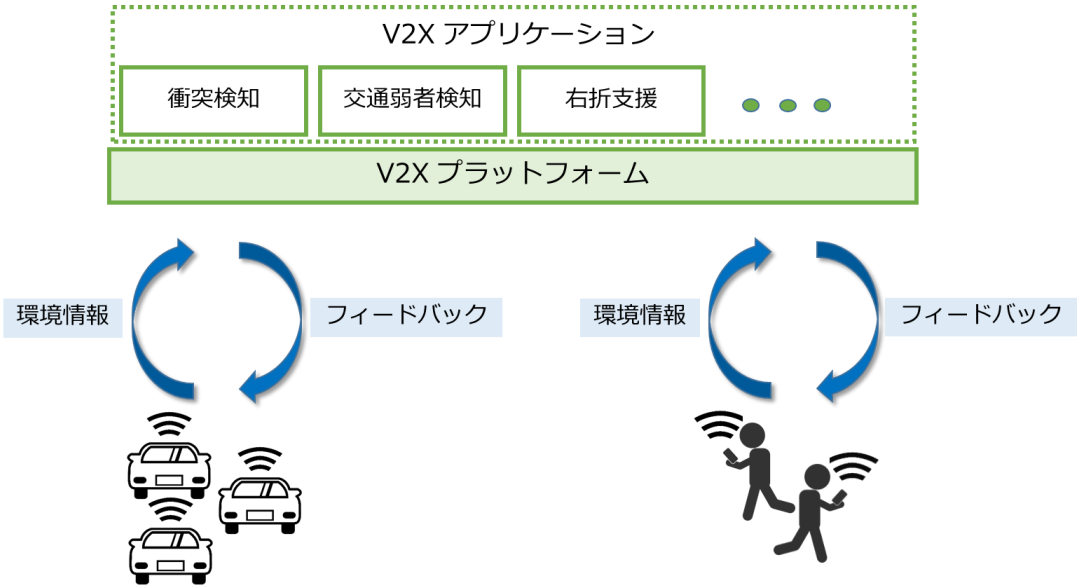
研究背景

- **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
- **Cellular-Vehicle to Everything (C-V2X) の登場**
  - C-V2X: セルラーシステムにより車両や人、インフラが相互接続されたシステム
  - セルラーシステムを介した相互通信による様々な道路交通問題の解決
  - 4G や 5G セルラーシステムを活用した V2X の強み
    - V2X 専用の通信網が不要
    - カバレッジ率の高さ



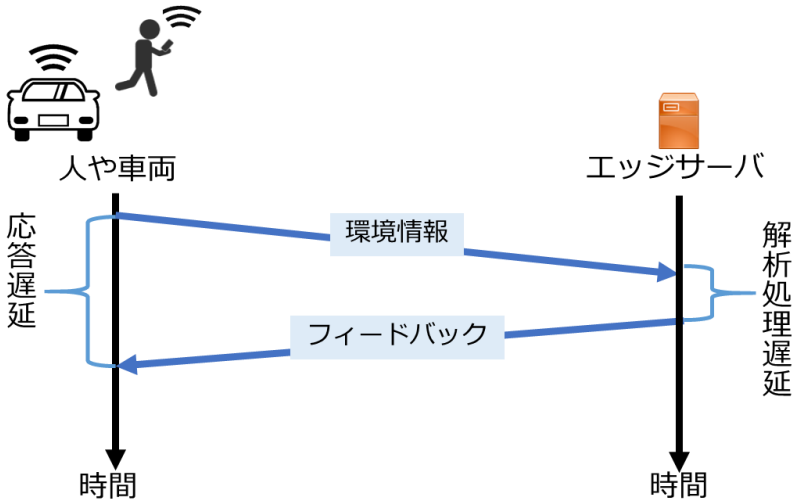
C-V2X プラットフォームへの MEC 応用

- **C-V2X プラットフォーム**
  - 車両や人からの環境情報収集、環境情報の解析、フィードバックを行うための基盤
  - 環境情報を集約し活用する C-V2X アプリケーションが実現
- **MEC による C-V2X プラットフォームの低遅延化**
  - MEC は環境情報の収集とフィードバックにかかる伝送遅延を削減し、リアルタイム性を要する C-V2X アプリケーションの実現に貢献
  - 例、衝突検知アプリケーション
    - 位置速度情報等の動的情報を基に交差点における車両間の衝突を予測・警告



リアルタイム性を要する C-V2X アプリケーションを実現する上での課題 3

- **セルラーシステムの遅延性能の検証**
  - MEC を導入した場合、伝送遅延のボトルネックとなるのはセルラーシステム
  - セルラーシステムの遅延性能を検証し、
    リアルタイム性を要する C-V2X アプリケーションの実現可否を判断する必要性
- **解析処理のリアルタイム性確保**
  - “環境情報収集/フィードバックにかかる伝送遅延”と“解析処理遅延の増大”が
    フィードバック情報のリアルタイム性を低下
  - MEC による伝送遅延の低減には言及されてきたが、
    解析処理遅延のリアルタイム性確保については言及されず



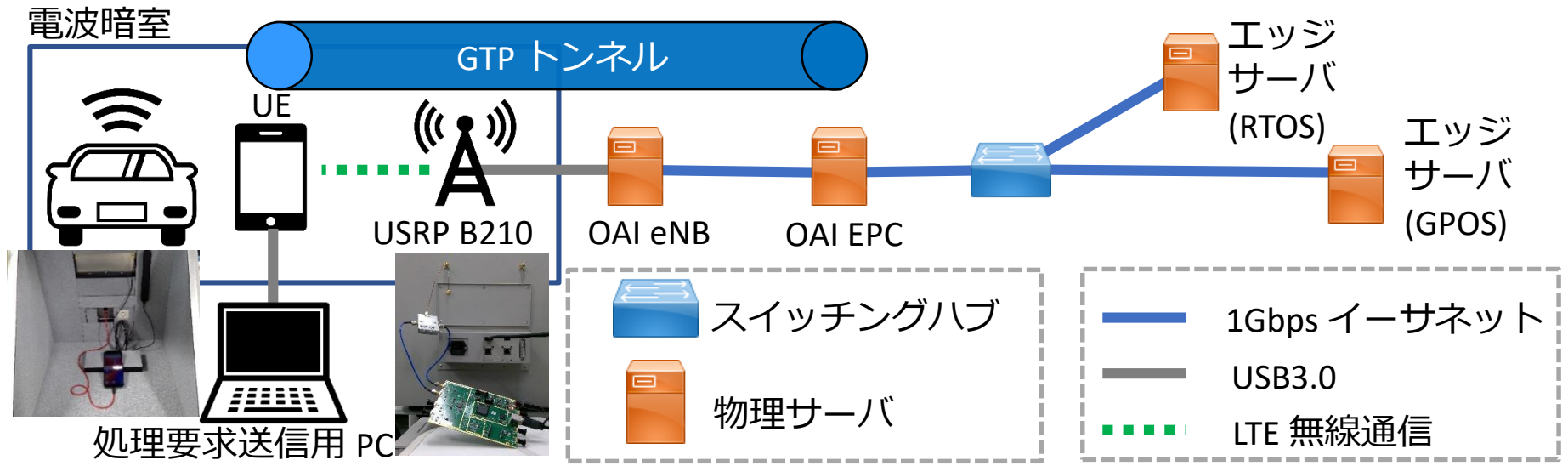
研究目的とアプローチ

- **研究目的**
  - MEC を利用する C-V2X システムの遅延性能の確認
  - C-V2X アプリケーション解析処理のリアルタイム性確保
- **アプローチ**
  - 遅延性能確認のため、MEC を利用する C-V2X システムを実機実装
  - 解析処理のリアルタイム性確保のため、MEC 環境へ Realtime OS (RTOS) を導入
    - RTOS：プロセス応答時間（外部入力からプロセス開始までの時間）が保証される OS
- **研究手順**

1. LTE システムを利用する MEC 環境を構築
2. 遅延性能確認のための C-V2X アプリケーションの実装
   - リアルタイム性を要するアプリケーションの代表例、衝突検知アプリケーションを実装
3. 遅延性能の確認及び、RTOS 導入効果の測定
   - RTOS 導入前後で
     環境情報の収集からフィードバックまでにかかる遅延 (以降、応答遅延) を比較

構築した実験環境

- **UE・エッジサーバ間で通信を行う 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汎用OS (GPOS) を導入したもの
    - RTOS には Realtime Linux と GPOS には Linux を使用
  - 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

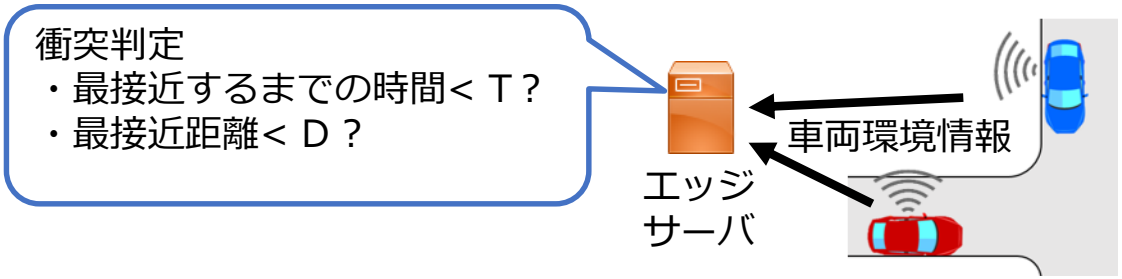
- **エッジサーバを用いて車両環境情報を収集・解析し、衝突しそうな車両に警告メッセージを発するアプリケーション**

- **衝突検知アルゴリズム[1]を用いて衝突可能性を計算**
  - 計算時間: $O(n)$
  - 注目す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
    - 近い未来に最接近し、かつ最接近距離が閾値以下であれば衝突する車両衝突する車両ペアと判定

Algorithm 1 Collision detection pseudocode[1]

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: $\quad \vec{d}(t) \leftarrow \vec{x}(t) + \vec{x^b}(t)$
6: $\quad D(t) := |\vec{d}(t)|^2 \leftarrow (\vec{v} - \vec{v^b}) \cdot (\vec{v} - \vec{v^b})t^2 + 2(\vec{x_0} - \vec{x_0^b}) \cdot \cdot (\vec{v} - \vec{v^b})t + (\vec{x_0} - \vec{x_0^b}) \cdot (\vec{x_0} - \vec{x_0^b})$
7: $\quad t^* := t : \frac{\mathrm{d}}{\mathrm{d}t} D(t) = 0 \leftarrow \frac{-(x_0 - \vec{x_0^b}) \cdot (\vec{v} - \vec{v^b})}{|\vec{v} - \vec{v^b}|^2}$
8: $\quad$ **if** $t^* < 0$ **or** $t^* > t2c_t$ **then**
9: $\quad\quad$ **continue**
10: $\quad d^* \leftarrow \sqrt{D(t^*)}$
11: $\quad$ **if** $d^* \leq s2c_t$ **then**
12: $\quad\quad \mathrm{C} \leftarrow \mathrm{C} \cup \{b\}$
13: **return** C



[1] 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.

衝突検知アプリケーションの処理の流れ

1. UE からエッジサーバへ処理要求を一定間隔で送信

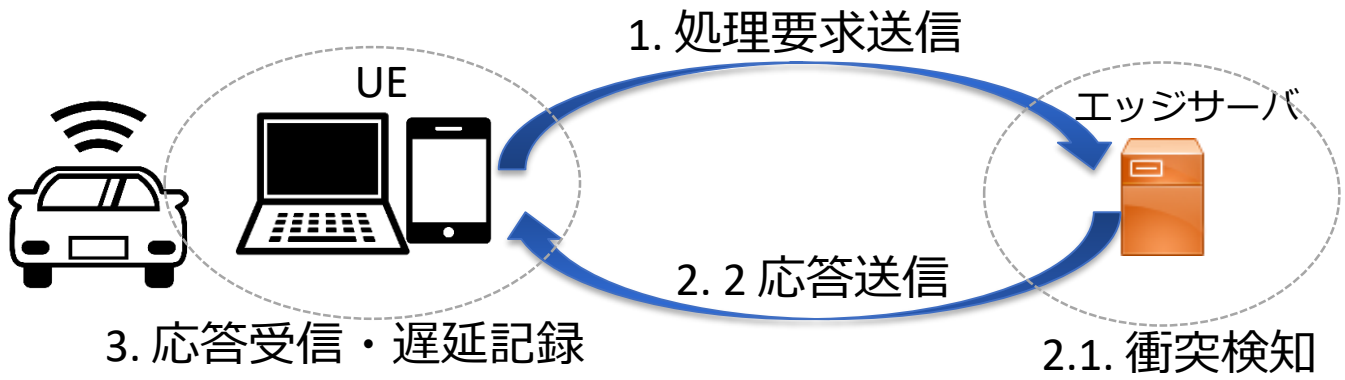
- 処理要求: 車両環境情報、リクエスト ID
- UE 側で送信時刻をリクエスト ID に紐づけて保存

2. エッジサーバが処理要求を受信、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

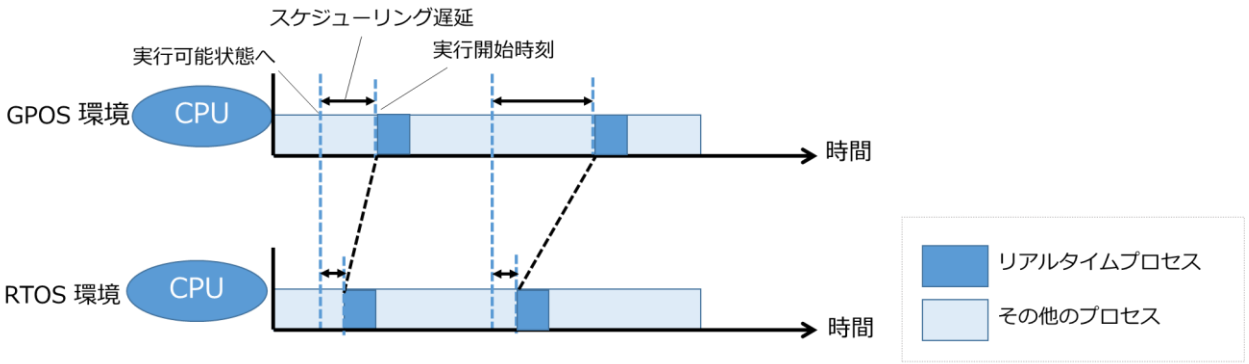
- 受信した応答に含まれるリクエスト ID から送信時刻を参照し、
  受信時刻との差を記録



RTOS による解析処理のリアルタイム性確保

- **GPOS 環境における解析処理のリアルタイム性悪化**
  - GPOS 環境では処理要求の到着から解析処理開始までの遅延が増大
  - 要因はスピンロックによるスケジューリング遅延の増大
    - GPOS では優先度の高いプロセスであっても、
      スピンロックで資源を待つプロセスから CPU を横取り出来ない
- **RTOS による処理要求の到着から解析処理開始までの遅延を削減**
  - スケジューリング遅延がほぼ決定的になるのでリアルタイム性が向上
    - スピンロックを横取り可能なロック機構に置き換え

スケジューリング遅延：プロセスが実行可能状態となってから実行されるまでの遅延
スピンロック：共有資源の同時変更を防ぐためのロック機構。
資源を待つ他のプロセスは資源が開放されるまで CPU を消費しロック取得試行を反復



Realtime Linux の効果によるスケジューリング遅延削減

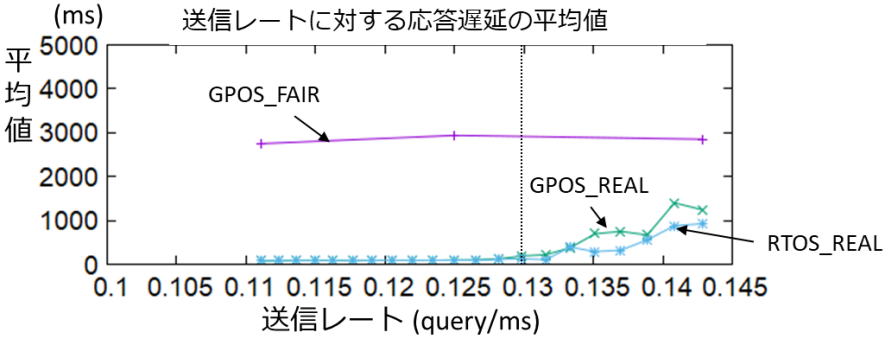
衝突検知アプリケーションの応答遅延測定実験

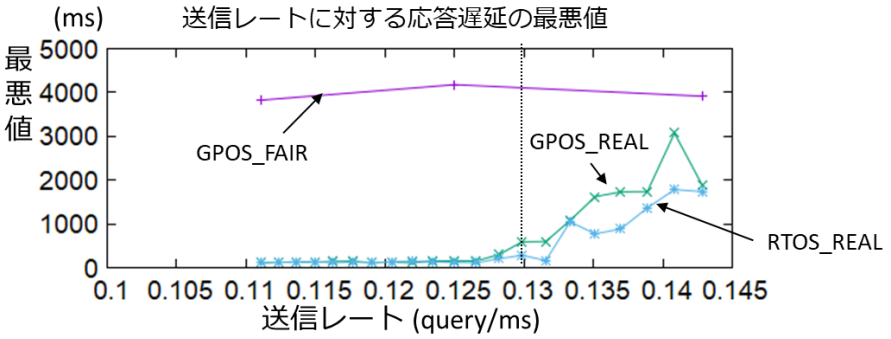
- **実験目的**
  - 構築した C-V2X システムの遅延性能の確認
  - RTOS 導入によるフィードバック情報のリアルタイム性向上効果の測定
- **実験方法**
  - 処理要求の到着率を変更しつつ、
    多様な負荷状況下での衝突検知アプリケーションの応答遅延を計測
    - 1 回目のテストは送信間隔を 7.0ms に設定。 2 回目以降 0.1ms ずつ引き上げる。
    - 1 回のテストでは約 30 秒間（4000 個の）処理要求を MEC に送信
    - 車両環境情報のエントリ数 50,000 に設定
  - リアルタイム性向上効果の測定のため、
    衝突検知処理に異なる条件を適用した場合で応答遅延を計測
    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
  - 想定される資源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

応答遅延の測定結果

[2] 5G Automotive Association,"White Paper C-V2X Use Cases Methodology, Examples and Service Level Requirements",June,2019,https://5gaa.org/wp-content/uploads/2019/07/5GAA_191906_WP_CV2X_UCs_v1-3-1.pdf, [Online; last accessed September 2020]

- **リアルタイムスケジューリングの効果による遅延削減**
  - 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
  - 平均値にして最大 3000ms 程度、最悪値にして最大 4000ms 程度の遅延削減効果
- **RTOS 導入効果による遅延削減**
  - 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
  - 低負荷時には効果が見られないが、送信レートが一定値を超えると差が顕著に
    - 送信レート 1/8.2 (図中点線部) に注目すると 平均値は 192ms から 129ms に、最悪値は 592ms から 185ms に削減
- **構築した C-V2X システムの遅延性能**
  - 低負荷領域 (送信レート ~1/8.1) では、遅延が 80-100ms 程度となり、5G Automotive Association の定める遅延のサービスレベル要件 100ms を充足[2]

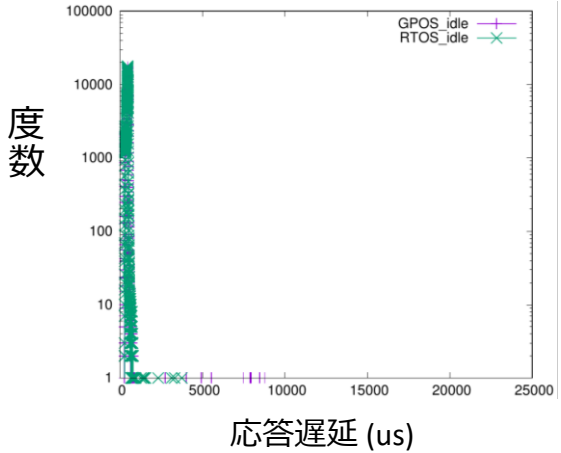




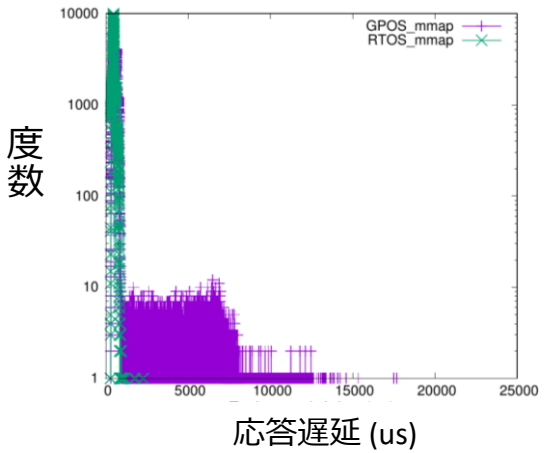
（追加実験）低遅延通信環境での RTOS の効果検証

- **RTOS の効果を検証するため伝送遅延の小さいイーサネット環境で実験**
  - 長時間 (10 万個の) 衝突検知処理要求を送信
  - 車両環境情報のエントリは 1000、送信間隔は 1ms の指数分布に従うように設定
  - バックグラウンドで様々なアプリケーションを動作させテスト
- **追加実験の結果**
  - バックグラウンド負荷無しの場合、RTOS 導入効果で遅延のスパイクが抑制
    - 最悪値が 8.7ms から 3.7ms に、平均値は変化せず 0.4ms
  - プロセス生成と多数の共有メモリを確保を繰り返すアプリケーションが動作している場合、RTOS の効果により応答遅延が抑制
    - 最悪値が 17.6 ms から 2.2ms、平均値が 0.6ms から 0.4ms
  - C-V2X システムにおいても低遅延通信が実現すれば RTOS の効果が顕在化することを示唆

応答遅延の度数分布（バックグラウンド負荷無）



応答遅延の度数分布（バックグラウンド負荷:mmapfork）



まとめと今後の課題

- **まとめ**
  - MEC を利用する C-V2X システムを構築、遅延性能を確認
    - OpenAirInterface を用いた 4G LTE 環境を構築
    - 衝突検知アプリケーションを実装
    - 5G Automotive Association の定める遅延のサービスレベル要件を充足
  - RTOS 導入効果により処理集中時の応答遅延が削減
    - 処理集中時に応答遅延の平均値が 60ms 程度、最悪値が 400ms 程度削減
    - LTE 環境の伝送遅延とそのばらつきにより、RTOS の効果が相対的に小さく
  - 追加実験により低遅延通信環境では RTOS の効果が顕在化することが示唆
- **今後の課題**
  - 5G システムの構築
    - 超低遅延が実現すると言われている次世代モバイルネットワーク環境での実機実験
    - 低遅延なセルラーシステムを用いた V2X システムにおいて
      RTOS により得られる恩恵を明らかにすることが課題