

## 科学研究費助成事業 研究成果報告書

令和 4 年 6 月 27 日現在

機関番号：32641

研究種目：若手研究

研究期間：2020～2021

課題番号：20K19790

研究課題名（和文）コネクテッドカーの安全な自動運転制御を実現するネットワーク制御方式の研究

研究課題名（英文）A Network Management Protocol for Connected Cars

研究代表者

吉田 雅裕（Yoshida, Masahiro）

中央大学・国際情報学部・准教授

研究者番号：60785913

交付決定額（研究期間全体）：（直接経費） 3,200,000円

研究成果の概要（和文）：本論文では、自律型モビリティのCANデータを対象に、高頻度に発生するショートパケットの効率的な圧縮方式を提案する。提案方式は、自律型モビリティからエッジサーバまでの無線アクセス網区間をパケットキャッシング（RE）による圧縮を行い、エッジサーバからクラウドまでのインターネット区間をパケットコーディング（gzip）による圧縮を行うという、二段階の圧縮を組み合わせた方式である。公道で実際の自動車を用いた評価実験を行った結果、提案方式は、毎秒1,000メッセージ発生するショートパケットを、リアルタイムに84%圧縮できることを示す。

研究成果の学術的意義や社会的意義

本研究は、コストの安い既存の携帯電話網を自動運転に活用するという点で研究成果の実現性が高く、コネクテッドカーが高速移動する際の携帯電話網の通信品質の安定化、高信頼化に役立つ通信プロトコルを実現するという点で社会への波及効果が期待できる。また、本研究は、先進国だけでなく、携帯電話網の通信品質が不安定な発展途上国において自動運転の早期実用化につながるため、社会的に大きな意味を持つ。

研究成果の概要（英文）：Connected cars generate a huge amount of Internet of Things (IoT) sensor information called Controller Area Network (CAN) data. Although each CAN data packet is very small, a connected car generates thousands of CAN data packets per second. Therefore, real-time CAN data collection from connected cars to a cloud system is one of the most challenging problems in the current IoT. In this paper, we propose an Edge computing-enhanced network Redundancy Elimination service (EdgeRE) for CAN data collection. In developing EdgeRE, we design a CAN data compression architecture that combines in-vehicle computers, edge datacenters, and a public cloud system. EdgeRE includes the idea of a hierarchical data compression and dynamic data buffering at edge datacenters for real-time CAN data collection. Across a wide range of field tests with connected cars and an edge computing testbed, we show that the EdgeRE reduces the bandwidth usage by 88% and the number of packets by 99%.

研究分野：コンピュータネットワーク

キーワード：自動運転車

## 1. 研究開始当初の背景

LTE や 5G などのモバイル網は、今後の IoT を支える重要な通信インフラの一つである。また、人工知能の進化により自動運転車は非常に注目されているが、自動運転車に IoT の技術を適用したコネクテッドカーは今後の主流になっていくと言われている。コネクテッドカーの通信には LTE や 5G などのモバイル網の利用が想定されているが、コネクテッドカーなどの移動体が移動する際に、モバイル網ではモバイル基地局の切り替え(以降、ハンドオフ)が発生する。ハンドオフの発生頻度は、移動体の移動速度が速い場合や、モバイル基地局が密集している都市部などで高頻度に発生する。図.1 は、東京都内の自動車道路を移動した際の NTT ドコモのモバイル端末のハンドオフの発生回数を積み上げたグラフである。Android の TelephonyManager API を用いて、モバイル端末のハンドオフの発生回数を取得した。東京都内の一般道路を 60km/h 程度で走行する自動車は、人間が徒歩で移動する時よりも高い頻度でハンドオフが発生する。また、東京都内の高速道路を走行する場合は、80~100km/h の速度で自動車移動するため、一般道路よりも高速道路のほうがハンドオフの発生頻度は高くなる。図.2 は、東京都内の自動車道路を移動する際に、NTT ドコモのモバイル端末から AWS の東京リージョンに生成した VM に対して、車載カメラで撮影した静止画像を 1 秒ごとに送信した際の往復伝送遅延時間を示している。一般道路を 60km/h 程度で走行する時よりも、高速道路を 80~100km/h の速度で走行する時のほうが、往復伝送遅延時間が増加していることが分かる。また、最悪のケースでは通常時の 80 倍近くの往復伝送遅延時間を生じることもある。エッジコンピューティングとクラウドコンピューティングを比較した場合、通常はエッジコンピューティングのほうが往復伝送遅延時間は短くなる傾向がある。しかし、モバイル網を高速移動する場合は、エッジコンピューティングとクラウドコンピューティングの違いは関係なく、自動車が高速移動した場合にどちらも同じように往復伝送遅延時間が悪化してしまう。これは、エッジコンピューティングはクラウドコンピューティングよりも有線ネットワーク区間の遅延時間を短くする効果があるのに対し、無線ネットワーク区間の遅延時間を短くする効果に乏しいからである。コネクテッドカーのように高速移動する IoT 端末は、モバイル網のハンドオフの影響を受けやすいため、ハンドオフが頻発する区間の往復伝送遅延を短くすることが自動運転車の早期実現に非常に有用となる。これらの理由から、本研究では、ハンドオフ発生時の通信性能を向上する通信方式の検討と実装を行う。

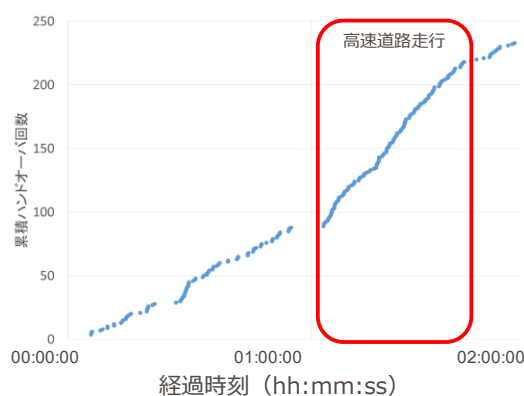


図.1 ハンドオフの発生頻度

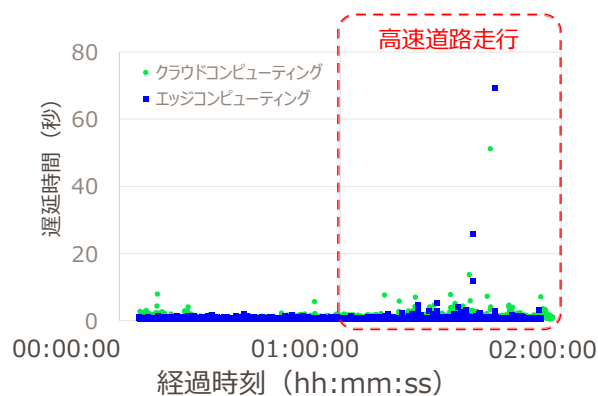


図.2 サービスの応答遅延時間

## 2. 研究の目的

本研究では、高速道路を 80km/h 以上で走行するコネクテッドカーを対象に、シームレスな携帯電話基地局のエリア切り替えを実現する通信プロトコルの確立を目的とする。コネクテッドカーの通信方式には、携帯電話網だけでなく、WiFi、DSRC、ミリ波などの様々な通信方式が存在する。WiFi、DSRC、ミリ波などの通信方式は、道路に新しい通信設備を設置する必要があるが、携帯電話網の場合は既設の通信設備を利用することが可能である。本研究は、コストの安い既存の携帯電話網を自動運転に活用するという点で研究成果の実現性が高く、コネクテッドカーが高速移動する際の携帯電話網の通信品質の安定化、高信頼化に役立つ通信プロトコルを実現するという点で社会への波及効果が期待できる。

本研究は、以下の 2 つの創造性を有する。1 つ目の創造性は、通信事業者ごとに携帯電話基地局の設置場所が異なるという性質を利用し、複数の通信事業者のネットワーク帯域を合成させるという工夫である。コネクテッドカーが高速移動する際に、どの通信事業者のネットワーク帯域を利用するかによって、携帯電話基地局のハンドオフのタイミングは異なる。複数の通信事業

者のネットワーク帯域の中から、ハンドオフが発生していない通信事業者のネットワーク帯域を利用することで、高速移動時にも安定した通信品質を提供できる通信プロトコルを実現する。2つ目の創造性は、携帯電話網の電波強度やコネクテッドカーの位置情報からハンドオフのタイミングを予測する工夫である。複数の通信事業者のネットワーク帯域を無駄遣いしないようにするために、ハンドオフが発生しない区間では単一の通信事業者のネットワーク帯域を利用し、ハンドオフが発生する区間でのみ複数の通信事業者のネットワーク帯域を組み合わせることで、ネットワーク利用効率の良い通信プロトコルを実現する。本研究は、先進国だけでなく、携帯電話網の通信品質が不安定な発展途上国において自動運転の早期実用化につながるため、社会的に大きな意味を持つ。

### 3. 研究の方法

本研究では、Linux カーネルに実装された MPTCP と呼ばれる通信プロトコルの改良を行い、80km/h 以上で高速道路を走行する場合でも、シームレスな携帯電話基地局のエリア切り替えを実現できる通信プロトコルを実装する。実際のコネクテッドカーを用いて高速道路で評価実験を繰り返しながら、高速移動時の通信遅延時間を 200%以上高速化する技術を確認することを目指す。

### 4. 研究成果

#### ハンドオフ発生時の通信品質の実験評価

80km/h 以上の速度で移動する移動体を用いてモバイル網のハンドオフを高頻度に発生させ、IoT 端末からクラウドまでのデータ転送の通信品質がどのように変化するかを評価する。図. 3 に実験環境の概要を示す。ノート PC とスマートフォンを USB テザリングによって接続する。ノート PC では iperf が動作しており、1,500Byte のパケットサイズの通信パケットを生成しながらクラウドに送信する。スマートフォンはモバイル回線に接続しており、ノート PC が生成したパケットをクラウドに送信する役割を持つ。モバイル回線には NTT ドコモの LTE 回線を用いた。また、スマートフォンには Android の TelephonyManager API を用いた計測用アプリを常時動作させており、ハンドオフの発生タイミング、電波強度、スループット、往復伝送遅延時間 (RTT)、パケットロス、端末位置情報 (GPS 情報) をミリ秒単位で記録することができる。ハンドオフを高頻度に発生させるために、実験環境一式を所持した状態で JR 中央線の中央特快に乗り、三鷹駅から中野駅までの区間で通信実験を実施した。



図. 3 実験環境

図. 4 に、ハンドオフ発生時の通信品質評価の実験結果を示す。緑色の縦線はハンドオフの発生タイミングを表しており、緑色の線の前後において、スマートフォンは異なるモバイル基地局に接続されていることを示している。左上のグラフはスループット、右上のグラフはRTT、左下のグラフはパケットロス、右下のグラフは電波強度をそれぞれ表している。ハンドオフの発生タイミングの前後において、特に問題なく安定的に通信が行われることもあるが、ハンドオフの発生と同時にパケットロスの値が異常に高くなることもある。TCP で通信を行う場合はパケットの再送が繰り返されるため、スマートフォンにおいて送信予定のパケットが滞留し、スループットの低下や往復伝送遅延の増加などが引き起こされてしまう。コネクテッドカーが自動運転を行う際は、非常にリアルタイム性の高い通信が求められるため、ハンドオフの発生による通信品質の劣化をなるべく起こさないような通信方式が必要となる。

## ハンドオフの発生タイミング

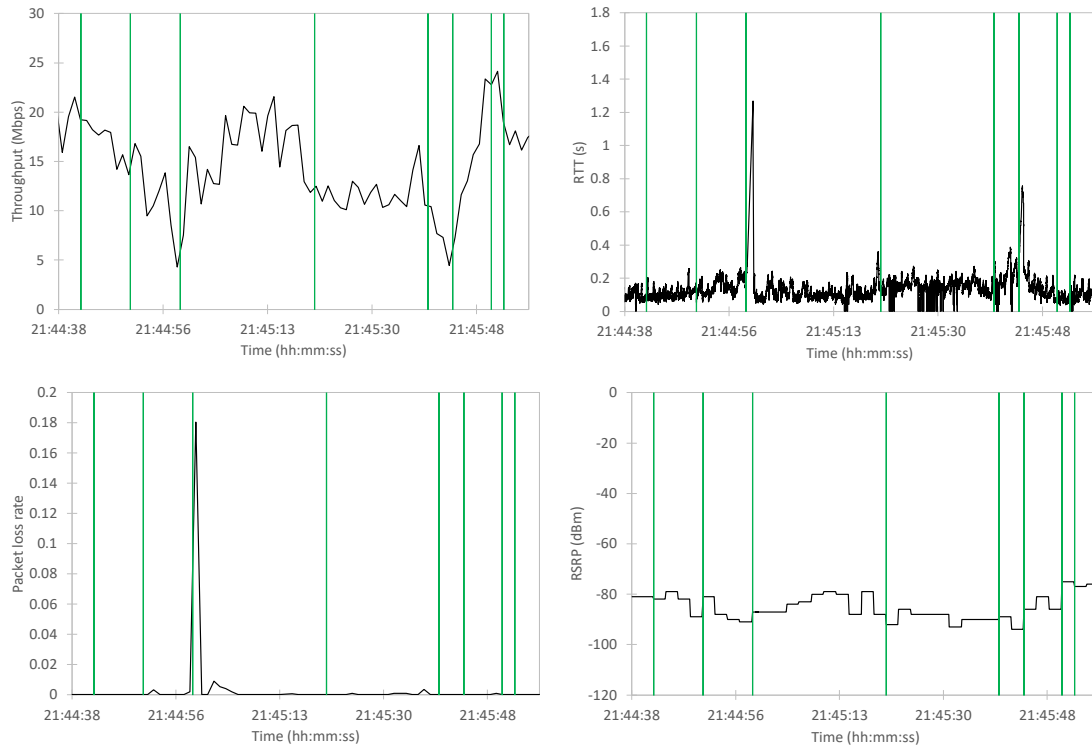


図. 4 ハンドオフ発生時の通信品質

### 提案方式

4章、および、5章の結果から、異なる世代のモバイル網を複数用いるか、異なる通信事業者のモバイル網を複数用いることで、モバイル網のハンドオフが同時に発生する可能性を非常に小さくすることができることが分かる。そこで本研究では、異なる世代のモバイル網や異なる通信事業者のモバイル網を合成することで、ハンドオフによる通信性能の低下を回避することができ、自動運転車などの高速移動にも耐えられるモバイル網のリアルタイム通信方式を提案する。複数の通信網を合成する通信プロトコルとしては、近年のIoT分野で注目を集めているMPTCPの活用が有効であると考えられる。しかし、Liらの論文\*では、既存のMPTCPはモバイル網の通信状況が良い時は通信性能を向上できるが、通信状況が悪い時はTCPよりも通信性能が著しく悪くなるという問題点が指摘されている。そのため、既存のMPTCPを単純に適用するだけでは不十分であり、有線網よりも不安定なモバイル網において通信のリアルタイム性を向上することができるようにMPTCPを改良する必要がある。また、複数のモバイル網を同時に利用する際に、同じパケットを複数のモバイル網に同時に流し続けることは冗長的であり、モバイル網の通信帯域に大きな負荷が発生するため、ハンドオフが高確率で発生する時に限定して冗長パケットを生成する通信方式を検討する必要がある。

\*L. Li et al., “A Measurement Study on Multi-path TCP with Multiple Cellular Carriers on High Speed Rails,” in ACM Sigcomm’18, 2018.

### 実装

実装した提案方式の概要は以下の通りである。LinuxカーネルのMPTCP実装に対して、新しいPacket Schedulerのプラグインを実装した (<https://www.multipath-tcp.org/>)。提案方式は、モバイル網のハンドオフの発生タイミングに応じて、複数のNICにどのようにパケットを流すかを制御することができる。ハンドオフが発生するタイミングでは、複数のNICに対して同じパケットを冗長的に送信する。ハンドオフが発生しないタイミングでは、複数のNICのうち、RTTが最も短いNICのみにパケットを送信する。

Linuxカーネルは、モバイル網のハンドオフなどのユーザスペースで起きうる事象を把握することができない。そこで、ユーザスペースの情報をLinuxカーネルに伝達する必要がある。Linuxカーネルのユーザランドとカーネルランドの通信には、Linuxのプロセス間通信のために用意されたNetlinkインタフェースの仕組みを利用した。

モバイル網のハンドオフの発生状況や、RTT、ジッターなどの各種の通信性能の計測は、AndroidのConnectivityManagerやNetworkInfoなどのAPIを利用した。異なる通信事業者のSIMを挿



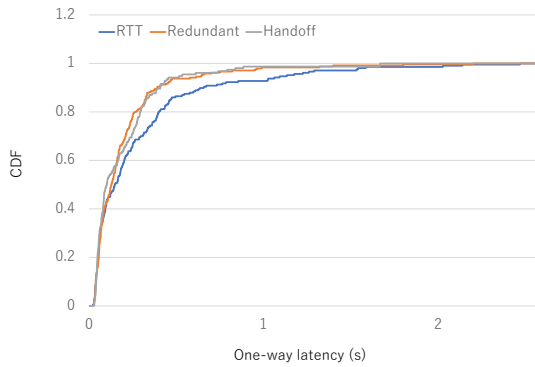


図. 5 通信遅延時間

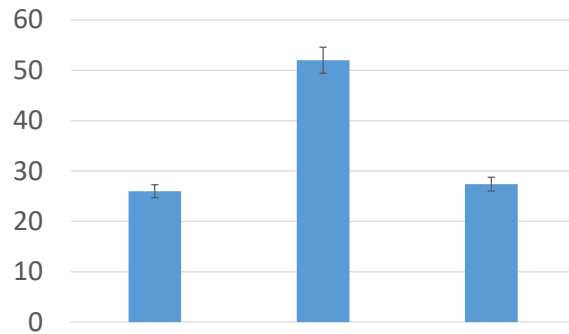


図. 6 平均トラフィック量

入したスマートフォンを3台用意し、それぞれのネットワーク状況を計測しながら、トラフィック制御に必要な情報をLinuxカーネルに伝達する。モバイル網の情報を受け取ったPacketSchedulerは、送信待ちの packets を複数NICで冗長的に送信するか、最良のNICを1つ選んで送信するかを判断しながら、ハンドオフ発生による伝送遅延の発生を抑えるためのパケットスケジューリングを行う。

### 評価結果

時速80km以上の鉄道に乗りながら、テストパケットがモバイル端末からクラウドに到達するまでの通信遅延時間と、複数のNICから生成された平均トラフィック量を5分間計測した。通信遅延時間の計測結果を図. 5に示す。評価実験で用いたパケットスケジューラは、MPTCPに標準で搭載されているRTT、Redundantと、提案方式のHandoffの3種類を用いた。RTTは、複数のNICの中で最も通信遅延時間が低いNICを選んでパケットを送信する。Redundantは、常に複数のNICに対して同じパケットを複製しながら送信する。Handoffは、モバイル網のハンドオフが発生するときは複数のNICに対して同じパケットを複製して送信し、ハンドオフが発生しないときは通信遅延時間が最も低いNICにパケットを送信する。図. 5の通信遅延時間のCDFを確認すると、RTTよりもRedundantのほうが通信遅延時間は小さいことがわかる。モバイル網でハンドオフが発生すると、一時的に通信遅延時間が増加する。そのため、RTTの場合はハンドオフの前後で通信遅延時間が増加してしまう。一方、Redundantの場合は、片方のNICでハンドオフが発生しても、もう片方のハンドオフが発生していないNICに同じパケットを複製して送信しているため、ハンドオフによる通信遅延時間の増加を抑えることができる。提案方式のHandoffは、通信遅延時間が増加するハンドオフのタイミングに複数NICへのパケットの複製を行うことで、Redundantとほぼ同等の通信遅延時間に抑えることができている。

図. 6に、3種類のパケットスケジューラが生成した平均トラフィック量を示す。Redundantは複数のNICに対して、常に同じパケットを複製して送信するため、複数のNICから生成される平均トラフィック量が大きくなっている。一方、RTTは基本的に1枚のNICからパケットを送信するため、生成される平均トラフィック量はRedundantの半分となっている。また、提案方式のHandoffは、パケットを複製するタイミングを限定することで、平均トラフィック量をRTTと同等程度まで抑えることができている。ちなみに、HandoffよりもRTTのほうが平均トラフィックが大きくなっている理由は、現在のRTTのパケットスケジューラの実装では、受信側のバッファがいっぱいになるまでは、他の余裕があるNICも使ってパケットを送信しようとするからである。なお、RTTではパケットの複製は行われず、複数のNICに異なるパケットを送信する。

### まとめ

本研究では、LTEや5Gなどのモバイル網において、ハンドオフの発生時に通信性能が劣化するという課題に対し、複数のモバイル網を合成するというアプローチにより、ハンドオフの発生時の通信性能を向上する通信方式の検討と実験評価を実施した。そして、LinuxカーネルのMPTCPスタックに対して提案方式の実装を行い、実際のモバイル網を用いた実験評価を行った。今後は、提案方式の実装の高度化と、さまざまな条件下での評価実験を進める予定である。

## 5. 主な発表論文等

〔雑誌論文〕 計4件（うち査読付論文 1件/うち国際共著 1件/うちオープンアクセス 1件）

1. 著者名 K. Kanai, M. Yoshida, S. Kawano, S. Mizuno, and A. Nakao	4. 巻 45
2. 論文標題 Report on NS English Session at 2021 IEICE General Conference - BS-7: AI technologies and their applications for future network systems and services -	5. 発行年 2021年
3. 雑誌名 IEICE Communication Society Global Newsletter	6. 最初と最後の頁 1-4
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 A. Nakao, T. Oishi, S. Mizuno, S. Kawano and M. Yoshida	4. 巻 45
2. 論文標題 Report on the 2020 NS English Session Awards and Award Ceremony	5. 発行年 2021年
3. 雑誌名 IEICE Communication Society Global Newsletter	6. 最初と最後の頁 1-4
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 A. Nakao, T. Oishi, Y. Okazaki, S. Mizuno, M. Yoshida and Y. Tanigawa	4. 巻 44
2. 論文標題 Annual Report of Technical Committee on Network Systems	5. 発行年 2021年
3. 雑誌名 IEICE Communication Society Global Newsletter	6. 最初と最後の頁 1-4
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 M. Yoshida, K. Mori, T. Inoue, H. Tanaka	4. 巻 1
2. 論文標題 EdgeRE: An Edge Computing-enhanced Network Redundancy Elimination Service for Connected Cars	5. 発行年 2021年
3. 雑誌名 Proceedings of IEEE IoTNAT2021	6. 最初と最後の頁 1-8
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

〔学会発表〕 計2件（うち招待講演 2件 / うち国際学会 0件）

1. 発表者名 吉田雅裕, 河野伸也, 水野士郎
2. 発表標題 2020年度ネットワークシステム研究会における研究動向報告
3. 学会等名 電子情報通信学会NS研究会（招待講演）
4. 発表年 2021年

1. 発表者名 吉田雅裕, 三原孝太郎, 池邊隆
2. 発表標題 2021年度ネットワークシステム研究会における研究動向報告
3. 学会等名 電子情報通信学会NS研究会（招待講演）
4. 発表年 2022年

〔図書〕 計1件

1. 著者名 岡嶋裕史・吉田雅裕	4. 発行年 2021年
2. 出版社 技術評論社	5. 総ページ数 240
3. 書名 はじめてのAIリテラシー	

〔出願〕 計5件

産業財産権の名称 COMMUNICATION RANGE CONTROL DEVICE, METHOD, AND PROGRAM	発明者 Masahiro Yoshida et al.	権利者 同左
産業財産権の種類、番号 特許、PCT/JP2019/039809	出願年 2021年	国内・外国の別 外国
産業財産権の名称 DATA COMPRESSION APPARATUS, DATA COMPRESSION METHOD, AND PROGRAM	発明者 Masahiro Yoshida et al.	権利者 同左
産業財産権の種類、番号 特許、PCT/JP2019/049407	出願年 2021年	国内・外国の別 外国
産業財産権の名称 DATA COMPRESSION TRANSMISSION SYSTEM, INTERMEDIATE SERVER, METHOD, AND PROGRAM	発明者 Masahiro Yoshida et al.	権利者 同左
産業財産権の種類、番号 特許、PCT/JP2019/050620	出願年 2021年	国内・外国の別 外国

産業財産権の名称 DISTRIBUTED PROCESSING SUPPORT APPARATUS, DISTRIBUTED PROCESSING SUPPORT METHOD, AND PROGRAM	発明者 Masahiro Yoshida et al.	権利者 同左
産業財産権の種類、番号 特許、PCT/JP2019/043067	出願年 2021年	国内・外国の別 外国

産業財産権の名称 NETWORK MANAGEMENT SYSTEM, MANAGEMENT DEVICE, RELAY DEVICE, METHOD, AND PROGRAM	発明者 Masahiro Yoshida et al.	権利者 同左
産業財産権の種類、番号 特許、PCT/JP2019/022719	出願年 2021年	国内・外国の別 外国

〔取得〕 計0件

〔その他〕

-

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------