様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　５ 年　 ６ 月 ２１ 日現在

機関番号：９４３０１
研究種目：基盤研究(C)（一般）
研究期間：2020～2022
課題番号：２０Ｋ１１７８３

研究課題名（和文）実環境とシミュレーション環境の融合による無線通信パラメタの探索

研究課題名（英文）Simulator-assisted search of wireless parameters for link quality improvement

研究代表者
玉井　森彦（Morihiko, Tamai）

株式会社国際電気通信基礎技術研究所・適応コミュニケーション研究所・主任研究員

研究者番号：９０５２３０７７

交付決定額（研究期間全体）：（直接経費）　　3,400,000円

研究成果の概要（和文）：無線LANは，単なる高速通信のみでなく，高信頼性や公平性など多様な通信要求への対応が求められてきている．無線LANは規格の更新に伴い，より多様な通信パラメタを設定可能となっているが，通信パラメタは多数存在し，かつそれらが互いに関係し合うため，それらの組み合わせも考慮すると，通信パラメタの探索空間は膨大となる．本研究では，実環境上での端末間の通信状況から得られる情報をもとに，シミュレーション環境を援用した通信パラメタの選択に役立てるために必要な技術である，実環境からの情報取得方式を提案する．

研究成果の学術的意義や社会的意義
本研究では，(1) MAC層情報の取得に関して，キャプチャツールの並列化を行うことで，MAC層情報を高速にシミュレーション環境下に取り込む方式の実現と，(2) 物理層情報をシミュレーション環境下にできるだけ正確に取り込む方式として，無線フレームのプリアンブル部の信号への正確な同期方式を考案した．これらの方式により，実環境での端末間の通信状況から，それをシミュレーション環境内へできるだけ高速に，かつ正確に取り込むことができるようになり，シミュレーション環境と実環境間の整合性の向上が可能となる．

研究成果の概要（英文）：Wireless LAN has been demanded to meet various communication requirements, including not only high-speed communication but also high reliability, fairness, and so on. With the amendments of the 802.11 standards, wireless LAN has become capable of setting a wider range of communication parameters. However, there are many communication parameters and they are related to each other, the search space becomes huge when their combinations are taken into account. In this study, we propose a method for obtaining information from real communication environments, which is an important technique for selecting communication parameters with the aid of a simulation environment reflecting communications between terminals in a real environment.

研究分野：モバイルコンピューティング

キーワード：無線LAN

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

IoT（Internet of Things）技術の進展に伴い無線 LAN の適用環境は多様化している．無線 LAN は，単なる高速通信のみでなく，高信頼性や公平性など多様な通信要求への対応が求められてきている．無線 LAN は規格の更新に伴い，より多様な通信パラメタを設定可能となっており，多様な通信要求に応じて通信性能を適切に調整するための手段は用意されていると言える．一方で，通信パラメタは多数存在し，かつそれらが互いに関係し合うため，それらの組み合わせも考慮すると，通信パラメタの探索空間は膨大となる．このような膨大なパラメタの中から，環境に応じて適切なパラメタを選択することが求められる．この問題への一つの解決策として，本研究では，実環境上での端末間の通信状況から得られる情報をもとに，それをシミュレーション環境下における通信モデルのパラメタ設定に利用するとともに，シミュレーション環境を援用した適切な通信パラメタの選択に役立てることを想定する．

２．研究の目的

本研究では，シミュレーション環境援用における実環境下での端末間の通信状況からの情報取得方式に焦点をあてる．シミュレーション環境でのモデルのパラメタ設定に必要となる情報として特に，物理層および MAC（Medium Access Control）層から得られる情報が重要となる．物理層に関する情報の取得については，USRP（Universal Software Radio Peripheral）等のソフトウェア無線機をベースとしたセンサを，MAC 層に関する情報の取得については，市販の無線 LAN インタフェースを搭載したセンサを想定する．これらのセンサを用いて物理層情報および MAC 層情報を取得するにあたっては，実環境下で発生する膨大なデータ量に対しいかに遅れなくシミュレーション環境下に取り込むかを考える必要がある．またそれと同時に可能な限り正確な情報を取得する必要もある．これらの点への対策が不十分である場合，構築したシミュレーション環境と実環境との間に無視できない不整合が生じる．これらの点への対策として，本研究では，(1) MAC 層情報の取得に関して，キャプチャツールの並列化を行うことで，MAC 層情報を高速にシミュレーション環境下に取り込む方式の実現と，(2) 物理層情報をシミュレーション環境下にできるだけ正確に取り込む方式として，無線フレームのプリアンブル部の信号への正確な同期方式を開発した．これらの方式について，以下に詳述する．

３．研究の方法

はじめに，MAC 層に関する情報を高速にシミュレーション環境内に取り込むための，パケットキャプチャツールの並列化方式に関して述べる．本研究で使用したパケットキャプチャツールは，tcpdump と Wireshark の 2 つである．これらのツールの役割分担は次の通りである．tcpdump は，無線 LAN インタフェースでの受信フレームを pcap 形式のデータとして取得するために使用される．一方で，Wireshark は，受信フレームに含まれる各ネットワークプロトコルのヘッダ部に含まれる情報を抽出するために使用される．本研究における MAC 層情報抽出のためのデータフローを図 1 に示す．

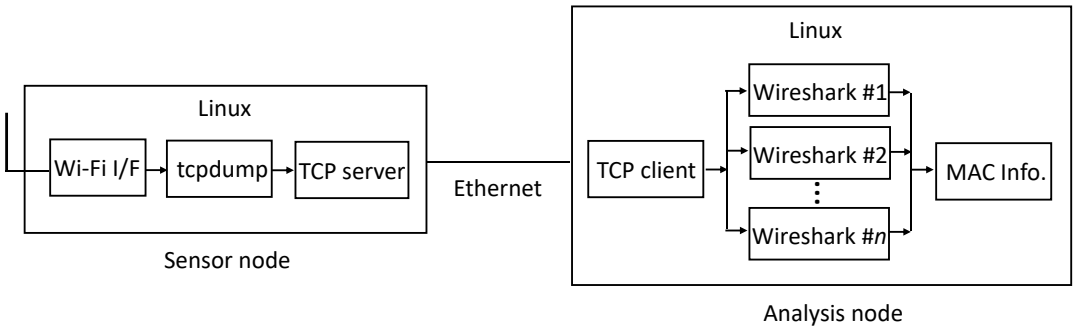

図 1　MAC 層情報の抽出のためのデータフロー

まず，センサノード側において無線 LAN インタフェースにより受信された各無線 LAN フレームは，tcpdump により順次キャプチャされ，pcap 形式のデータとして出力される．次に，センサノードと解析ノード間の TCP コネクションを介して，pcap 形式のデータがセンサノードから解析ノードへ順次転送される．解析ノード側では，複数の Wireshark をそれぞれ独立したプロセスとして並列して実行し，センサノードから受信された各 pcap 形式の無線フレームのデータをいずれかの Wireshark プロセスに渡し，ヘッダ情報の抽出を行う．各 Wireshark プロセスから出力されるヘッダ情報は，センサノードでの各無線フレームの受信順を維持するように整列され，最終的に各無線フレームの MAC 層情報の時系列が出力される．なお，送受先 IP アドレス

のようなより上位の層の情報も抽出可能であるが,無線環境に関する情報の抽出の観点からは,MAC 層情報の取得が主となる.

Wireshark の主要な機能として，各受信フレーム（パケット）のヘッダ情報（バイナリデータ)を解析し,その内容をテキストデータへ変換することが挙げられる.この機能は,Wireshark のソースコードパッケージ下の「epan/dissectors」ディレクトリ下のファイル群により実現されている．Wireshark の簡便なコマンドインタフェースとして，tshark コマンドが用意されている．これを用いることで，pcap 形式のデータから，各種プロトコルのヘッダ情報を抽出してテキストデータへと変換することができる.tshark コマンドを利用して MAC 層情報を抽出しようとする場合，単純な方法としては，tshark コマンドの出力結果をさらにパースして必要な情報だけ抽出する別のプログラムを動作させることが考えられる．しかしこの方法には次のような問題点がある．すなわち，tshark の出力結果のテキストデータが膨大なため，その出力を別のプログラムに入力するという方法には，出入力処理だけで相当の計算量が必要となる．さらに，その膨大な入力テキストデータをパースし，必要な情報のみ抽出する処理にも相当の計算量が必要となる．従ってこの単純な方法では，送受信機間の高速な通信に伴い発生する大量の pcap データに対し，MAC 層情報の抽出処理が追い付かない場合がある．そこで本研究では，Wireshark のソースコードへ最低限の変更を加え,Wireshark の実行プロセス単独で MAC 層情報の抽出処理を完結させる方法を採用するものとした．なお，MAC 層情報の抽出に Wireshark を頼らず,自前で 802.11 規格のヘッダ情報(バイナリデータ)をパースすることも考えられるが，Wireshark のソースコードを見ても分かるように，その仕様は相当に複雑であり，Wireshark の内部パーサを頼る方法以外に現実的な方法はないように思われる．そして，その方法を採る場合には，上記で述べた計算量の問題に由来し，抽出処理の遅れの問題が生じる.

上記のように，一つの Wireshark プロセス内で MAC 層情報の抽出処理を完結させる方法を採ったとしても，送受信機間で高速な通信が行われる場合には，MAC 層情報の抽出処理が追い付かない可能性がある．そこで，近年の CPU には複数のコアが搭載されていることを考慮して，図 1 に示したように複数の Wireshark プロセスを並列して実行し,抽出処理のスループットを向上させる方法についても実装を行った.

次に,物理層情報の抽出に関して述べる.物理層情報については,ソフトウェア無線機(USRP)により取得する．ソフトウェア無線機を用いて得られる物理層情報としては，例えば，信号強度，伝搬路特性，SN 比，位相回転等が挙げられる．特に，伝搬路特性については，送受信機間の通信路特性の再現を行う上で重要なデータであり，これを利用してシミュレーション環境内における高精度な通信路モデルの構築が可能となる.

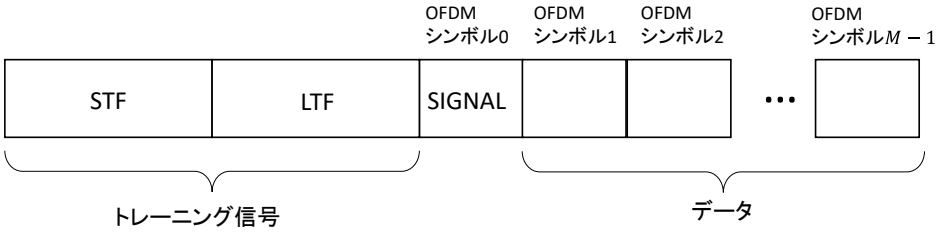

図 2　802.11a/g 対応の無線フレームの構成

実通信環境からの物理層情報の抽出においては，できるだけ正確なデータを取得できることが望ましい．この抽出処理は，無線フレームの受信側において，ソフトウェア無線機を用いて IQ（In-phase and Quadrature）信号を取得し，その解析を行うことで実現する．その際，抽出処理における物理層情報の正確さを左右する重要な処理として，プリアンブルとの同期処理が挙げられる．802.11a/g 対応の無線フレームのフレーム構成を図 2 に示す．各無線フレームの先頭のプリアンブル部には，STF（Short Training Field），LTF（Long Training Field）と呼ばれるトレーニング信号が含まれており，これらの検出を行うことで，無線フレームの存在を把握するとともに,復調対象となるサンプルの開始時点の特定を行う.この同期処理の精度は,後続の信号の復調に伴い得られる物理層情報の正確さに影響を与える．本研究では，特に STF 部の信号パターンに対し，その開始時点のサンプル位置をできるだけ正確に特定するアルゴリズムを考案した.

４．研究成果

キャプチャツールによる MAC 層情報の高速抽出方式の実現のため，Wireshark 3.6.5 をベースとして,ソースコードへの最低限の機能追加と,ツールのインタフェース部の実装を行った.本研究で必要となる Wireshark 本体の機能は,動的リンクライブラリである libwireshark.so, libwiretap.so，libwsutil.so の 3 つのライブラリにより提供されている．これらのライブラリに含まれる関数や構造体の仕様は，Wireshark のバージョンに依存して変更される可能性が

あり，本研究に関連する機能の追加を行うにあたって，Wireshark 本体のソースコードへ依存するコードの量はできるだけ少なくしたい．これについては，Wireshark 本体のコードには一切変更を加えることなく，追加で約 800 行のコードを作成することにより実現した．この 800 行が，Wireshark のバージョンに依存して変更を要する可能性のあるコード行数である．また，ユーザインタフェース部のコード行数は，約 800 行であり，C++言語と C 言語により実装されている．

図 3 に，pcap 形式のデータに含まれる各無線フレームについて，再送フラグの値（0 は再送でないことを，1 は再送であることを表す）を提案ツールにより抽出する場合のコードの例を示す．wsk_func 関数の第二引数に渡されるラムダ式の本体は，Wireshark による各無線フレームのパースが完了した後に，個々の無線フレームごとに呼び出される．図 3 のコードでは，Wireshark による再送コードを表すデータのパース結果の文字列（例えば，「.... 0... = Retry: Frame is not being retransmitted」）に対し，再送フラグの値（0 か 1）のみを抽出し，標準出力へ出力する処理を行っている．このソースコードは，Wireshark の本体（libwireshark.so 等）と直接リンクされ実行バイナリに組み込まれ，Wireshark の本体と同一のプロセス内で実行される．そのため，前述したオーバヘッドを回避し，高速に MAC 層情報を抽出することが可能となる．

```
int main(int argc, char *argv[])
{
 wsk_args args;

 args.input_file = argv[1];

 const bool failed = wsk_func(args, [](const wsk_output_map& output_map) {
  // this block will be called for each received frame
  for (auto& kv : output_map) {
   const string key = kv.first;
   string value = kv.second.at(0);
   if (key == "wlan.fc.retry") {
    value = value.substr(5, 1);
    cout << value << endl;
   }
  }
 });

 return 0;
}
```

図 3　各無線フレームの再送フラグの値を抽出する場合のコード例

次に，複数の Wireshark プロセスを並列して実行し，抽出処理のスループットを向上する方法について述べる．これは次のように実現される．複数の Wireshark プロセスのとりまとめの役割を担う一つのプロセス（以下，コーディネータとよぶ）を実行する．コーディネータは，必要な数の Wireshark プロセスを実行し，センサノードから受信した pcap データをパースし，一つ一つの無線フレームごとに pcap データの分割を行う．各部分データは一つの無線フレームの情報を保持している．この部分データをラウンドロビンにより各 Wireshark プロセスへ入力データとして与える．例えば，Wireshark プロセスの数が 3 の場合，1 番のプロセス，2 番のプロセス，3 番のプロセス，1 番のプロセス，という順で各無線フレームのデータを Wireshark プロセスへ入力する．各 Wireshark プロセスにより各無線フレームの MAC 層情報が抽出されるが，その出力順序は Wireshark プロセスへの CPU 割り当て状況に依存するため，抽出処理の早く終わったフレームの順で情報を出力すると，入力順序との整合性が失われてしまう．そこで，各 Wireshark プロセスへのデータ入力の際，別途各フレームにグローバルなフレーム番号を割り当てておき，その番号を Wireshark プロセスによる抽出処理後の出力の際にデータに付随させておく．これにより，コーディネータ側でフレーム番号順に並び変えを行い，入力順と出力順との間に不整合が生じないようにする．本機能により複数個の Wireshark プロセスを同時並列に実行することができるため，単一の Wireshark プロセスで処理する場合と比べ，より大量の pcap データの処理が可能となった．

次に物理層情報の抽出のための，STF 部の信号に対する同期アルゴリズムについて述べる．提案アルゴリズムでは，以下のステップで STF 部の信号の開始時点のサンプル位置を特定する．

(1) あらかじめ定められた閾値に対し，信号強度がその値を上回ったタイミングのサンプルを最初のサンプルとして，連続した 207 サンプルに対し，既知の STS (Short Training Sequence) 信号（16 サンプル）との相互相関関数の値を計算することで，計 192 サンプル分の相互相関関数の値の信号列を求める．(2) 192 サンプル分の信号列を，16 サンプルごとにひとまとまりとして，計 12 グループに分割する．(3) 各グループにおいて，相互相関関数の値のピークの位置の添え字番号（0 以上 15 以下の整数）を求め，12 グループ分の添え字番号から成る 12 要素の配列を生成する．(4) 配列の隣接するもの同士での差を求めることで，計 11 要素の配列を生成する．(5) 生成された 11 要素の配列に対し，値が 0 ならそのまま，さもなくば値を 1 に更新する．以上の操作により，11 要素の 0 か 1 の値を持つ配列が生成される．以下では便宜上，この 11 要素分の配列の値を連結した文字列（例えば，「10000000001」）により，配列を表す．

この配列は，STF が 192 サンプル分の範囲のどの範囲に存在するかを表しているもので，具体的には，0 の値が連続して 9 個現れる場所が，その範囲となっている．その範囲が特定できれば，それに伴い，STF の開始サンプル番号も次のように特定できる．例えば，「10000000001」の場合には，STF の開始サンプル番号は，16 以上，31 以下の範囲のいずれかであり，その値は，ステップ(3)で求められたピーク位置の添え字番号（0 以上 15 以下の整数）を参照することで特定することができる．

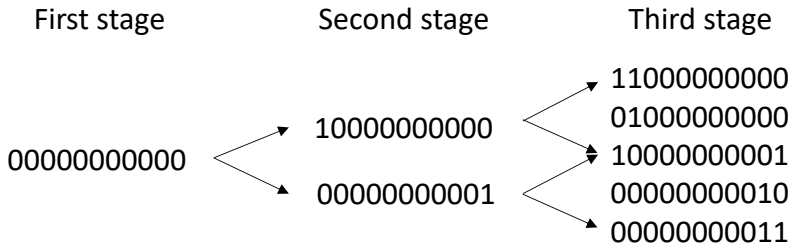

図 4　STF の存在範囲を表す配列が取りうる値

上記の最後のステップで求められる，STF の存在範囲を表す配列が取りうる値は，図 4 のように整理できる．図では，配列の値の可能性として第一段階から第三段階まで存在し，第三段階に該当する値であれば，0 が 9 個連続する範囲が一意に特定できるため，そこから STF の開始サンプル番号も特定できる．第一段階および第二段階に該当する値の場合には，0 が 9 個連続する範囲にあいまいさが残っているため，さらに絞り込みを行い，図内に矢印で示されるように，次段階に進む必要がある．絞り込みは次のように行う．すなわち，上記のステップ(3)で求められる相互相関関数のピークの値が最も小さいグループに対し，それに該当する配列の要素の値を 1 とする．以上のアルゴリズムにより，STF との高精度な同期が実現でき，その結果として，より正確な物理層情報の抽出が可能となる．

５．主な発表論文等

〔雑誌論文〕　計0件

〔学会発表〕　計0件

〔図書〕　計0件

〔産業財産権〕

〔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