

協調走行に向けた通信プロトコルの提案に関する研究

情報科学科 周 靖凱

指導教員：田 学軍

1 はじめに

近年、無線ランはよく利用しており、携帯電話のみならず、家電製品や、自動車など、様々な物がすべて、無線ランを利用することになった。特に、自動車の普及により、有人運転から半無人運転、さらに無人運転に変わっていく。将来、すべての自動車は無人運転をなるに違いない。本研究では、無人運転になった後に、自動車は電車のように次から次へ、走っていくという隊列走行になっておる。そのために、無線ランが必要であるが、普通な無線ランは利用できる帯域が限られており、通信遅延も大きい。無線通信 Mac 層プロトコルを改善する必要があり、遅延を減少するためには、新たな Mac 層プロトコルを書くようになり、性能を改善し、評価を行う。

2 IEEE 802.11 と問題点

IEEE 802.11 の MAC プロトコルは、分布式協調機能を定義している (DCF、Function と点協調機能 (PCF、Point Cordining Function) の 2 つのアクセスメカニズム、その中で DCF は競争に基づくアクセス方法で、すべてのノード競争メディア、PCF は競争していない、ノードは特定の時間でメディアを単独で使用することができる。DCF は、基本的なアクセスプロトコルであり、PCF はオプション機能である。今回使用しておるのは IEEE 802.11 DCF のほうである。問題点として、802.11 プロトコルはそのままに使用すると、遅延が発生し、発信しようとしてでも、競争で、発信権利を取るの、発信したい車両は発信権利を取り遅い。

3 提案手法 OTP (Order Transmit Protocol)

OTP (Order Transmit Protocol) 順番送信プロトコルである。その OTP は通信順番があり、車群として OneHop の範囲で、最初にその Hop に入った車両の地理関係は優先であり、その車の発信権利も高くことを決める。次に入った車の優先順位は下げて続けて行く。受信の面に考えると隣 Hop との干渉があるので、2 Hop 外のノードは同時発信することができる。下の図 1 のようになる。

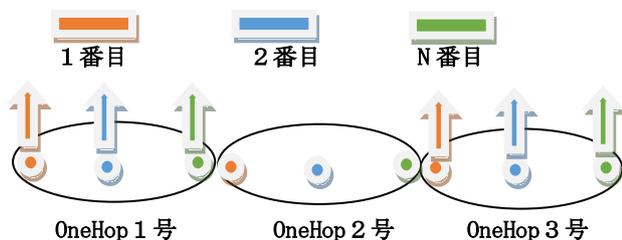


図 1：同時発信

方法として、以下に OTP の簡単なアルゴリズムを示す。

STEP1: 遅延時間 DIFS の以上になった場合は 802.11 から OTP に切り替え。

STEP2: OneHop 内で一回リングでの送信完了後に先ノードの再送信。

STEP3: 送信途中でノードの加入。

STEP4: 送信途中でノードの脱退。

IEEE 802.11 は順番なく、一回送信した後、発信権を競争する必要があることで、送信時間は大分かかるが、OTP では、順番が決めており、最初に入ったノードの発信後に、2 番目のノードをすぐ発信できるようになり 2 Hop 外のノードは同時発信できるので、時間節約できる。

4 シミュレーション環境 (NS-3) の整備と Debug

検査必要な技術

シミュレーションするために、パソコン環境は普通から NS-3(network simulator-3)に変わる必要がある。NS-3 とは NS-2 の拡張ではなく、新しいシミュレーターであり、両方とも C++によって作成されたが、NS-3 は NS-2 の API をサポートしていないが、全く新しいシミュレーターである。

プログラムを書く時に、よく Debug が出てくる。その Debug を検出するためには、そのソフトを利用しないと検出できない。それを改善するためには Eclipse というソフトで Debug 断点調節する。下の図 2 のようになる。

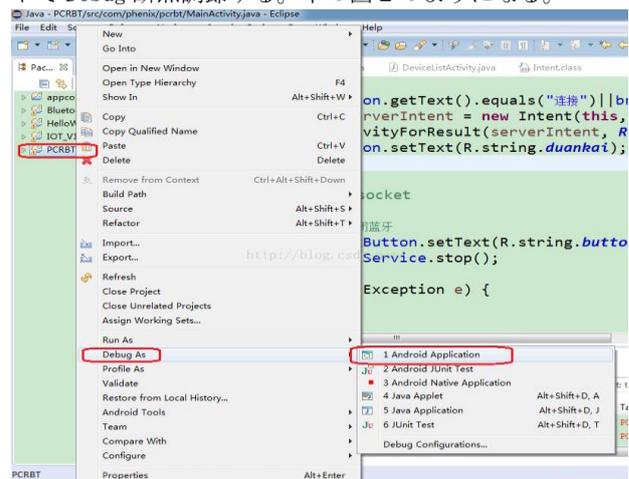


図 2：Eclipse により、Debug 調整

5 まとめ

OTP はノードの送信率が高くなり、遅延を減少するというプロトコルの提案。Eclipse を利用して、NS-3 環境でもっとより良い Debug 調整ができて、順調的に、完成しやすいプログラム書けるようになる。

参考文献

[1] [https://ja.wikipedia.org/wiki/Ns_\(シミュレータ\)#ns-3](https://ja.wikipedia.org/wiki/Ns_(シミュレータ)#ns-3) の仕様概要

[2] 児玉 敦司, ロボットカーを用いた隊列走行アルゴリズムの検討, 愛知県立大学 情報科学学部 情報科学学科 卒業論文, 2014.

[3] 河辺 泰貴, マルチホップ環境における IEEE802.11 の性能改善に関する研究, 愛知県立大学 情報科学学部 情報科学学科 卒業論文, 2016.