

ランダム AIFSN 方式を用いたメディアアクセス制御の 無線 LAN 端末への実装

平田龍一[†] 西尾理志[†] 守倉正博[†] 山本高至[†]

[†] 京都大学情報学研究科 〒606-8501 京都市左京区吉田本町

E-mail: [†]info15@imc.cce.i.kyoto-u.ac.jp

あらまし 無線 LAN においてメディアアクセス制御方式として用いられている CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) では、無線 LAN が多数存在している環境においてフレーム衝突が多発しスループットが低下する。この問題を解決するために我々はランダム AIFSN 方式を提案している。ランダム AIFSN 方式では端末ごとにランダムな待ち時間 AIFS を設定することで端末を仮想的にグループ化し、競合する端末の数を減らすことでフレーム衝突を回避する。これまではシミュレーション評価により有効性が示されてきたが、実験による評価はなされていない。本稿では、ランダム AIFSN 方式を廉価な通信機器に実装した。実験により、端末上でランダム AIFSN 方式が動作していることを確認した。更にスループット低下要因の調査を行い、デバイスドライバへの値の書き込みがボトルネックとなることを明らかにした。

キーワード 無線 LAN, メディアアクセス制御, ランダム AIFSN, 実装, Raspberry Pi

Implementation of Random AIFSN Scheme on Off-the-Shelf WLAN Devices

Ryuichi HIRATA[†], Takayuki NISHIO[†], Masahiro MORIKURA[†], and Koji YAMAMOTO[†]

[†] Graduate School of Informatics, Kyoto University Yoshida-Honmachi, Sakyo-Ku, Kyoto, 606-8501 Japan

E-mail: [†]info15@imc.cce.i.kyoto-u.ac.jp

Abstract Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a medium access control mechanism which is employed in wireless local area networks (WLANs). In CSMA/CA, the throughput deteriorates as a number of nodes increases due to increase of frame collisions. To solve this problem, we have proposed the random AIFSN scheme which maintains a small number of contending nodes by virtually dividing nodes into contending and non-contending groups. The performance has been evaluated by computer simulations, but an experimental evaluation is still an open issue. In this paper, we implement the random AIFSN scheme on an off-the-shelf WLAN device. The experimental results confirm the random AIFSN scheme works on the device and show that writing AIFSN value in a hardware driver becomes the bottleneck and decreases the WLAN throughput.

Key words wireless LAN, medium access control, Random AIFSN, densely deployed WLANs, implementation, Raspberry Pi

1. はじめに

あらゆるモノとモノをネットワークでつなぐ Internet of Things (IoT) が注目を集めている。そのアプリケーションの 1 つにスマートハウスやスマートシティがある。空調や照明、ドアロックなど様々な家電をネットワーク越しに制御したり、カメラや温度計、照度センサや各種ウェアラブルセンサなどを多数配置し人やその周囲の環境をセンシングし、得られたビッグ

データからよりよいサービスを提供する試みが始まりつつある。IoT 実現に向け重要となるのが、廉価で汎用性がある無線通信である。その候補として無線 LAN がある。無線 LAN は既に広く家庭に普及しており、さらに Raspberry Pi [1] や Kinoma Create [2] など小型で廉価で IoT 開発向けと期待されるデバイスにも対応している。しかし、IoT では多数の端末を設置する必要があり、無線 LAN 通信が混雑する可能性がある。

オフィスや家庭で利用する据置型の無線 LAN に加え、スマー

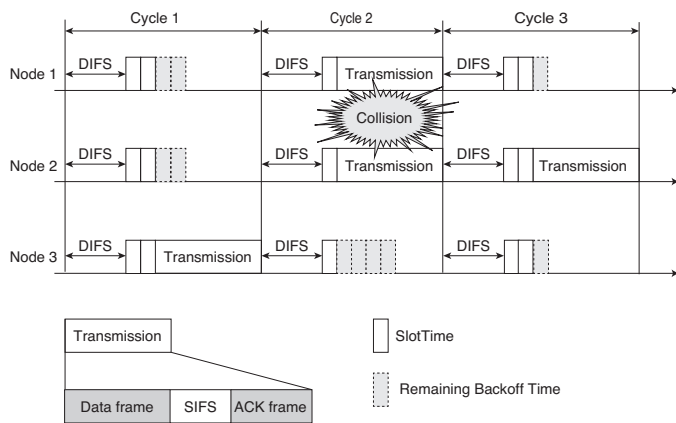


図 1 CSMA/CA の送信手順 .

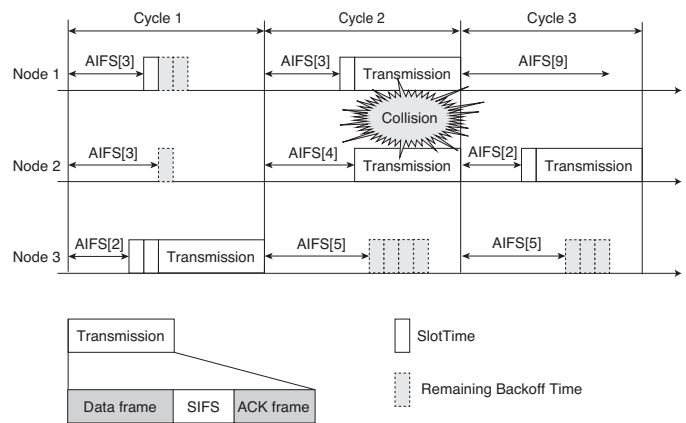


図 2 ランダム AIFSN 方式の送信手順 .

トフォンのテザリング機能やモバイルルータを用いた移動無線 LAN システムの普及により、無線 LAN のチャンネルは不足している。特に 2.4 GHz 帯では干渉なく利用可能なチャンネルが 3 つしかないため、複数の無線 LAN が同一の周波数チャンネルを利用することが避けられなくなっている。現行の無線 LAN 機器はそのほとんどが IEEE 802.11 規格に準拠しており、メディアアクセス制御に CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) が用いられている。CSMA/CA では、各端末が送信前に端末ごとにランダムに定められたバックオフ時間を待つことで、同時送信によるフレームの衝突を回避している。しかし、多数の無線 LAN 端末が同一チャンネルを利用している場合、同じバックオフ時間を選択する確率が上がり、フレームの衝突が多発し、スループットが低下するという問題が発生する。

この問題を解決するために我々は CSMA/CA の拡張としてランダム AIFSN (Arbitration Inter-Frame Space Number) 方式を提案している [3, 4]。ランダム AIFSN 方式では、通常は固定長で設定されている AIFS 期間を端末が送信を行うごとにランダムに決めることで AIFS 期間により端末が仮想的にグループ化され、競合する無線 LAN の数を減らすことができる。我々はこれまでランダム AIFSN 方式の性能についてシミュレーションによってのみ評価してきた。本稿では、ランダム AIFSN 方式を廉価な IoT 開発デバイスである Raspberry Pi に実装する。実験により実装時のスループット低下要因を明らかにし、更にランダム AIFSN 方式の動作確認をする。

2. ランダム AIFSN 方式

2.1 CSMA/CA

IEEE 802.11 規格において用いられている CSMA/CA について述べる。図 1 に CSMA/CA における送信の手順を示す。フレームを送信する前に各端末は DIFS 期間キャリアセンスを行い、チャンネルが使用されている (ビジー) か未使用 (アイドル) かを確認する。DIFS は式 (1) で定められる時間で全端末で同じ長さである。

$$DIFS = SIFS + 2 \times SlotTime \quad (1)$$

もしチャンネルがビジーであれば、端末はチャンネルがアイドルになるまで待ち、チャンネルがアイドルであれば、端末ごとにランダムに割り振られるバックオフカウンタを減らし始める。バックオフカウンタを減らしている端末は、チャンネルがビジーであることを検知すると再びアイドルになるまでバックオフカウンタを減らすのを止める。バックオフカウンタを減らし終えた端末はデータを送信し、そのデータを正常に受信した端末は ACK を返送する。

2.2 ランダム AIFSN 方式

ランダム AIFSN 方式では、各端末はデータを送信する前に CSMA/CA で用いられていた固定の待ち時間 DIFS の代わりに端末ごとにランダムに決められる AIFS 期間を待つ。従来 AIFS は QoS (Quality of Service) 制御に用いられ、その値は IEEE 802.11e で規定されるアクセスカテゴリごとに固定値が使用される。提案手法では、式 (2) に従い AIFS 期間を設定する。

$$AIFS = SIFS + AIFSN \times SlotTime \quad (2)$$

AIFSN は $[2, AIFSN_{max}]$ の範囲から一様ランダムに選択される整数値である。式 (2) で定められる AIFS を $AIFS[AIFSN]$ と表す。例えば、AIFSN が 2 のとき、AIFS は $AIFS[2]$ となり、DIFS と同じ長さである。図 2 にランダム AIFSN 方式における送信の手順を示す。各端末は AIFS 期間キャリアセンスをし、チャンネルがビジーかアイドルかを確認する。AIFS 期間待った端末はバックオフカウンタを減らし始める。AIFSN は端末がデータを送信することに再設定される。

2.3 仮想グループ化

ランダム AIFSN 方式では仮想グループ化により端末は競合端末と非競合端末に分けられる。競合端末はバックオフカウンタを減らすことのできる端末であり、非競合端末はバックオフカウンタを減らすことのできない端末である。

図 2 のサイクル 1 において端末 2 は非競合端末で端末 1 と 3 は競合端末である。このサイクルにおいて最も小さい AIFSN を待つ端末 3 がバックオフカウンタを 2 つ減らしてデータを送信する。端末 2 がチャンネルのビジーを検知してバックオフカウンタを減らせないのに対し、端末 1 はバックオフカウ

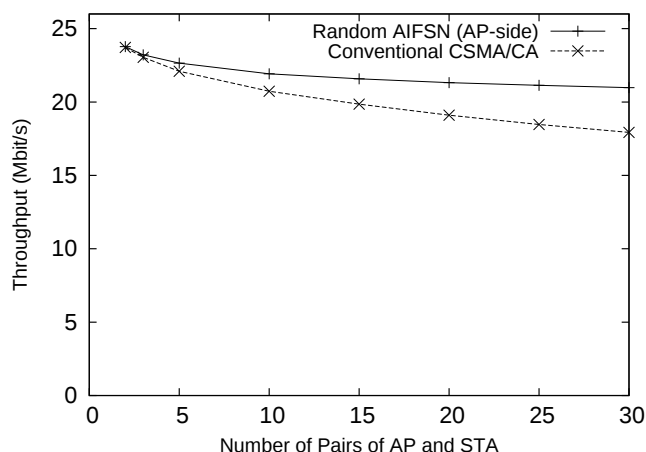


図3 シミュレーション評価によるランダム AIFS 方式のスループット特性。

表1 実験機器。

AP, STA	Raspberry Pi 1 model B/B+
WLAN adapter	Buffalo WLI-UC- GNM2
OS	FreeBSD 10.0
Server	Apple MacBook Pro Retina
Traffic generator	iperf 2.0.5
Wireless LAN	IEEE 802.11g
Wired LAN	Ethernet 100 BASE-T

ンタを1つ減らす。サイクル2においては、端末3は大きい *AIFS* が設定されており、結果的にバックオフカウンタを減らせないため非競合端末となる。このように大きい *AIFS* を設定された端末は結果的に非競合端末となる。そのため、小さい *AIFS* を獲得した端末のみが競合端末となるので、競合端末数は少なくなり衝突率を低減できる。

AP1台とSTA1台のペアが複数存在している環境において各ペアでTCP下り通信を行っている環境におけるシステムスループットをネットワークシミュレータ QualNet [5] を用いて評価した。図3にAP-STAのペア数に対するシステムスループットを示す。ペア数が多い時、従来のCSMA/CAに比べ、ランダム AIFS 方式は高いスループットを維持できた。

3. Raspberry Pi への実装

3.1 実験機器

本実験における実験機器を表1に示す。データサーバは Apple MacBook Pro Retina を用い、有線 LAN で AP に接続した。AP および端末としては Raspberry Pi [1] を使用した。OS は FreeBSD [6] を用い Raspberry Pi 上で動作させ、さらに IEEE 802.11b/g 規格に準拠した無線 LAN アダプタを装着して端末として使用した。AP は、端末と同様に IEEE 802.11b/g 規格準拠の無線 LAN アダプタを装着した Raspberry Pi 上で FreeBSD を動作させ、hostapd によって AP として動作するように設定した。

実装では、FreeBSD 上の Linux カーネルモジュールの1つ

である net80211 [7] に変更を加えた。net80211 は無線 LAN の MAC 機能の一部をソフトウェアで提供する software MAC (SoftMAC) の実装の一種で、様々な Linux kernel で使用されている [8]。net80211 は端末モード、AP モード、アドホックモード、メッシュモード (IEEE 802.11s)、モニターモードなど様々なモードに対応している。net80211 では伝送レート制御や *AIFS* や CW_{min} など各種パラメータの設定、上位レイヤから来たパケットをデバイスドライバに受け渡すなどの機能を担っている。上位レイヤからデバイスドライバへのパケットの受け渡し時には、IEEE 802.11 フォーマットへの変換と各種パラメータ (伝送レート、シーケンス番号、暗号化方式、送信時間など) の設定が行われている。デバイスドライバからのフレームは、復号や上位レイヤのフォーマットへの変換や、受信したフレームがアグリゲーションフレームの場合は Block ACK の送信処理を行う。

また、今回実装に用いた FreeBSD の net80211 では、WiFi-AggressiveMode が使用されている [9]。これは、AP が高い優先度を持つアクセスカテゴリのキューの状態を監視し、帯域に余裕があるようならば、端末に TXOP (Transmission Opportunity) Window 期間の間、SIFS 間隔でデータフレームの連続送信を許可するという機能である。TXOP Window の更新はビーコンやアソシエーションレスポンスで通知される。高い優先度を持つアクセスカテゴリのキューにフレームが溜まり始めたら帯域が混雑していると判断し、TXOP Window を初期値に戻し連続送信を停止させる。

3.2 ランダム AIFS 方式の実装方法

ランダム AIFS 方式の実装方法として、FreeBSD 上の net80211 の ieee80211_vap_transmit 関数の変更がある。ieee80211_vap_transmit 関数は上位レイヤからのパケットをハードウェアドライバに受け渡す関数である。上位レイヤで生成されたパケットが必ず通る関数であり、変更も比較的容易である。

ieee80211_vap_transmit 関数において、*AIFS* の更新を行うことでランダム AIFS 方式が実装できる。具体的にはランダム AIFS 方式が有効か無効かを知らせるフラグのチェックを行い、有効であれば2から $AIFS_{max}$ の整数値の乱数を発生させ *AIFS* として設定し、ドライバに値を書き込む処理を行う。しかし、トラフィック量が多い時、この書込時間がオーバーヘッドになり、スループットを低下させる可能性がある。

[3,4] で提案しているランダム AIFS 方式では、フレーム送信を行う度に *AIFS* を更新しており、一度大きな *AIFS* を獲得した端末は長くフレーム送信を行えず、*AIFS* の分布は一樣ではなくなる。一方、本実装では、上位レイヤから IEEE 802.11 MAC 層にパケットが到達する度に *AIFS* が更新される。そのため、端末が送信時に使用する *AIFS* は2から $AIFS_{max}$ の間で一樣分布する。*AIFS* の分布は異なるが、2.3 で述べた仮想グループ化の効果は現れるため、衝突率低減が見込まれる。また、*AIFS* が送信完了によらずランダムに変わるため、大きな *AIFS* を獲得してしばらく送信できなくなる端末が生じず、端末間の公平性が向上することが期待される。

表 2 実験諸元.

WLAN Band	2.4 GHz
WLAN Channel	1 CH
PHY rate	12 Mbit/s or 54 Mbit/s
SIFS	10 μ s
Slottime	20 μ s
WiFiAggressiveMode	enable
CW _{min}	15
CW _{max}	1023

3.3 実験評価に向けた変更

検証を簡単にするため、伝送レート固定で実験を行う必要がある。実験システムにおいて `ifconfig` コマンドから伝送レートを固定設定にしたところ、表示上は指定した伝送レートになるが、実際はより低い伝送レートで通信するという問題が生じた。本稿ではこの問題を回避するため、`net80211` 上で使用される伝送レートセットを特定の値に限定することで、AP と端末が固定レートで通信するよう変更した。

4. 実験評価

4.1 実験シナリオ

まず、実験機器や環境の影響を調査するため基礎実験を行った。スループット低下要因としては、チャンネル干渉、端末性能、ドライバへの AIFSN の書込が考えられる。チャンネル干渉及び端末性能の影響の評価には通常の CSMA/CA 準拠の端末を用いた。チャンネル干渉の影響の調査では、実験場所として研究室及び比較的干渉波が少ない地下教室で測定を行った。端末性能の影響の調査では、AP には Raspberry Pi を用い、送信端末として Raspberry Pi を用いた場合と MacBook Pro を用いた場合を比較した。最後にドライバへの AIFSN の書込の影響の調査では、通常の CSMA/CA 準拠の端末と `ieee80211_vap_transmit` 関数でドライバへの書込を行う端末とを比較した。伝送レートは 12 Mbit/s と 54 Mbit/s の二通で行った。

次に、ランダム AIFSN 方式の動作確認を行った。ランダム AIFSN 方式を実装した端末の通信中に `ifconfig` コマンドを実行し AIFSN を計測した。通常の端末の場合、AIFSN はアクセスカテゴリごとに固定値で定義されるが、ランダム AIFSN 方式を実装した端末では、通信中は `ifconfig` コマンドを実行する度に値が更新される。

トラヒックは端末から AP または AP にサーバとして有線 LAN で接続された MacBook Pro に対し `iperf` [10] により UDP 通信を行った。スループット測定では `iperf` を用いて各 10 秒間 10 回測定しその平均をとった。実験機器の都合上、`WiFiAggressiveMode` は有効な状態で実験を行った。その他諸元を表 2 に示す。

4.2 実験評価

図 4 及び図 5 に環境、端末、ドライバへの書込を変更した時のスループット及び $CW_{min} = 15$ の場合と TXOP 有効で SIFS 間隔でフレーム送信している場合のスループットの理論値 [11] を示す。伝送レートは 54 Mbit/s とした。スペクトラム

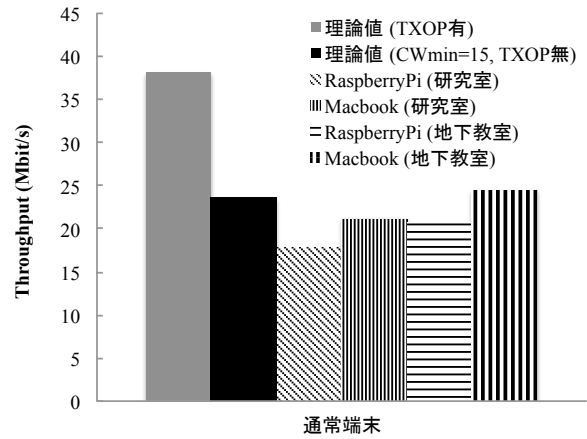


図 4 環境と端末の違いによるスループット性能差。

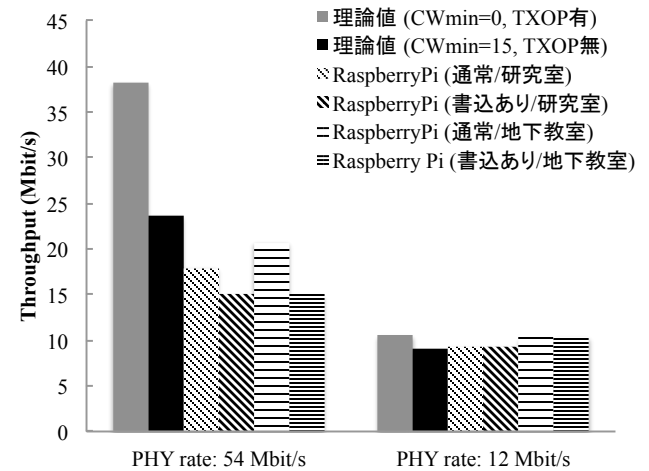


図 5 ドライバへの書込の影響評価。

アナライザにより測定された干渉波は、研究室では実験に使用したチャンネル 1 において最大 -65 dBm、チャンネル 6 と 11 において最大 -60 dBm であった。地下教室では、キャリアセンス閾値以上の干渉波はどのチャンネルにおいても観測されなかった。図 4 より、干渉波の少ない環境では 3 Mbit/s 程度改善し、送信端末を Raspberry Pi から MacBook Pro に変更することでさらに 3.5 Mbit/s 程度改善し、理論値のスループットとほぼ同じ値となった。

図 5 より、伝送レート 12 Mbit/s では通常端末とドライバへの書込を行う端末との間に差異は見られず、ドライバへの書込の影響はなかった。しかし、伝送レート 54 Mbit/s ではドライバへの書込を行った場合、15 Mbit/s 程度でスループットが頭打ちした。ドライバへの書込はパケットが上位レイヤから到着したときに行われるため、ここで遅延が生じると IEEE 802.11 ハードウェアのキューに到着するフレームの量が制限される。書込に 800 μ s 程度かかると仮定すると、1 s あたりにハードウェアのキューに到着するフレーム数は 1250 個程度となり、ペイロードが 1500 B とすると端末が 1 s 間に送信可能なフレームは最大でも 15 Mbit となる。これがスループットが 15 Mbit/s 程度で頭打ちになる原因と考えられる。

最後に、ランダム AIFSN 方式の動作確認を行った。図 6 に端

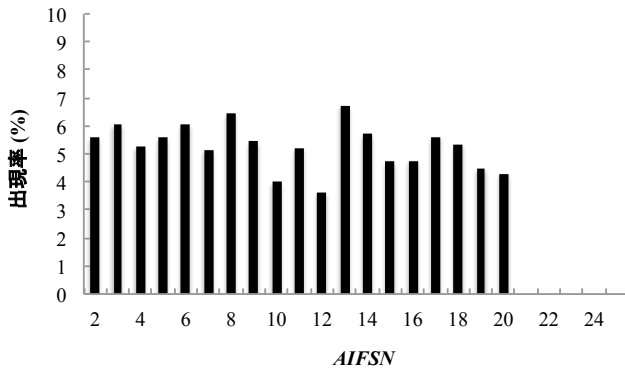


図 6 ランダム AIFSN 方式における AIFSN の出現率 .

末が通信中に ifconfig コマンドを 557 回実行したときの AIFSN の出現率を示す . 2 から 20 の範囲ではどの値も 5.5%程度で出現していることがわかる . また , 設定範囲外である 21 以上は出現しなかった . このことから AIFSN を指定した範囲で一樣ランダムに , かつ , 動的に設定できていることが確認出来た .

5. ま と め

本稿では , ランダム AIFSN 方式を Raspberry Pi に実装し動作確認を行った . 加えて , 本実装でのスループット低下要因を実験的に明らかにした . 実験結果では , 同一チャネル干渉や低い端末性能に加えてハードウェアドライバへの AIFSN の書込がハードウェアキューへのフレーム受け渡しを妨げスループットを低下させる可能性を明らかにした . また , ランダム AIFSN 方式を実装した端末の AIFSN が通信中は任意の範囲の一樣ランダムに設定されていることを実験で確認した . 今後はパケットキャプチャによる詳細な解析と実証実験によりランダム AIFSN 方式の有効性を示す .

文 献

- [1] “Raspberry Pi,”. <https://www.raspberrypi.org/>.
- [2] “Kinoma Create,”. <http://www.kinoma.com/create/about/>.
- [3] K. Ogawa, Y. Sangenya, M. Morikura, K. Yamamoto, and T. Sugihara, “IEEE 802.11ah based M2M networks employing virtual grouping and power saving methods,” Proc. IEEE Vehicular Technology Conference (VTC) 2013, pp.1–5, Las Vegas, USA, September 2013.
- [4] R. Hirata, T. Nishio, M. Morikura, K. Yamamoto, and T. Sugihara, “Cross-layer performance evaluation of random AIFSN scheme in densely deployed WLANs,” Proc. International Conference on Signal Processing and Communication Systems (ICSPCS) 2014, pp.1–6, Gold Coast, QLD, Australia, December 2014.
- [5] Scalable Networks Technologies. “QualNet,”. <http://web.scalable-networks.com/content/qualnet>.
- [6] “FreeBSD,”. <https://www.freebsd.org/ja/>.
- [7] “net80211,”. <http://www.unix.com/man-page/freebsd/4/net80211/>.
- [8] M. Vipin and S. Srikanth, “Analysis of open source drivers for ieee 802.11 wlans,” Proc. International Conference on Wireless Communication and Sensor Computing (ICWCSC) 2010, pp.1–5, Jan 2010.
- [9] “FreeBSD WiFiAggressiveMode,”. <https://wiki.freebsd.org/WifiAggressiveMode>.
- [10] “iperf,”. <http://iperf.fr>.