

Raspberry Pi を利用した無線ネットワーク状態評価システム

田中晋太郎[†] 田川 真樹^{††} 高嶋 健人 近藤 賢郎^{†††} 北口 善明^{††††}

[†] 東京大学 ^{††} 九州工業大学 ^{†††} 慶応義塾大学 ^{††††} 金沢大学

E-mail: †tanaka@hongo.wide.ad.jp

Evaluation System of Wireless Network Condition using Raspberry Pi

Shintaro TANAKA[†], Masaki TAGAWA^{††}, Taketo TAKASHIMA, Takao KONDO^{†††}, and Yoshiaki

KITAGUCHI^{††††}

[†] The University of Tokyo ^{††} Kyushu Institute of Technology ^{†††} Keio University ^{††††} Kanazawa University

E-mail: †tanaka@hongo.wide.ad.jp

1. 背景と目的

ユーザにインターネット接続性を提供するネットワークを運用する際、ネットワーク監視サーバを設置し提供ネットワークが正常に稼働しているかどうかを継続的に確認する事が多い。ネットワーク監視サーバは提供しているネットワークで障害が発生した際に迅速に障害の発生を通知し、その後の障害の原因の特定を容易にしたり、サービスの復旧までの時間を短縮するのに役立つ。サービスの死活監視を目的としてサーバに導入するツールには Nagios や Hinemos などが存在する。これらのソフトウェアをインストールしたサーバから ping などでインターネット疎通性を監視したり、DHCP サーバや DNS サーバなどネットワーク提供に必要な他のサーバのサービス稼働状況の監視が可能である。

しかし、このような監視手法では、実際のサービス利用者端末からのサービス利用における障害点を求めることは困難である。具体的に HTTP によるウェブサービスを利用する場合を考えると、以下のような障害点を挙げる事ができる。

- 物理的な接続性の問題
- IP 的な疎通性の問題
- 名前解決が正しくできない問題
- サービス稼働における問題

このように、それぞれのレイヤにおける障害を検出することが重要であり、無線アクセスネットワークなどの Association 障害を検出するためにも実際のサービス利用者視点での監視が必要と言える。本稿では、イベントネットワークなどで利用される無線アクセスネットワークを提供する際に必要となる、サービス利用の疎通性を継続的に監視する手法について述べる。ま

た、評価システムを実装し、実環境における検証により得られた考察をまとめる。

2. 設計と実装

以上の背景を踏まえ、我々は以下の様なシステムを設計し、実装した上で実際にイベントで提供した無線アクセスネットワークの状態の監視・評価を行った。

2.1 システム要件

このシステムに対する要求として、まず監視端末のデプロイが容易であることが挙げられる。我々が携わったイベントの会場は広く、様々な場所でイベント参加者が無線ネットワークへ接続することが想定された。このため、これらの複数の場所にテスト端末を設置し、無線ネットワークの状態監視を行う必要があった。このように、会場の規模に応じて多数のテスト端末を設置するためにはテスト端末の初期設定が簡単である必要がある。

また、今回のイベントにおいては運用中にネットワークの構成が変更されることがある。このためテスト端末が実行するテストケースを変更できるのが好ましい。つまり、テスト端末の再設定を行わなくても容易にテストケースの定義変更が可能であることがシステムに対する要求として挙げられる。テストケースは TCP/IP の階層モデルにおける階層を下位から順にテストし、障害が発生した際に障害の原因となっている階層を特定しやすくする必要がある。

さらに他のネットワーク計測システムにも言えることだが、監視対象の状態をひと目で確認可能なように可視化し、障害発生時にはアラートメールなどでネットワーク管理者に迅速に通知する機能も必要である。

2.2 アーキテクチャ

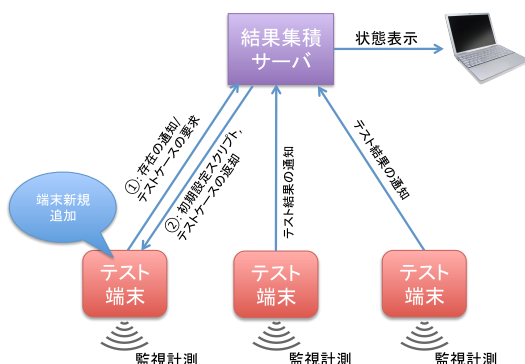


図1 アーキテクチャ概要

これらの要求を踏まえて今回構成したシステムの概要図を図1に示す。テスト端末は小型なシングルボードコンピュータである Raspberry Pi を利用する。これに USB 式の無線 LAN アダプタを接続して無線ネットワークの接続性に関するテストを実行する。Raspberry Pi は非常に安価なため、多数のテスト端末を配置する際のコストが抑えられる。

テスト端末は計測用の無線インターフェースの他に有線のインターフェースを持ち、有線で常時インターネット接続性のあるネットワークへ接続する。テスト端末がデプロイされた場合にはこの有線ネットワークを通じて結果集積サーバからテスト端末の初期設定シェルスクリプトを取得する。

端末の初期設定シェルスクリプトによって Raspberry Pi を構成することで、Raspberry Pi の起動 SD カードには個別の設定内容を不要とし、複製を簡単に行えるように工夫している。

しかし Raspberry Pi は電源システムの堅牢性が十分とはいえないため、Linux カーネルをカスタムし、起動 SD カードを読み取り専用としながら、RAM ディスクを併用することで、システムに対して読み書き可能な記憶領域を提供するようにする。これによって、突発的な電源断が起こっても起動 SD カードのファイルシステムが壊れることはない。

初期設定の完了したテスト端末は自らの存在を結果集積サーバに通知し、テストケースを取得する。テストケースは、識別 ID、テストコマンド、テストコマンドが正常終了した時の返り値、タイムアウト値によって定義されている。

その後、テスト端末はテストケースを順次実行する。テストコマンドがテストコマンドが正常終了した時の返り値と等しければ成功と見なし、コマンド実行時間がタイムアウト値を超えたとき、もしくは異なる返り値を得た場合には失敗と見なす。テストケースが完了した端末は、有線ネットワークを通じて結果

集積サーバにテスト結果を通知する。

またテスト端末は定期的にテストケースの更新をチェックし、更新がある場合には新しいテストケースをダウンロードしてすることでテストケースの動的な更新にも対応する。

結果集積サーバではテスト端末から通知された各テストケースに対する結果をデータベースに集積している。集積したデータは JSON 形式の Web API を通じて Web ページに結果を可視化している。

2.3 計測シナリオ

テストケースとしては以下のような項目を定義し、繰り返しテストを実行した。

- 特定 SSID に対しての無線 Association
- DHCP によるアドレス/デフォルトルート取得
- デフォルトゲートウェイへの ping
- DNS による名前解決
- 外部サイトへの ping
- 外部サイトへの Web アクセス
- 接続中の SSID からの Deassociation

イベントで提供している各 SSID に対して、これらのテストケースを実行した。また IPv4/IPv6 のデュアルスタックでのインターネット接続性を提供していたため、IPv4/IPv6 両方でテストを実行した。このテストケースは TCP/IP の階層モデルでの階層ごとに段階的なテストを行う構成になっており、また下位レイヤでのテストで障害が検知された場合にその上位レイヤのテストを実行しないこととすることで、障害が発生した際にも障害の原因を特定しやすくなっている。

3. 考察と今後の課題

実際に構築した監視システムを運用している中で、Raspberry Pi が熱により強制的に再起動してしまうという症状が見られた。このため運用の中では Raspberry Pi のケースの蓋を開けた状態で監視を続けるという対処を行ったが、Raspberry Pi のチップにヒートシンクを取り付けるなどの適切な熱対策が必要だと考えられる。

またそれ以外にもユーザの PC では正常にインターネットへの疎通性があるのに Raspberry Pi ではエラーが起きているといったテスト結果が報告されることがあった。これについては今後システムの監視ログと他の箇所でも収集したログを突き合わせて監視が正しく行えているかの評価が必要である。

今回はテスト結果を OK か NG でしか報告していないが、テストスクリプトの実行時に出力された結果詳細も監視サーバに集積し、より詳細に障害の状況の解析を行えるようにしたい。