

大阪大学

1

Implementation and Evaluation of a Cellular Resource Allocation Method Based on Prediction of Physical System Behavior in Cyber-Physical Systems

大阪大学大学院 情報科学研究科
情報ネットワーク学専攻 村田研究室 中村 樹

修士論文発表会 2024.2.8

1

2

研究背景

- 通信技術の進化による CPS (Cyber-Physical Systems) の実用化
 - フィジカルシステムの情報をサイバーで分析しフィードバックする制御ループにより現実世界の課題に対処
 - 多岐にわたる領域で応用
 - 高度交通システム、産業、医療、スマートホームなど
- CPS におけるセルラーリソース制御
 - 適切なタイミングで必要な量のセルラーリソースを割り当てる制御が必要
 - 大量のデバイスや様々なアプリケーションが異なる通信要件で同時に動作する
 - 一度に割り当てられるリソースには限りがある
 - セルラーシステムのリソース割当てとサイバーシステムの分析は独立
 - フィジカルシステムの振る舞いの予測を活用していない

図. CPS の例

2

3

研究目的とアプローチ

- 研究目的
 - フィジカルシステムの振る舞いの予測に基づくセルラーリソース制御の実証
- アプローチ
 - 5G システムの拡張
 - MEC (Multi-access Edge Computing) を活用したフィジカルシステムの振る舞いの予測
 - 予測結果をセルラーリソース制御に反映
 - セルラーリソース割当て実験
 - 実験用アプリケーションの構築
 - 予測に基づくセルラーリソース割当て手法の有効性を提示

図. 本研究のアプローチ

3

4

実証に伴う課題

- アプリケーション側でリソースを使い切れない課題が顕在化
 - リソース制御により、セルラーシステムが割り当てる帯域が短い周期で変動
 - 割り当てられる帯域を最大限に利用できる制御が必要
- 解決策
 - アプリケーションが利用可能な帯域情報の受け渡し
 - 利用可能帯域に合わせたアプリケーション側のレート制御

図. アプリケーション側のレート制御

4

5

実験用アプリケーション

- 交通カメラによるモニタリングシステムを想定
 - 要件：車両を映しているカメラは高品質で動画をストリーミング
 - 解決策：MEC を活用した車両位置予測に基づくセルラーリソース制御

図. 実験用アプリケーションの構成

5

6

実験用 5G システムの構築

- RU (Radio Unit)、gNB (gNodeB)、5G-Core から構成
 - gNB：OAI、5G-Core：Open5GS をそれぞれ利用
- 5G システム諸元
 - スロット構成：5 ms 周期、DL、S、UL、4、Mix、1 で構成
 - 帯域幅：100 MHz
 - 周波数方向の RB (Resource Block) 数：273 個
 - RB：端末へのリソース割当て単位

図. 5G システム構築

6

MECによる車両位置予測

7

- マルコフ連鎖を用いて車両の到達可能領域を求める既存手法[14]を利用
 - 車両の位置速度情報から、車両が t 秒後にセグメント s に存在する確率 $x(t, s)$ を算出
- 車両位置予測手法の拡張
 - 空間方向の拡張
 - 複数のセグメントを含む領域 S_t に車両が存在する確率 y_t に変換
 - $$y_t(t) = \sum_{s \in S_t} x(t, s)$$
 - 時間方向の拡張
 - T 秒後まで予測を行い、その中で $y_t(t)$ の最大値 p_t を車両存在確率として定義
 - $$p_t = \max_{1 \leq t \leq T} y_t(t)$$

[14] M. Athoff, O. Sturberg, and M. Buss, "Model based probabilistic collision detection in autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 10, pp. 299–310, July 2009.

7

5G システムの拡張

8

- 車両位置予測結果をリソース割当へ反映
 - gNB が所持している RNTI と MEC が所持している車両存在確率のマッピング

RNTI	車両存在確率
Ox1234	0.1
Ox1111	0.5

8

セルラーリソース割当手法

9

- OAI の Proportional Fair (PF) アルゴリズム
 - 端末のスループット ul_thr_ue を指数平滑化により算出
 - $$ul_thr_ue \leftarrow (1 - a) * ul_thr_ue + a * b$$
 (a : 平滑化定数, b : 直近の送信データ量)
 - PF の係数 $coeff_ue$ をスループットと端末が送信したいデータ量 tbs から算出
 - $$coeff_ue = \frac{tbs}{ul_thr_ue}$$
 - 係数 $coeff_ue$ の小さい順にスケジューリング
- 車両位置予測結果を PF に反映
 - スループットを車両存在確率 p_t で重み付け
 - $$ul_thr_ue \leftarrow \frac{(1 - a) * ul_thr_ue + a * b}{1 + [p_t * 50]}$$
 - 見かけ上のスループットを小さくすることで、確率の高い端末に対して優先的にスケジューリング

9

ストリーミングアプリケーションのレート制御

10

- WebRTC を利用した映像ストリーミング
 - WebRTC (Web Real-Time Communication)
 - 動画、音声、データなどのストリーミング通信を実現、実装は libwebrtc
 - ネットワーク遅延、利用可能帯域に基づいて動画品質を制御
- libwebrtc のレート制御アルゴリズムの拡張
 - GCC (Google Congestion Control) アルゴリズム
 - 帯域に空きがあることを認識してから徐々にビットレートを上げていく幅域制御
 - 割り当てられる帯域が短い間期で変動するアプリケーションでは、帯域を切り替えることが困難
 - 拡張 1: レート制御の高速化
 - パケットロスが発生しなかった時に 2 倍のビットレートを目標に上昇
 - 拡張 2: 上限ビットレートの設定
 - レートの急激な上昇を防ぐために、車両位置予測結果に応じた上限ビットレートを設定

10

実験システム

11

- ミニチュア車両を用いた車両走行システム
 - 現実世界の 1/30 スケールに調節
 - 車両は時速 2 km で等速直線運動
 - 2 台のカメラの設置
 - カメラ 1: [5.6 8.3] → 10 ~ 15 秒後に車両通過
 - カメラ 2: [11.1 13.9] → 20 ~ 25 秒後に車両通過

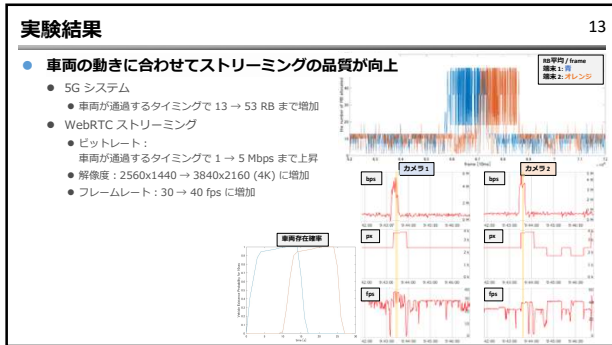
11

実験動画

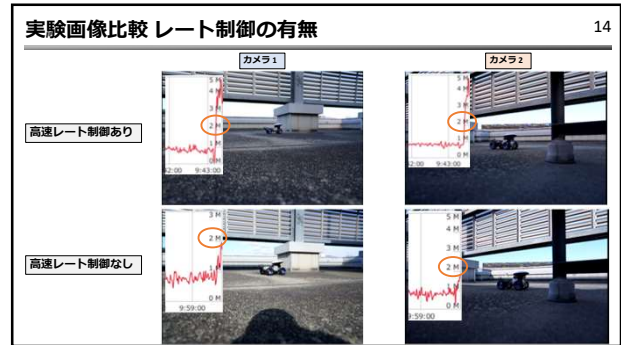
12

- 車両の動きに合わせてビットレートが増減
 - カメラ1
 - カメラ2

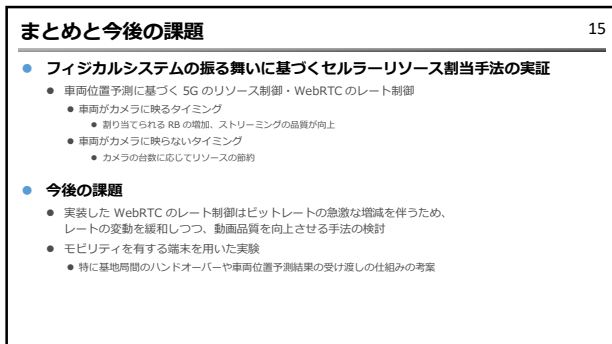
12



13



14



15