無線ネットワークTAPデバイスを用いた無線LANエミュレータの設計

加藤 新良太[1]　高井 峰生[2]　石原 進[3]

概要：車車間通信や、Internet of Things (IoT) 等を利用するネットワークシステムやアプリケーションの研究・開発において、実際にネットワークを構築せずにそれらの動作や性能を評価できれば、評価環境の構築における経済的あるいは時間的コストを減らせる。そのような評価方法の 1 つにネットワークエミュレーションがある。これを用いると、ネットワークシステムの一部を抽象化し、それ以外の部分は、そのシステムを改変せずに実行できる。一部のネットワークシミュレータは、エミュレーション機能をもっており、その機能の実装にネットワーク TAP デバイスと呼ばれる仮想 Ethernet デバイスを利用している。ネットワーク TAP デバイスは、OS との間でやりとりするデータパケットをユーザ空間との間で送信・受信できる。しかしながら、ネットワーク TAP デバイスでは、無線 LAN フレームや送信電力や受信強度等の無線 LAN デバイスの制御情報を扱えため、ETSI ITS-G5 DCC などのクロスレイヤ制御を伴うネットワークシステムの評価が困難である。そこで、著者らは、ユーザ空間との間で無線 LAN フレームの送信・受信に加えて、無線 LAN デバイスの制御情報をやりとりできる無線ネットワーク TAP デバイスを提案している。本稿では、無線ネットワーク TAP デバイスを用いた無線 LAN ネットワーク向けのエミュレータを設計・実装について述べる。このエミュレーション環境は、Linux システムで動作する hostapd 等のネットワークアプリケーションをソースコードの改変することなく実行・評価できる。動作検証によって、無線 LAN エミュレータを用いてそれらのアプリケーション間で無線 LAN フレームの交換ができることを示した。

Design of a wireless LAN emulator with a wireless network tap device

ARATA KATO[1]　MINEO TAKAI[2]　SUSUMU ISHIHARA[3]

1. はじめに

ネットワークシステムやアプリケーションの開発において、物理的なネットワークを構築することなく、ネットワークシステムやアプリケーションの挙動を把握でき、かつその性能を評価できる環境があれば便利である。そのような評価方法の 1 つにネットワークエミュレーションがある。ネットワークエミュレーションは、ネットワークシステムを構成するハードウェア・ソフトウェアの挙動や、電波伝搬やノードのモビリティ等の通信環境における挙動のうち、一部を抽象化してネットワークシミュレータで模擬し、抽象化を行わない部分は元のシステムを改変することなく実環境と同様に実行できる評価方法である。

EXata[1]、ns-3[2]、Scenargie[3] 等の一部のネットワークシミュレータは、ネットワークエミュレーション機能を実現している。それらの機能では、オペレーティングシステム (OS) 内のユーザアプリケーションとネットワークシミュレータを接続し、ユーザアプリケーション間でやりとりするデータパケットに対して、転送遅延を発生させたり、帯域制御を行うことができる。これらのネットワークシミュレータが実現するエミュレーション機能では、ネットワーク TAP デバイスと呼ばれる仮想 Ethernet デバイスが利用されている。

ネットワーク TAP デバイスは、ネットワークシミュレー

[1] 静岡大学大学院総合科学技術研究科
Graduate School of Integrated Science and Technology, Shizuoka University
[2] 大阪大学/カリフォルニア大学ロサンゼルス校
Osaka University/University of California, Los Angels
[3] 静岡大学学術院工学領域
College of Engineering, Academic Institute, Shizuoka University

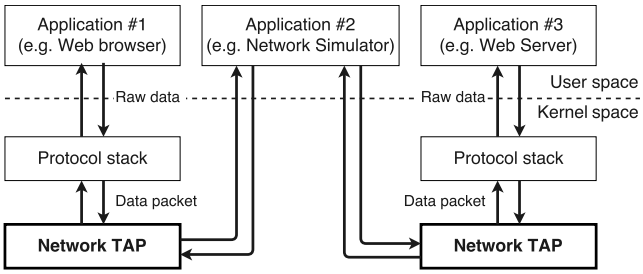

図 1 ネットワーク TAP デバイスによるエミュレーション例

タと OS 等の実システム間のインタフェースとして用いられる。**図 1** は、ネットワーク TAP デバイスを用いたネットワークシステムのエミュレーション例である。ネットワーク TAP デバイスは、OS に対して Ethernet デバイスであるように振る舞い、ユーザ空間との間でデータパケットをやりとりする機能を持つ。このため、実際はユーザ空間との間でやりとりしているデータパケットを、OS に対して Ethernet デバイスを通じてやりとりしているように見せることができる。

しかしながら、ネットワーク TAP デバイスを利用したエミュレーション機能では、SSID や受信信号強度 (RSSI) 等の無線 LAN の情報を必要とするネットワークシステムの評価は難しい。そのようなネットワークシステムの一例に、欧州電気通信標準化機構 (ETSI) が策定する車車間通信制御プロトコル (ETSI ITS-G5 DCC)[4] がある。ETSI ITS-G5 DCC では、ビーコンの送信制御に受信信号強度 (RSSI) の情報を必要とする。各車両は、ビーコンの衝突を避けるため、他車両から受信したビーコンの RSSI の挙動を監視し、自身の送信出力や送信間隔を動的に変える。ネットワーク TAP デバイス用いたエミュレーション環境では、無線 LAN デバイスの操作や制御等を行えないため、ETSI ITS-G5 DCC の制御方式を模擬できない。

そこで、著者らは無線 LAN に対応した無線ネットワーク TAP デバイス (wtap80211) を設計・開発した [5]。無線ネットワーク TAP デバイスは、OS に対して無線 LAN デバイスとして振る舞い、無線 LAN フレームと無線 LAN デバイスの制御情報の双方をユーザ空間との間でやりとりできる仮想無線 LAN デバイスである。

著者らは、[5] において、wtap80211 が Linux システムから送信された無線 LAN フレームをユーザ空間に送信できること、またユーザ空間で作成した擬似ビーコンを Linux システムで受信できることを示した。しかし、Linux システムで動作する hostapd 等の無線 LAN アプリケーション間における通信が行えるかの検証は行われていなかった。

本稿では、無線ネットワーク TAP デバイスを利用した無線 LAN エミュレータの設計・実装について述べる。Linux システムで動作するネットワークアプリケーションを用いた動作検証によって、この無線 LAN エミュレータを用いることで、ソフトウェアの改変やソフトウェア内の処理を

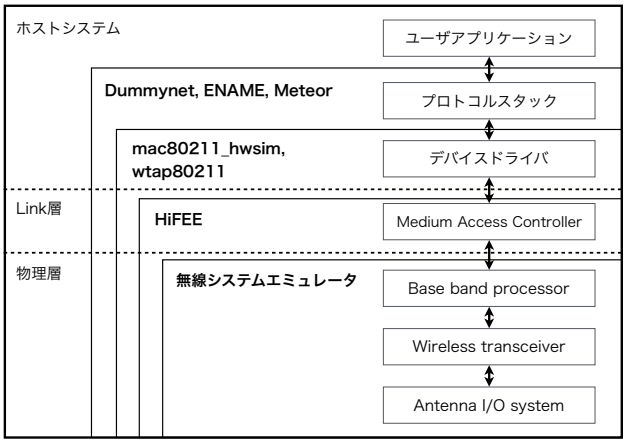

図 2 既存のネットワークエミュレータの分類

抽象化を行うことなく、Linux システム向けに実装されたネットワークシステムやアプリケーションの動作や性能を評価できることを示す。

以下、2 章で既存のネットワークエミュレータと wtap80211 に対する先行技術について述べ、3 章で無線ネットワーク TAP デバイスについて説明する。4 章で無線ネットワーク TAP デバイスを用いたネットワークエミュレータの設計について行い、5 章でその動作検証について述べる。6 章で本論文のまとめを述べる。

2. 関連研究

ネットワークエミュレーションの実現方法には、ネットワークシステムを抽象化する方法と抽象化する対象によって様々なものがある。本章で述べる既存のネットワークエミュレーション環境を、抽象化する対象別に分類したものを**図 2** に示す。例えば、**図 2** 中の中央左にある HiFEE は、リンク層以下を抽象化していることを表している。以下、IEEE 802.11 無線 LAN に着目して、抽象化の対象別に既存のネットワークエミュレーション技術の特徴とその課題について述べる。

2.1 Dummynet と netem

Dummynet は、BSD 系 OS に標準で搭載されているデータパケットの挙動を模擬するネットワークエミュレータである [6]。Dummynet は、BSD 系 OS のファイアウォールツール (ipfw) が持つパケットフィルタ機能を利用して、特定のネットワークインタフェースを監視し、そのインタフェースから送信されるデータパケットに対して遅延やパケットロスを発生させることができる。このため、データパケットの転送遅延やパケットロスがユーザアプリケーションの動作に及ぼす影響を評価できる。しかし、Dummynet は、データリンク層以下の挙動を模擬しないため、Ethernet や IEEE 802.11 等の異なるデータリンク層以下の通信規格の違いを考慮に入れた評価はできない。また、Linux では、

Dummynet と同様の機能を持つ netem を中心とするトラフィックコントロールの仕組みがある。

2.2 EMANE

EMANE (Extendable Mobile Ad-hoc Network Emulator) は、IEEE 802.11 無線 LAN におけるモバイルアドホックネットワーク (MANET) 上で動作するアプリケーションの挙動を評価可能なネットワークエミュレータである [7]。EMANE は、データリンク層と物理層の挙動を抽象化する。このエミュレータは、コンテナ型仮想化技術を利用することで、多数のネットワークノードによる無線通信を 1 台あるいは少数のホストマシンで模擬できる。コンテナ型仮想化技術とは、CPU リソース、メモリ領域、ネットワーク空間、プロセス等の OS のリソースを分割・隔離し、1 つの OS 上に複数の仮想システムを構築する仮想化技術である。この技術により、OS のプロトコルスタックやユーザアプリケーションを利用して、1 つの OS 上で複数端末間の通信を模擬できる。しかし、無線 LAN デバイスとそのデバイスドライバの動作は模擬できないため、無線 LAN フレームの送信・受信以外のネットワークシステムやアプリケーションによる無線 LAN デバイスの操作・制御を模擬できない。

2.3 HiFEE

河合らは、IEEE 802.11 無線 LAN に対応する仮想ハードウェアを利用したエミュレーション環境 HiFEE (High Fidelity Emulation Environment) を提案している [9]。HiFEE は、IEEE 802.11 PHY と MAC 副層を抽象したネットワークエミュレータである。このエミュレータは、実在する無線 LAN デバイス (Atheros 社製 AR9160) を再現した仮想ハードウェアを搭載する仮想マシンを利用する。この仮想マシン同士で、ホスト OS 上のネットワークシミュレータを介して IEEE 802.11 無線 LAN による通信を模擬する。

HiFEE で提供される仮想ハードウェアは、無線 LAN デバイスドライバが無線 LAN デバイスのハードウェアに対して送信する命令を実機と同様に解釈・実行するため、Linux システムで動作する実際の無線 LAN デバイスドライバ (ath9k) を実行できる。しかし、仮想マシンの構築・実行には、ホストマシンに高い計算処理能力が要求されるため、ネットワークノードが多数ある無線通信環境を再現すると実時間性が低下する。

2.4 Meteor と NETorium

明石らは、データリンク層以下を抽象化するネットワークエミュレータ Meteor を提案している [10]。Meteor は、あらかじめ各ノード間の信号品質を表すパラメータ (遅延時間、フレームロス率、etc.) をユーザが任意に定義する障害物やノードの位置等が記述された無線通信環境のシナリオから算出し、その算出したパラメータに基づいて Ethernet フレームに対して転送遅延やフレームロスを発生させる。

また、明石らは、Meteor を利用した無線ネットワークエミュレーション環境 NETorium を提案している [12]。NETorium は、Meteor に加えて Asteroid と呼ばれるアプリケーションで構成される。Asteroid は、Linux システムから送信された無線 LAN フレームを Ethernet フレームにカプセル化することで有線 LAN で無線 LAN フレームをやりとりし、有線 LAN ネットワーク上に擬似的な無線 LAN 環境を構築するアプリケーションである。NETorium は、Asteroid によって Ethernet フレームにカプセル化された無線 LAN フレームに対して、Meteor を用いて転送遅延を制御する。

NETorium では、エミュレーション環境の構築に Ethernet で接続された有線ネットワークを構築する必要がある。明石らの実装では、StarBED[11] と呼ばれる 1000 台以上の計算機サーバからなるテストベッド設備を用いて、仮想無線 LAN ノードを構築している。

2.5 無線システムエミュレータ

真野らは、IEEE 802.11 や LTE、TV ホワイトスペースで利用される周波数帯における電波伝搬を正確に模擬できる無線システムエミュレータを提案している [8]。無線システムエミュレータは、主にネットワークノードを模擬する仮想マシンと電波伝搬を模擬する空間エミュレータで構成される。このシステムでは、実際の無線デバイスが信号の送信過程で生成するデジタル I/Q データを基に電波伝搬をシミュレータで模擬する。無線システムエミュレータは、実時間性を確保するため、電波伝搬の模擬に十分に高速な計算処理装置が必要とする。真野らのシステム実装では、FPGA を用いて電波伝搬の計算処理が行われている。

2.6 mac80211_hwsim

本稿で議論する無線ネットワーク TAP デバイスに類似する技術に、mac80211_hwsim と呼ばれる Linux 用の仮想無線 LAN デバイスがある [13]。mac80211_hwsim は、Linux のシステムに対して無線 LAN デバイスであるように振る舞い、IEEE 802.11 フレームをユーザアプリケーションとの間でやりとりするが、無線 LAN デバイスの制御情報をユーザ空間との間でやりとりできない。

この無線 LAN デバイスの制御情報は、無線 LAN デバイスのハードウェアを制御するパラメータ、送信キューの制御パラメータ、BSSID の設定情報、無線 LAN フレームの送信・受信時の状態情報等が含まれる。例えば、無線 LAN デバイスのハードウェアを制御するパラメータには、ハードウェアスイッチの on/off フラグ、Reguratory domain に基づく送信電力の上限値や IEEE 802.11 の省電力機能で使

われるリッスン間隔が含まれる。このほかに、送信キューの制御パラメータには、キュー長やコンテンションウィンドウの最大・最小値、BSSID の設定情報には、ESSID やネットワークインタフェースのソフトウェアスイッチの情報等が含まれる。

このため、無線 LAN デバイスの制御情報の変更に応じて振る舞いを変えるネットワークシステムやアプリケーションの挙動を mac80211_hwsim を利用して模擬することは難しい。

2.7 既存エミュレーション環境の課題

既存のエミュレーション環境は、大まかにネットワークノード 1 台あたりの挙動を忠実に模擬するエミュレータ (HiFEE、無線システムエミュレータ) と、多数のノードで構成される大規模なネットワークの挙動を模擬するエミュレータ (EMANE、NETorium) に分類できる。前者は、ノード単体の挙動の再現度が高いが、その挙動を模擬するために高性能な計算機を必要とするため、スケーラビリティが低い。後者は、大規模なネットワークシステムの評価に適しているが、ネットワークノード 1 台あたりの再現性は前者と比較すると低い。また、エミュレーション環境の構築に経済的あるいは時間的なコストが多大にかかる。

車車間通信システムの評価においては、多数の自動車で送信出力を制御する場合、多数の通信ノードを模擬できる規模性と、電力制御や無線 LAN フレームの送信制御等のネットワークシステムによる無線 LAN デバイスの制御を模擬できる再現性の双方を満たすことが必要である。また、仮想マシンや大規模なテストベッド設備を必要としないエミュレーション環境が望まれる。

3. 無線ネットワーク TAP デバイス

本章では、本稿で述べる無線 LAN エミュレータに用いる無線ネットワーク TAP デバイスの概要と基本動作について説明する。

3.1 無線ネットワーク TAP デバイスの概要

無線ネットワーク TAP デバイス (wtap80211) は、著者らが設計・開発した、Linux システム上で無線 LAN デバイスとして振る舞うネットワーク TAP デバイスである [5]。**図 3** に wtap80211 の概要を示す。wtap80211 は、Linux 3.19 で動作し、Linux システムにおいて一種の無線 LAN デバイスドライバとして動作する。wtap80211 は、Linux OS に対して無線 LAN デバイスが実際に存在するように振る舞う。ただし、wtap80211 自身では実際の無線 LAN デバイスで行われるキャリアセンスや変調等の処理は模擬しない。これらの処理に関しては、wtap80211 を介して利用するネットワークシミュレータに委ねられる。

wtap80211 が模擬する仮想無線 LAN デバイスの構成を

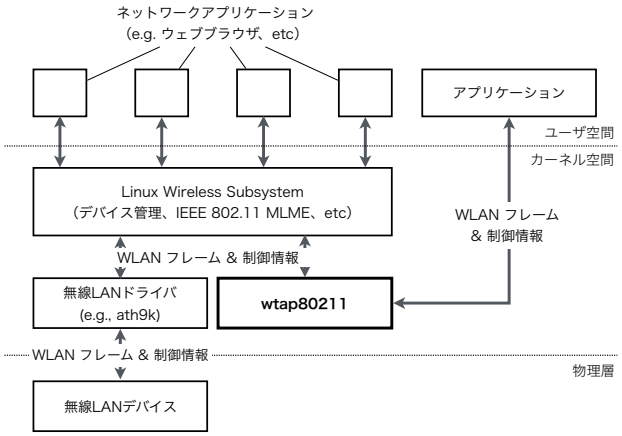

図 3 wtap80211 の概要

表 1 wtap80211 の構成情報

動作環境	Linux 3.19
無線規格	IEEE 802.11a/b/g/n/ac, IEEE 802.11p/s/ad (チャネルリストのみ)
動作モード	Managed (STA)/Master (AP)/IBSS/Monitor, Mesh/OCB

表 1 に示す。wtap80211 は、仮想的に構築された 2.4 GHz、5.0 GHz、60 GHz 帯の 3 つのチャネルリストを保持し、IEEE 802.11a/b/g/n/ac に加えて、IEEE 802.11p/s/ad に対応した無線 LAN デバイスとして振る舞うことができる。ただし、IEEE 802.11p の機能への対応は、Linux カーネルのバージョンに依存する。

また、wtap80211 は、ユーザ空間との間に Linux の無線 LAN システムを介さないデータパスを持っており、Linux 内の無線 LAN システムとユーザ空間との間のインタフェースとして動作し、無線 LAN フレームの送信・受信に加え、無線 LAN デバイスの制御情報をユーザ空間に通知できる。

3.2 基本動作

wtap80211 は、Linux システムにインストールされると仮想無線 LAN デバイスの情報を Linux の無線 LAN 制御システムに登録し、この仮想無線 LAN デバイスが Linux システムで利用可能であるように振る舞わせる。このため、Linux の無線 LAN 制御システムは、実際の無線 LAN デバイスに対して行うのと同様に、wtap80211 の仮想デバイスに対して無線 LAN フレームの送信・受信処理と、無線 LAN デバイスの制御処理を実行できる。

一般に、無線 LAN デバイスドライバは、無線 LAN フレームやデバイスの制御情報を PCIe や USB 等のバス (Bus) を介して無線 LAN デバイスとやりとりするが、wtap80211 は、ユーザ空間との間に Netlink ソケット [14] を持ち、Linux の無線 LAN システムを介さずにユーザ空間との間で無線 LAN フレームや無線 LAN デバイスの制御情報をやりとりする。

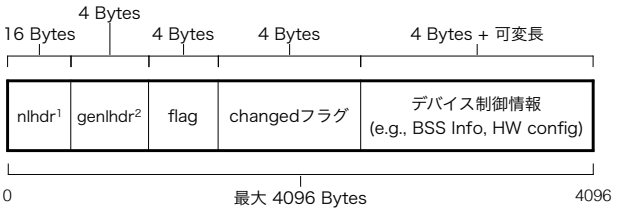

図 4 フレームメッセージのフォーマット

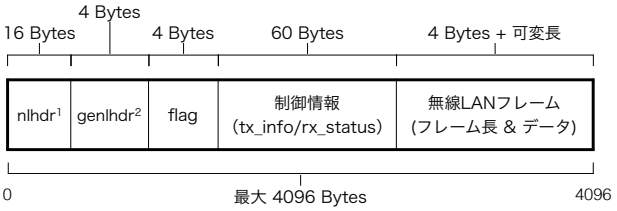

図 5 制御メッセージのフォーマット

具体的には、wtap80211 は、Linux システムから無線 LAN フレームの送信命令か無線 LAN デバイスの制御命令を受け取ると、ユーザ空間に向けて Netlink メッセージを送信する。wtap80211 から送信されるメッセージは 2 種類あり、1 つは、無線 LAN フレームを格納したメッセージ (フレームメッセージ)、もう 1 つは、無線 LAN デバイスの制御情報を格納したメッセージ (制御メッセージ) である。**図 4** 及び**図 5** に、それらのフォーマットを示す。 フレームメッセージと制御メッセージの先頭 20 Bytes (nlhdr と genlhdr) は、Netlink ソケットのプロトコルヘッダである。その直後に配置されている flag は、wtap80211 の内部で使うために予約された領域である。

フレームメッセージには、無線 LAN フレームのデータと送信・受信時の制御情報が格納される。送信・受信時の制御情報には、送信電力もしくは RSSI、使用周波数、帯域幅が含まれる。また、IEEE 802.11n/ac で送信する場合は、MCS のインデックス番号等が追加の情報として含まれる。

制御メッセージには、Linux システムが wtap80211 の仮想デバイスに対する設定値の変更、あるいは制御する場合に該当するデバイスの制御情報が格納される。制御メッセージに格納される情報には、 *a*) Basic Service Set (BSS) 設定情報、*b*) 受信フレームのフィルタリングルール、*c*) 送信キューの設定情報、*d*) 仮想無線 LAN デバイスのハードウェアの設定情報の 4 種類がある。例えば、BSS 設定情報には、基準送信電力、SSID、転送レート、ビーコン情報等のパラメータが含まれる。

wtap80211 の制御メッセージに含まれる制御情報と、フレームメッセージに含まれる送信・受信時の制御情報は、Linux の無線 LAN 制御システムで使われているデータ構造と同じであるため、Linux のカーネルヘッダを用いることで解析できる。RSSI や SSID 等の無線 LAN の情報を必要とするユーザアプリケーションは、フレームメッセージと制御メッセージを解析することで、無線 LAN フレームと無線 LAN デバイスの制御情報を取得する。

4. wtap80211 を利用した無線 LAN エミュレータ

本章では、3 章で述べた無線ネットワーク TAP デバイス (wtap80211) を利用した無線 LAN エミュレータの概要と基本設計について述べる。

4.1 要求要件

2 章で述べた既存のネットワークエミュレータは、アプリケーションやプロトコルスタック等のネットワークシステムで用いるソフトウェアを実環境と同様に実行できるが、仮想マシンの構築・実行に高性能な計算機環境を必要とする場合や、FPGA やテストベッド設備を必要とするため、経済的あるいは時間的コストが大きいという課題があった。これらの課題を解決するため、wtap80211 を利用する無線 LAN エミュレータに求められる要件は、

(a) ネットワークシステム・アプリケーションのソフトウェアをソースコードの改変なしに実行可能であること、

(b) OS もしくはユーザアプリケーションによって行われる無線 LAN デバイスの制御を模擬できること、

(c) 仮想マシンや大規模なテストベッド設備等がなくても多数のネットワークノード間の無線通信を模擬できること

の 3 つとなる。

条件 (a) は、実環境と同一のソフトウェアを動作させることで、ネットワークシステムのソフトウェア部分の挙動がソースコードの改変により実環境と異なることを防ぐために必要となる。また、エミュレーション評価のためにシステムの一部を改変する作業コストを省くことができる。

条件 (b) は、無線通信状況の変化に伴って動的に無線 LAN デバイスの制御を行うネットワークシステムにも対応することが求められる。このためには、シミュレータがネットワークシステムによるデバイス制御を把握できる仕組みが必要である。

条件 (c) では、仮想マシンや特別な計算機環境がない場合でも、多数の無線 LAN ノードを同時に模擬することが求められる。

そこで、wtap80211 を利用して単一の Linux システム上

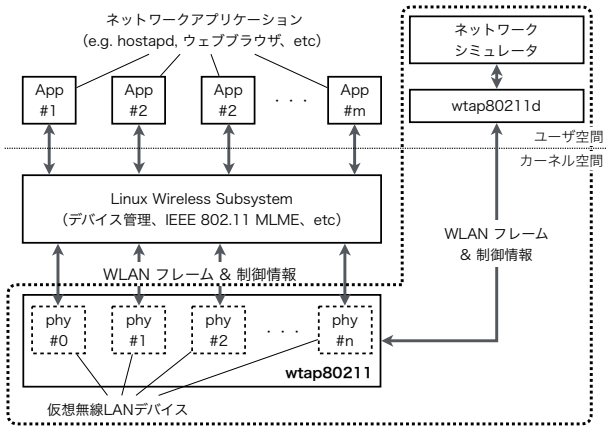

図 6 無線 LAN エミュレータの概要

で動作するエミュレーション環境を設計した。

4.2 基本設計

設計した無線 LAN エミュレータは、wtap80211 がインストールされた Linux システムと、その Linux システム上で動作するネットワークシミュレータで構成される。**図 6** に無線 LAN エミュレータの概要を示す。破線の角丸の四角で囲まれた部分は、無線 LAN エミュレータによって抽象化される部分を示しており、MAC 副層と物理層の挙動を wtap80211 とユーザ空間上のネットワークシミュレータで抽象化する。wtap80211 では IEEE 802.11 MLME の処理を実行し、キャリアセンスと物理層の挙動はネットワークシミュレータで模擬する。

抽象化されていない部分では、Linux システムのネットワークプロトコルスタックとユーザアプリケーションが動作する。これにより、Linux システムで動作するアプリケーションであれば、エミュレーション評価のためにアプリケーションの抽象化やソースコードの改変は必要ない。

抽象化部分では、wtap80211 を利用して複数台の仮想無線 LAN デバイスを構築し、複数の無線 LAN ノードが存在する環境を模擬する。wtap80211 の各仮想無線 LAN デバイスは、Netlink ソケットによってネットワークシミュレータと接続し、シミュレータとの間で無線 LAN フレームと無線 LAN デバイスの制御情報をやりとりする。

ユーザアプリケーションとプロトコルスタックは、実際の無線 LAN デバイスを用いて通信する場合と同様に、Linux の無線 LAN システムを介して、wtap80211 の仮想無線 LAN デバイスと無線 LAN フレームや無線 LAN デバイスを制御する。一方で、ネットワークシミュレータは、wtap80211 とやりとりする Netlink メッセージのデコード/エンコードを担うデーモンアプリケーション (wtap80211d) を介して wtap80211 と通信し、wtap80211 との間でやりとりする無線 LAN フレームと無線 LAN デバイスの制御情報を基に、物理層における無線 LAN フレームの振る舞いを模擬する。

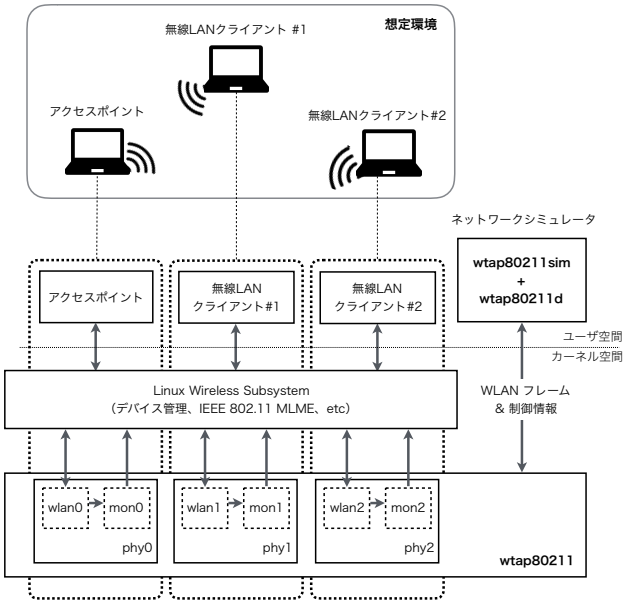

図 7 動作検証環境の概要

5. 動作検証

4 章で述べた要件を無線 LAN エミュレータを、Linux 3.19.0-78-generic (Ubuntu 14.04.05 LTS) が動作する単一の Linux マシンに実装した。[5] において、著者らは wtap80211 をユーザ空間との間で無線 LAN フレームと無線 LAN デバイスの制御情報をやりとりできることを示した。しかしながら、Linux システムをアクセスポイント化するアプリケーションである hostapd[15] 等の既存のネットワークアプリケーション間で通信を行えるかは検証していない。そこで、hostapd に加えて、動作モードの変更や使用チャネルの設定等の無線 LAN デバイスの操作に用いる iw コマンドと、ネットワークインタフェース有効化/無効化等の操作に用いる ip コマンドを用いて、複数の仮想無線 LAN デバイス間で認証及びアソシエーション処理が行えるかを検証し、

(a) ネットワークシステムやアプリケーションをソースコードの改変を行うことなく実行可能であること、

(b) 単一の Linux システムのみで複数の無線 LAN ノードによる通信を模擬可能であること

の 2 点を満たしているかを確認した。ただし、多数のネットワークノード間における無線通信の模擬は本稿では行っていない。**図 7** に動作検証環境の概要を示す。この動作検証では、そのうち 1 台はアクセスポイント、残る 2 台はステーションとして動作する合計 3 台の無線 LAN ノードで構成される無線 LAN 環境を想定した。

ユーザ空間には、hostapd が稼働するプロセスと、iw コマンドや ip コマンド等を通じてネットワークインタフェースを操作するプロセスが 2 つ (STA1、STA2) 動作し、そ

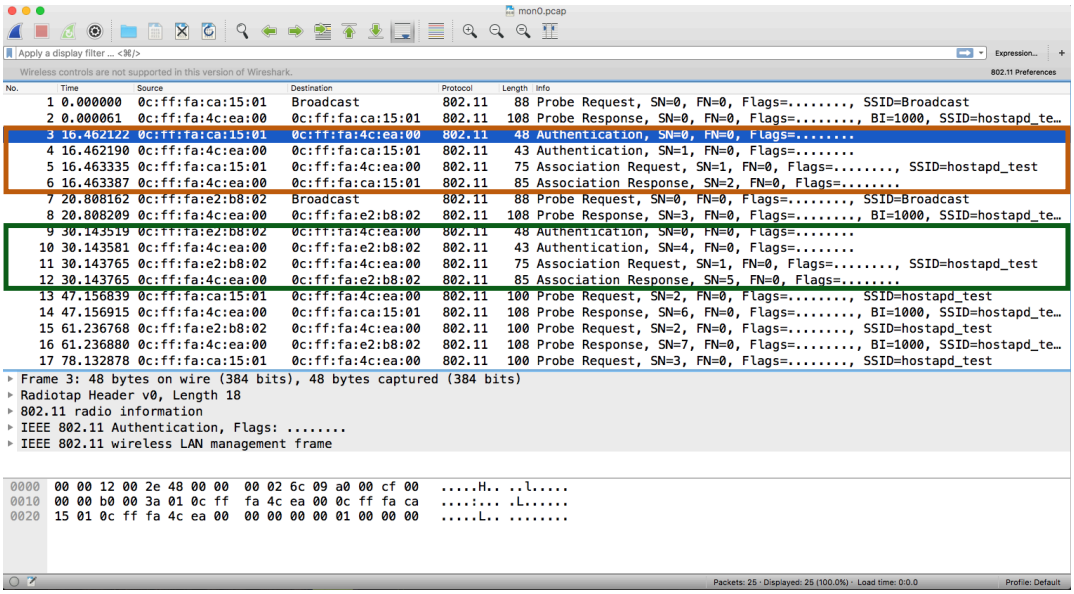

図 8 AP モード動作時の wlan0 における無線 LAN フレームのキャプチャ結果

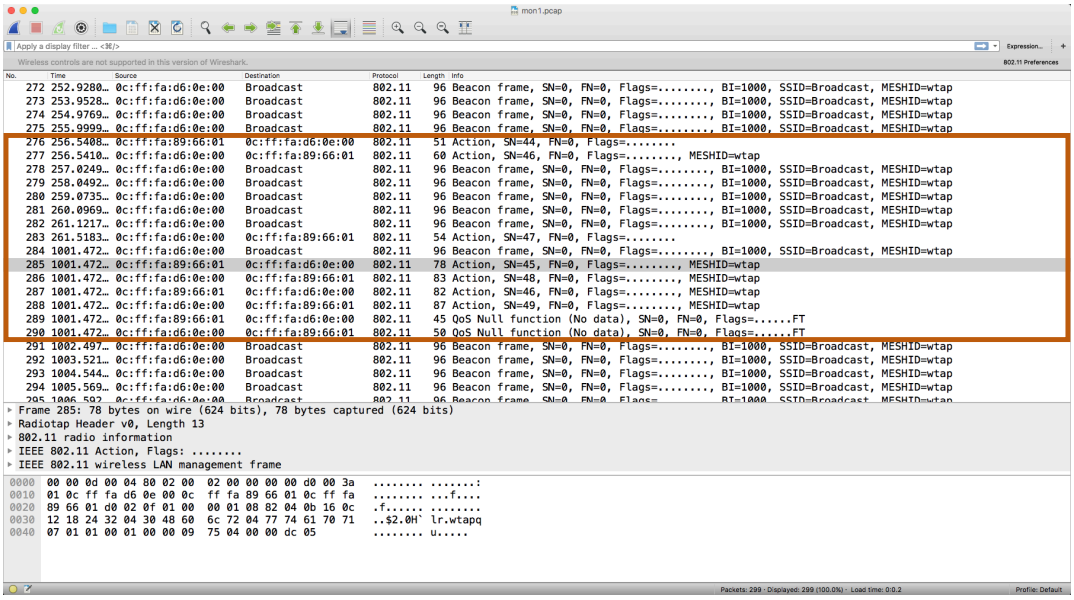

図 9 Mesh Point モード動作時の wlan1 における無線 LAN フレームのキャプチャ結果

のほかにシミュレータ (wtap80211sim) が動作している。wtap80211sim は、この動作検証のために著者らが実装した、wtap80211d[5] の機能を拡張したネットワークシミュレータである。このシミュレータは、wtap80211 からフレームメッセージを受け取り、wtap80211d でフレームメッセージを解読すると、フレームメッセージの送信制御情報を基に、RSSI 等の受信制御情報を構築し、無線 LAN フレームと受信制御情報を wtap80211d を通じて wtap80211 に送り返す。

カーネル空間では、wtap80211 を用いて仮想無線 LAN デバイスを 3 つ (phy0、phy1、phy2) 生成し、それぞれの仮想無線 LAN デバイスに、ネットワークインタフェースを 2 つずつ割り当てた。1 つは、ネットワークアプリケーションから wtap80211 の仮想無線 LAN デバイスを操作するために用いるネットワークインタフェース (wlan0、wlan1、wlan2) である。もう 1 つは、wtap80211 の仮想無線 LAN デバイスで送信・受信する無線 LAN フレームをキャプチャするために用いるインタフェース (mon0、mon1、mon2) である。ユーザ空間の hostapd は、wlan0 を操作して wlan0 を AP として動作させる。STA1 は wlan1、STA2 は wlan2 を操作し、AP との接続を行う。

[5] において、著者らは、wtap80211 からユーザ空間に向けて送信された無線 LAN フレームをユーザ空間でキャプチャできることを示し、またユーザ空間から擬似ビーコンを wtap80211 で受信できることを示した。しかし、ネットワークアプリケーションが送信する本物の無線 LAN フ

レームを用いて、ネットワークアプリケーション間でデータ通信が行えるかは実証していない。そこで、wtap80211の各仮想無線 LAN デバイスに割り当てたモニタデバイスと、パケットキャプチャアプリケーションの Wireshark[16]を用いて、仮想無線 LAN デバイスで送信・受信される無線 LAN フレームをキャプチャし、無線 LAN エミュレータ上で、ネットワークアプリケーション間で無線 LAN フレームを用いた通信が行えていることを検証した。

図 8 に、STA1 及び STA2 で iw コマンドを用いて、AP とアソシエーション処理を行った際に、AP 側でキャプチャした無線 LAN フレームを Wireshark で見た際の結果を示す。AP-STA 間の認証はオープン認証とした。[5] で述べたように、wtap80211 の各仮想無線 LAN デバイスには、他の無線 LAN デバイスと重複しない MAC アドレスが割り当てられる。その MAC アドレスのフォーマットは、先頭 24 bits に固定値 (`0c:ff:fa`)、それに継ぐ 16 bits に乱数、最後の 8 bits にネットワークインタフェースの識別番号が割り当てられる。例えば、phy0 に MAC アドレスは、`0c:ff:fa:4c:aa:00` のように割り当てられる。これにより、MAC アドレスの末尾 8 bits を見ればどのネットワークインタフェースから送信されたかを区別できる。**図 8** の橙色の四角の部分では、wlan0 と wlan1 との間で認証フレームとアソシエーションフレームがキャプチャできていることがわかり、AP-STA1 間の認証に必要な無線 LAN フレームが交換されていることがわかる。また同様に、緑色の四角で囲まれた部分では、wlan0 と wlan2 との間で AP-STA2 間でも認証に必要な無線 LAN フレームがやりとりできていることがわかる。

次に、iw コマンドを用いて wlan1 と wlan2 の動作モードを Managed (STA) モードから Mesh Point モードに変更し、wlan1 と wlan2 の間でメッシュネットワークを構築できるかを検証した。**図 9** に、wlan1 でキャプチャした無線 LAN フレームを Wireshark で見た際の結果を示す。**図 9** の橙色で囲まれた部分で、メッシュネットワークを構築する際に交換される Action フレームが wlan1 と wlan2 の間でやりとりされていることが確認できる。

6. まとめ

本稿では、Linux システムで動作するアプリケーションのソースコードを改変することなく、ネットワークシステムやアプリケーションによって制御される SSID や信号強度等の無線 LAN デバイスの制御情報をユーザ空間で扱うことができ、かつ仮想環境やテストベッド設備を必要としない無線 LAN エミュレータの設計を行った。動作検証により、ネットワークアプリケーションによって設定・変更される無線 LAN デバイスの制御情報をネットワークシミュレータで扱うことができ、それらのネットワークアプリケーション間で無線 LAN フレームを交換できることを確かめた。

今後、本稿では行わなかった多数のネットワークノード間の無線通信を模擬できるかを検証する必要がある。特に、模擬できるネットワークノードの数は、無線 LAN エミュレータが動作する Linux マシンの計算機の能力に影響を受ける。また、無線 LAN エミュレータにおける無線 LAN フレームの転送遅延時間は、物理層の挙動を模擬するネットワークシミュレータの実装に依存することが考えられる。このため、wtap80211 が仮想無線 LAN デバイスを模擬する際に計算機にかかる負荷を検証することと、Scenargie 等の既存のネットワークシミュレータを接続した場合に、無線 LAN フレームの転送遅延を計測することが求められる。また、ETSI ITS-G5 DCC のオープンソースでの実装例である OpenC2X[17] を無線 LAN エミュレータで実行し、車両の移動や電波伝搬等の影響を考慮した評価を行う予定である。

謝辞

本研究の一部は、文部科学省研究費補助金基盤研究 (B) (15H02689) の支援を受けて行った。また、本研究を進めるにあたり、有益な助言とご意見を頂いた、Space-Time Engineering, LLC の金田茂氏、Space-Time Engineering Japan, Inc の前野誉氏に御礼申し上げます。

参考文献

[1] Scalable Network Techonologies: EXata, `http://web.scalable-network.com/exata` (参照 2017-05-09).
[2] ns-3 project: ns-3, `https://www.nsnam.org` (参照 2017-05-09).
[3] Space-Time Engineering, LLC: Scenargie, `https://www.spacetime-eng.com/en/products` (参照 2017-05-09).
[4] ETSI TS 102 687 v1.1.1 Intelligent Trasnport Systems (ITS): Decentralized Congestion Control Mechanisms for Intelligent Transport Systems Operating in the 5 GHz range; Access layer part, ETSI, July, 2011.
[5] 加藤新良太、高井峰生、石原進: Linux 向け無線ネットワーク TAP デバイスによる IEEE 802.11 無線 LAN エミュレーション環境の実装、信学技報, Vol. 116, No. 407, pp. 89–94, 2017.
[6] M. Carbone and L. Rizzo: Dummynet revisited, ACM SIGCOMM Computer Communication Review, Vol.40, No.2, pp.12–20, 2010.
[7] U.S. Naval Research Laboratory: The Extendable Mobile Ad-hoc Network Emulator (EMANE), `https://www.nrl.navy.mil/itd/ncs/products/emane` (参照 2017-05-09).
[8] 真野浩, 猿渡俊介: 無線システムエミュレータの実装と評価, 情報処理学会論文誌, Vol.55, No.5, pp.1541–1554, 2014.
[9] 河合孔明, 金田茂, 高井峰生, 峰野博史: シミュレータと仮想マシンを連携した高忠実度評価環境の提案, 情報処理学会研究報告, Vol.2015-MBL-75, No.22, pp.1–8, 2015
[10] 明石邦夫, 井上朋哉, ラズバン・ベウラン, 篠田陽一: Meteor: 大規模ネットワーク実験環境における無線ネットワークエミュレータの設計と実装, 電子情報通信学会論文誌, Vol.J98–B, No.4, pp.357–372, 2015.

[11] 国立研究開発法人情報通信研究機構: StarBED, http://starbed.nict.go.jp.
[12] K. Akashi, T. Inoue, S. Yasuda, Y. Takano, and Y. Shinoda: NETorium: High-fidelity Scalable Wireless Network Emulator, 12th Asian Internet Engineering Conference (AINTEC '16), pp.25–32, ACM, 2016.
[13] Linux Wireless: mac80211_hwsim, https://wireless.wiki.kernel.org/en/users/drivers/mac80211_hwsim, (参照 2017-05-09).
[14] P. N. Ayuso, R. M. Gasca, and L. Lefevre: Communicating between the kernel and user-space in Linux using Netlink sockets, Software – Practice and Experience, Vol. 40, No. 9, pp. 797–810, 2010.
[15] J. Malinen: hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator, https://w1.fi/hostapd (参照 2017-05-09).
[16] Wireshark foundation, Wireshark, https://www.wireshark.org (参照 2017-05-09).
[17] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann: Demo: OpenC2X – An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5, 2016 IEEE Vehicular Networking Conference (VNC '16), pp.152–153, IEEE, 2016.