

# 無線ネットワークのテストに向けた無線トラフィックエミュレータの設計 Design of Wireless Traffic Emulator Toward Testing of Wireless Networks

玉井森彦<sup>†</sup>  
Morihiro Tamai

長谷川晃朗<sup>†</sup>  
Akio Hasegawa

横山浩之<sup>†</sup>  
Hiroyuki Yokoyama

## 1. はじめに

工場、病院、ショッピングモール等の屋内環境において機器や環境の制御やモニタリングなどを目的に、IoT (Internet of Things) デバイスの普及が進んでいる [1]. IoT デバイス間の通信には Wi-Fi 等の無線通信の利用が想定されることが多いが、特に屋内では電波の減衰や反射の影響が大きく、また予測も難しい。そのため、IoT デバイス上で動作するアプリケーションの要求性能を達成しつつデバイスを安定的に運用するためには、デバイスのある程度計画的に配置することが重要となる。特に、これら IoT デバイスが設置される環境では、時間経過に伴い IoT デバイスの設置位置の変更や新たな IoT デバイスの追加などの要求が生じるため、そのような環境の変化による通信品質悪化への対応は重要な課題である。例えば、工場において工作機械間で無線通信による制御や監視を行っている場合、生産ラインの増強などで新たな工作機械を導入すると、干渉により既設もしくは新設の工作機械で達成されるべき通信品質の確保が困難となる可能性がある。このような場合、実際に新たな工作機械を導入するより前に、あらかじめ無線品質がどの程度変化するかを簡便な方法で確認し、設置計画に役立てられると便利である。

このような課題の解決に向け、本稿では、無線トラフィックのエミュレータを提案する。本エミュレータは複数の送信ノードから構成され、これらの送信ノードを用いて、対象環境内の IoT デバイスが発生させるトラフィックを模擬したトラフィックを発生させる。対象環境内に新たに IoT デバイスを追加したいという状況を想定すると、IoT デバイスの追加の前に、まず提案エミュレータにより模擬的にトラフィックを発生させ、その影響を観測することで、新たな IoT デバイスの追加により通信品質に許容できない悪化が生じないかどうかを確認できるようにする。これにより、実際に IoT デバイスを追加するより前に無線品質の確認ができ、必要であれば、追加可能な台数の見積もりや設置位置の設計などにも役立てることができる。

以上のようなトラフィックエミュレータの実現に向け、本稿では、各送信ノードがある一定周期でパケットを送信するという単純なトラフィックの模擬について、送信ノードのプロトタイプを実装するとともに、送信タイミングの精度を調査する実験を行った結果を述べる。

## 2. 提案システム

提案するトラフィックエミュレータの全体像を図 1 に示す。システムは複数台の送信ノードと、制御ノードから構成される。各送信ノードと制御ノード間は有

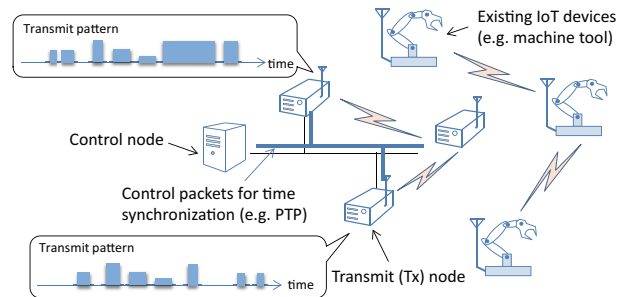


図 1: 提案システムの全体像。



図 2: 送信ノードのプロトタイプの外観。

線 (または無線) で接続され、PTP (Precision Time Protocol) などの時刻同期プロトコルを用いて制御ノードをマスタとして時刻同期を行う。送信ノードは、対象となる無線環境において新設を計画されている IoT デバイスの代わりに一時的に設置される。制御ノードは、新設される IoT デバイスが発生させるトラフィックを模擬した送信パターンを各送信ノードに通知し、各送信ノードは通知された送信パターンに従って無線インタフェースからパケットを送出する。送信パターンの表現方法として、単純な例としては、送信開始時刻、送信周期、パケットサイズの組で表現し、CBR (Constant Bitrate) トラフィック<sup>‡</sup>を生成することが考えられる。また、単位時間当たりのパケットの生成確率を表現する確率分布を与える方法なども考えられる。送信パターンの表現方法は、IoT デバイスが発生させるトラフィックをいかにモデル化するか、という問題と合わせて検討することが重要と考えられる。これについては今後の課題とする。

## 3. プロトタイプと実験

送信ノードのプロトタイプの実装を行った。プロトタイプの外観を図 2 に示す。送信ノードは、様々な環境下での利用における耐久性の観点から、外装をハードケース (エクスペローラ製 IEX-3818) とし、その内部に Intel Core i3-6100TE を搭載したファンレスコンピュータ (Neusys Technology 製 Nuvo-5100VTC) と、バッ

<sup>†</sup>株式会社国際電気通信基礎技術研究所, ATR

<sup>‡</sup>[https://en.wikipedia.org/wiki/Constant\\_bitrate](https://en.wikipedia.org/wiki/Constant_bitrate)

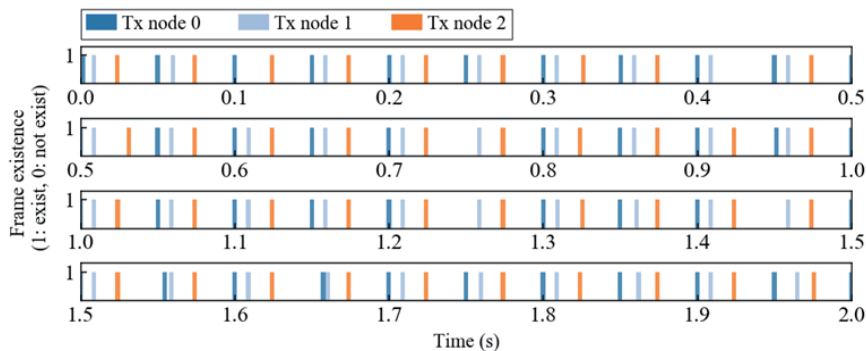


図3: 送信ノードごとに見たパケットの受信時刻.

テリ (50,000 mAh) を収納した. 図2は, ケースの蓋を開けた状態である. 無線 LAN インタフェースとして, サイレックス・テクノロジー製の SX-PCEAN2i を用いた.

開発したプロトタイプについて, パケットの送信タイミングの精度を調査する実験を行った. 具体的には, 三台の送信ノード (送信ノード0, 1, 2と呼ぶ) を用いて, 各ノードから周期的にパケットを送信し, 意図した周期でパケットの送信が行えているかどうかを調査した. 実験では, ATRの社屋内の一室を用いた. なお, 実用に即した状況で実験を行うため, 実験系とは無関係に配置されているアクセスポイントや端末等からのバックグラウンドトラフィックが存在する中で実験を行った. 時刻同期のため, 制御ノード上で `ptpd2`<sup>§</sup> をマスターモードで動作させ, 各送信ノード上で `ptpd2` をスレーブモードで動作させた. 各送信ノードは, 50ミリ秒周期でパケットを1つ送信する. 各パケットのサイズは, IPヘッダに対するペイロード (UDPヘッダ含む) が2000バイトとなるよう設定した. 各パケットは, ブロードキャストアドレス宛に送信される. 各送信ノードのOSは, Linux (debian 3.16.0-4-amd64) であり, パケット送信用のプログラムでは, ソケットを `socket(AF_INET, SOCK_DGRAM, 0)` で生成し, パケットは `sendto` 関数によって送信した. 各送信ノードの送信タイミングを次のように設定した. ノード0の送信開始時刻を  $t_0$  (秒) とすると, ノード1と2の送信開始時刻をそれぞれ  $t_0 + 0.01$ ,  $t_0 + 0.025$  とした. 無線インタフェースのドライバは, `ath9k`<sup>¶</sup> を用いた. 使用したチャンネル番号は11 (2.4GHz帯) であり, 802.11gのアドホック (IBSS) モードでドライバを動作させた. また, 伝送レートは, 6Mbpsに固定した.

送信タイミングを計測するため, `tcpdump` を動作させたノートPCを送信ノードの近くに設置し, パケットキャプチャを行った. キャプチャ結果から, 送信ノードごとのパケットの受信時刻を抽出した. 各パケットの受信時刻を時間軸上にプロットした結果を図3に示す. 図3は, 計測結果全体の内の2秒分を抜き出して4行に分けて (各行が0.5秒分) プロットしたものである. 図より, 各パケットは, 概ね適切なタイミングで受信されていることが分かる. 受信時刻のずれは, 理

表1: 600秒間でのパケット受信間隔の統計量.

送信ノード	データ個数	平均値 s	標準偏差 s
ノード0	10775	0.049995	0.002540
ノード1	10964	0.049995	0.002494
ノード2	10986	0.049999	0.002335

想的な場合の受信時刻に対し, 大きな場合でも10ミリ秒程度以内に収まっている. また, バックグラウンドトラフィックの影響などにより, 一部パケットロスが生じていることが分かる.

次に, パケット受信間隔が適切かどうかをより詳細に見るため, 600秒間パケットキャプチャを行い, 送信ノードごとにパケットの受信間隔の統計量 (平均値と標準偏差) を算出した. なお, 算出に使用するデータの抽出は, パケットロスが存在することも考慮し, 次のような方法で行った. 1つの送信ノードに対し, 連続する2つのパケットの受信に対し, 1つ目のパケットの受信時刻を  $t_1$  とすると, 2つ目のパケットの受信時刻  $t_2$  が,  $t_1 + 0.05$  ( $t_1$  の50ミリ秒後) の  $\pm 25$  ミリ秒以内に収まっている場合,  $t_2 - t_1$  を受信間隔として統計量の算出に用いた. 表1に, 有効な受信間隔のデータ個数と, 統計量の算出結果を示す. 表1より, 受信間隔はほぼ50ミリ秒であり, 標準偏差は3ミリ秒以内に収まっていることから, ほぼ送信周期 (50ミリ秒) でおりで送信が行えていることが分かる.

#### 4. おわりに

本稿では, IoTデバイスの新設などに伴う無線品質への影響を事前にテストすることを目的とした, トラフィックエミュレータを提案した. プロトタイプの実装を行い, 三台の送信ノードを時刻同期した上で, 適切なタイミングで各送信ノードから周期的にパケットを送信可能なことを確認した.

#### 謝辞

本研究は総務省の「電波資源拡大のための研究開発」の「狭空間における周波数稠密利用のための周波数有効利用技術の研究開発」の一環として実施した.

#### 参考文献

- [1] S. Wang, J. Wan, D. Li, C. Zhang: "Implementing Smart Factory of Industrie 4.0: An Outlook," *Intl Journal of Distributed Sensor Networks*, Vol. 2016, Article No. 3159805, 2016.

<sup>§</sup><http://ptpd.sourceforge.net/>

<sup>¶</sup><https://wireless.wiki.kernel.org/en/users/drivers/ath9k>