リアルタイムトラフィック可視化システムとその拡張性について

吉田和幸　宇野秀亮　池部実　吉崎弘一
大分大学　大分大学　大分大学　大分大学
yoshida@oita-u.ac.jp

要旨

ソフトウェア開発の初期段階においてプロトタイプを組み立てて実際に動かしてみる作業は有用である．インターネットトラフィックの増大に伴い，サーバやそこで動作しているサービスを調査するスキャンも増加している．スキャン活動のいろいろな側面をリアルタイムに表示するシステムを作成した．宛先 IP アドレスの下位2オクテット，宛先ポート番号など 16 ビットを取り出し，それを基に分類し，それぞれのパケット数をカウントして，256×256 に展開し，全体の分布を表示している．これにより，ネットワークトラフィックの概要の把握が容易になる．パケットヘッダの中の 16 ビット幅のいろいろな場所について分布を表示できるように容易に拡張できるように設計した．

1. はじめに

インターネットの普及に伴い，インターネットのトラフィックも増大している．「我が国のインターネットにおけるトラヒックの集計・試算」[1]によると，2020 年 11 月時点で日本における固定系ブロードバンド契約者の総ダウンロードトラフィックは約 19.8Tbps であり，前年同月比で 56.7%増加している．また，ブロードバンドサービス契約者の総アップロードトラフィックは約 2.4Tbps であり，前年同月比で 51.1%増加している．これに伴いサーバやそこで動作しているサービスを調査するスキャンも増加している．

個々のパケットや，サーバ・クライアント間の一連のパケット(フロー)を詳細に調べることは，不正侵入等の検知には重要であるが，大量のパケットが来るスキャン等の傾向を把握するには，個々のパケットやフローの解析では困難である．宛先 IP アドレスの第 3，第 4 オクテットやポート番号など，パケットヘッダ中の 16 ビットの箇所の分布をリアルタイムに表示するシステムを設計・試作した．

本論文では，2 章で関連研究について述べ，3 章でシステム構成について述べ，ラピッドプロトタイピングの観点からプログラム構成について述べる.4 章で実行例を示して，5 章でまとめと今後の課題について述べる．

2. 関連研究

NICTER (Network Incident analysis Center Tactical Emergency Response)[2]は，サイバー攻撃観測・分析・対策システムである．ダークネットを観測することで収集したパケットデータを解析している．ダークネットとは，インターネットに向けて公開されている IP アドレスのうち，未使用のアドレスによって構成されたネットワークであり，通常の通信では，存在するホストに対してパケットを送信するため未使用のアドレス空間宛に通信が発生することはない．そのため，ダークネットで観測されるすべてのトラフィックは不正な通信である可能性が高い．ブラックホールモニタリングと呼ばれる，受信されるパケットに対して一切応答せず，記録する手法を取っている．NICTER による観測・分析結果の一部は，NICTERWEB 2.0[2]によって一般公開している．NICTERWEB 2.0 ではダークネットトラフィックを，立方体にマッピングした「Cube」や，地図上に表示した「Atlas」によって，リアルタイムに可視化している．

新川ら[3]は IP アドレスの下位 8bit，ポート番号の下位 8bit を用いた 2 次元マトリクスを表示する手法を提案，実装した．2 次元マトリクス上の点をクリックすると，その通信に関する IP アドレス，ポート番号，時間，プロトコル，方向が表示される．しかし，下位 8bit が重複するホストの通信が複数ある場合，重なって表示されてしまうため，ホストごとの正確な通信状況を把握できない．

宇都木ら[4]は TCP コネクションごとのトラフィックを帯状に表示し，リアルタイムなトラフィック量によって色のグラデーションで示すことでトラフィックを可視化した．ポート番号や IP アドレスによってトラフィックをソートしたり，フィルタしたりできる．トラフィックの可視化に特化しており，パケットの宛先の分布などを把握できない．

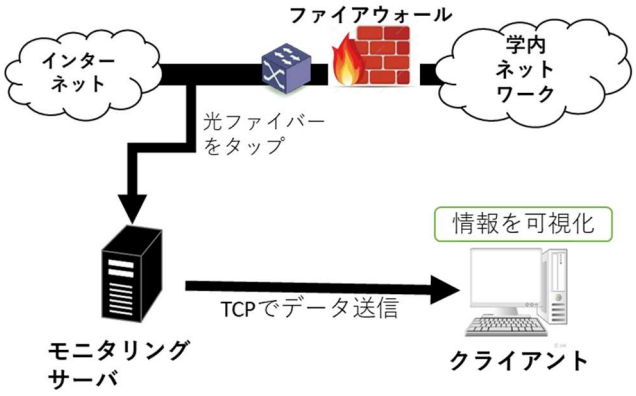


図 1　システム構成

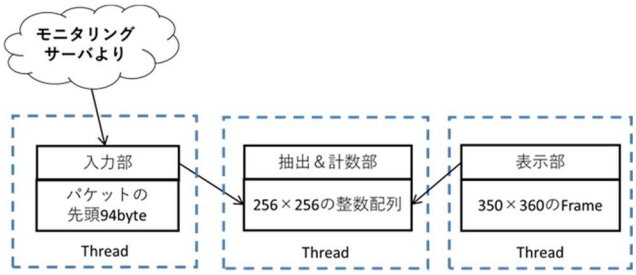


図 2　プログラム構成

3. リアルタイムトラフィック表示システム

3.1. システム構成

図 1 にシステム構成を示す. インターネットへのアクセス線(10Gbase-SR)×2 の光ファイバをタップにより分岐して, モニタリングサーバに送る. モニタリングサーバでは, tcpdump コマンド[5]によりインバウンド(外から内)パケットのヘッダ情報を収集し, ncat コマンド[6]を用いて, クライアント PC へ送る. クライアント PC では, Java プログラムにより, データを読み込み, 必要な部分を抽出・計数し, 結果を表示する. Java プログラムについては次節で詳述する.

3.2. プログラム構成

プログラムの構成を図 2 に示す. モニタリングサーバから毎秒 30000 個ほどのパケットヘッダが送られてくる. 入力部は, 独立した thread とし, 入力したパケットヘッダ情報をパラメータに抽出＆計数部のメソッドを呼び出し, 計数させる. 抽出＆計数部では, 計数結果がオーバフローしないように, 指数平滑移動平均を計算する thread を 10 秒ごとに動作させている.

```
protected int d[][]=new int[256][256];
public boolean cnt(byte b[]) {
        if(b[23]==1)                    // protocol=ICMP
            d[b[32]&0xff][b[33]&0xff]++;
                // 宛先IPアドレスの３，４オクテット
        return r;
}
public int getd(int x,int y) { return d[x][y]; }
```

図 3　抽出＆計数部のメソッド例

図 3 に, 抽出＆計数部の計数するメソッド(cnt)と, 結果を表示部へ渡すメソッド(getd)を示す. cnt メソッドは, パケットの先頭 94byte(イーサヘッダ 14byte, IP ヘッダ 20～60byte, TCP ヘッダ 20byte) を受け取り, 計数したいパケットを抽出し, 計数する. この例では, icmp パケットの, 宛先アドレス(学内ネットワーク側のアドレス)の第 3, 第 4 オクテットで計数している. getd メソッドは, 2つのインデックス値をもらい, 対応する計数結果を返している.

表示部は, 抽出＆計数部から計数結果の 256×256 の整数配列をもらって, 計数結果を色で表して, 256 dot ×256 dot の散布図として表示する. 図 4 **エラー! 参照元が見つかりません。**に計数結果から色への対応を示す. 計数値を白(0)～青(4)～水色(8)～黄色(16)～赤(32)～マゼンタ(64)～紫(128)～黒(255)へマップしている(()内の数は計数値). 値が小さいところでは, 早く変化し, 値が大きいところでは, ゆっくり変化するようにしている. 図 5 に表示例を示す. 抽出している 16 ビットのうち上位 8 ビット(0～255)を縦軸に, 下位 8 ビットを横軸にして 256×256 のマトリックスで表示し, 上下端, 左右端に 8 ごとに目盛りを表示している. この例では, インバウンドパケットの宛先 IP アドレスの第 3 オクテットを縦軸に示し, 第 4 オクテットを横軸に示す. なお, 縦軸では, 上端が 0, 下端が 255, 横軸では左端が 0, 右端が 255 である.

cnt メソッドの抽出・計数条件を変更することにより, ポート番号, 宛先アドレスの第 3, 第 4 オクテット, 送信元アドレスの第 1, 第 2 オクテット, 宛先アドレスの第 4 オクテットとポート番号の下位オクテット[3]など種々のフィールドを, 計数することができる. 入力部から複数の抽出＆計数部を呼び出せるようにして, 新たなフィールドの計数を容易に組み込めるようにしている(図 6).

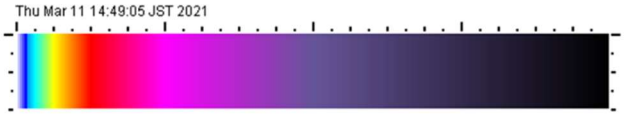


図 4　カラーマップ

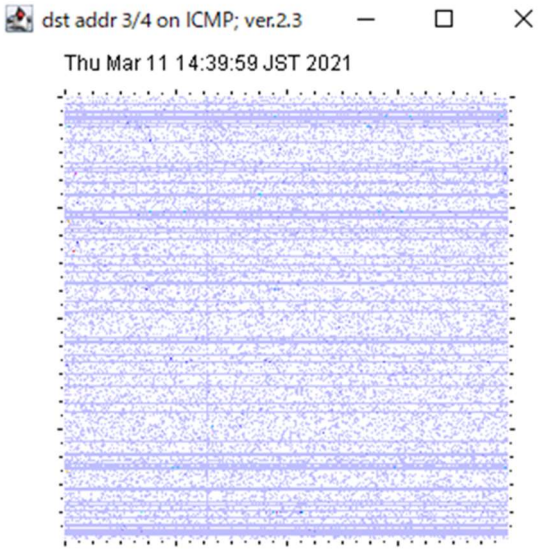


図 5 ICMP パケットの宛先アドレスの分布

3.3. 実行例

実行結果を図 5 のほかに，図 7～図 15 に示す．これらは，本システムを 52 分間実行した結果であり，モニタリングサーバ上で動かしている tcpdump コマンドによると 84,750,410 パケット受信し，233,253 パケット(0.3%)取りこぼしている．10 秒ごとに値を半分にする指数平滑移動平均を計算しているため，おおむね 20 秒間のパケット到着数を示している．

図 7 は，インターネットから学内LANに入ってくるすべてのパケット(インバウンドパケット)の宛先 IP アドレスの第 3，第 4 オクテットの分布である．学内 LAN では，/16 の

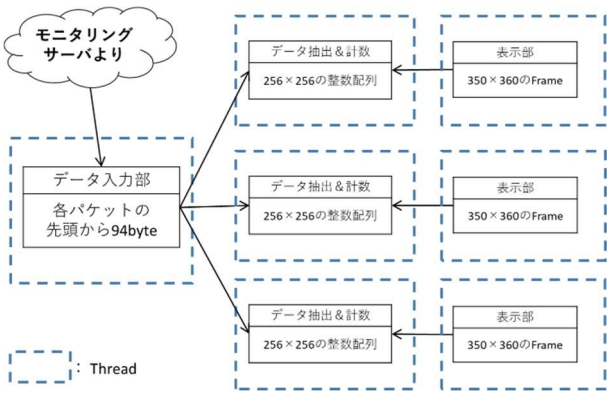


図 6 プログラムの拡張

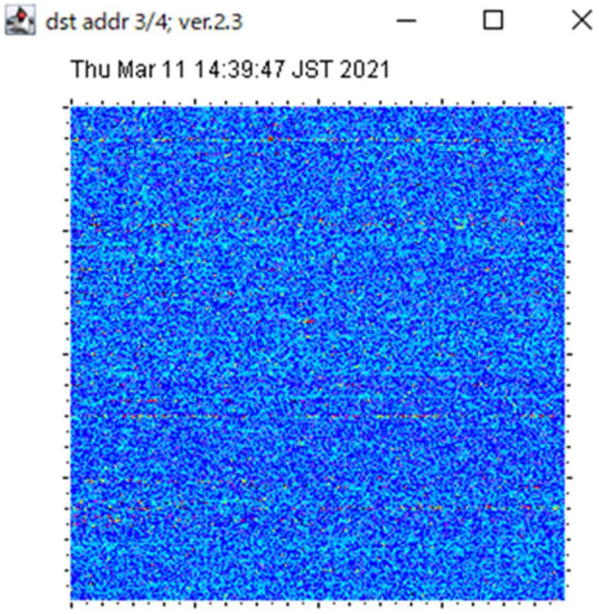


図 7 宛先 IP アドレスの分布

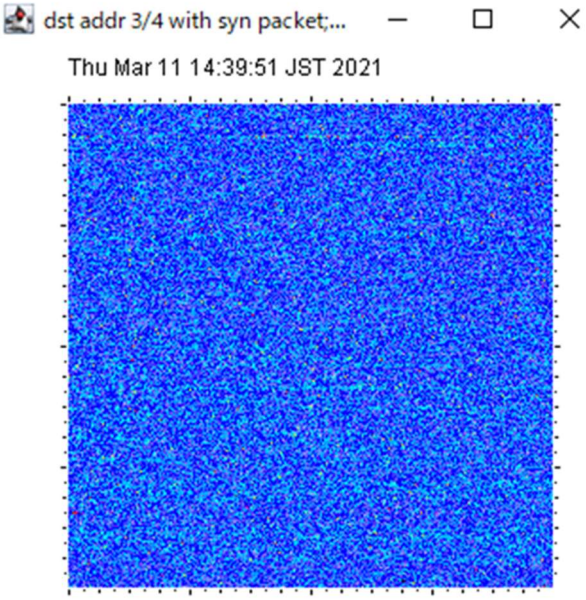


図 8 SYN パケットの宛先アドレスの分布

IP ネットワークを利用しているので，学内 LAN のアドレス空間すべてを表現できる．

図 8 は，インバウンドパケットのうち，TCP のコネクション確立要求である SYN フラグのみ立っているパケットの宛先 IP アドレスの第 3，第 4 オクテットの分布である．図

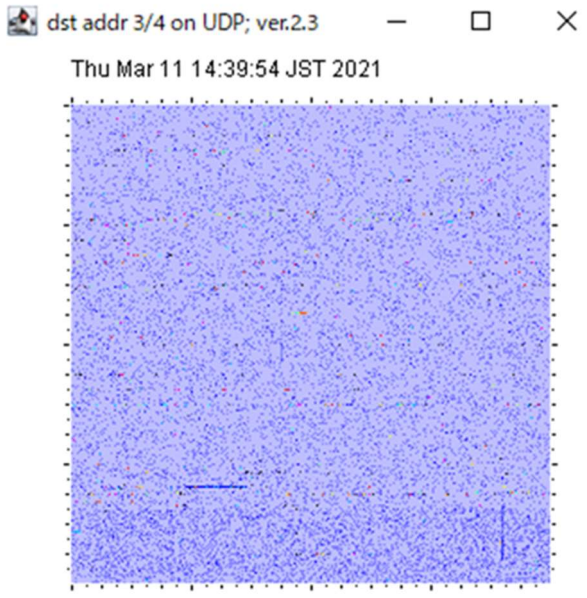


図 9 UDP パケットの宛先 IP アドレスの分布

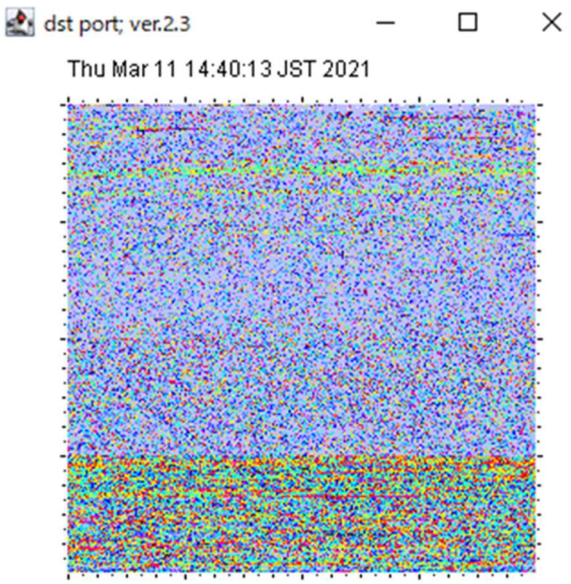


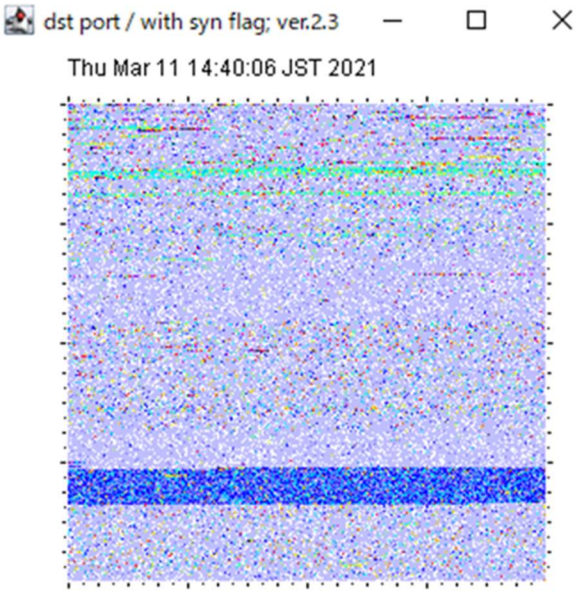
図 10 ポート番号の分布

6 と図 7 とで，図 6 のほうが計数値が大きい(水色)が，大きな違いはない．スキャンの大部分は，TCP スキャンであることがわかる．

図 5，図 9 は，インバウンドパケットのうち，ICMP，UDP パケットの宛先 IP アドレスの第 3，第 4 オクテットの分布である．図 9(UDP)では，アドレス空間全体をスキャンされてはいるが，短い横線と縦線が見える．第 3 オクテットを固定して，第 4 オクテットを変化させるスキャンと，第 4 オクテットを固定して第 3 オクテットを変化させるスキャンが行われたことがわかる.図 9(ICMP)は，図 3 で示したプログラムで集計したものである．横線が目立つが，少数の縦線も確認できる．

図 10 には，すべてのインバウンドパケットのポート番号の分布，図 11 には TCP の SYN フラグのみ立っているパケットのポート番号の分布である．図 10 では，ポート番号の上位バイトが 192 以降の部分が多くなっている．これば，学内の PC が外部のサーバにアクセスするときのエフェメラルポートであろう．この影響を排除するため，外部からの TCP コネクション要求に絞ると図 11 になる．上位バイトが 32～40 付近(ポート番号 8000～10000)と 192～216 付近(ポート番号 49000～55000)で，スキャンが多いことがわかる．

文献[3]にならい，図 12 に，すべてのインバウンドパケットの宛先 IP アドレスの第 4 オクテットとポート番号の下位バイトの分布を，図 13 に，TCP の SYN フラグのみ立っているパケットの宛先 IP アドレスの第 4 オクテットとポート



番号の下位バイトの分布を示す．図 12 で縦線が目立つ

図 11 SYN パケットの宛先ポート番号の分布

のは，図 10 と同様にエフェメラルポートの影響である．それを排除した図 13 では，横線(ポートを固定して，IP アドレスを変化させる)が目立っている．うっすらではあるが縦線(IP アドレスを固定して，ポート番号を変化させる)のスキャンも見える．

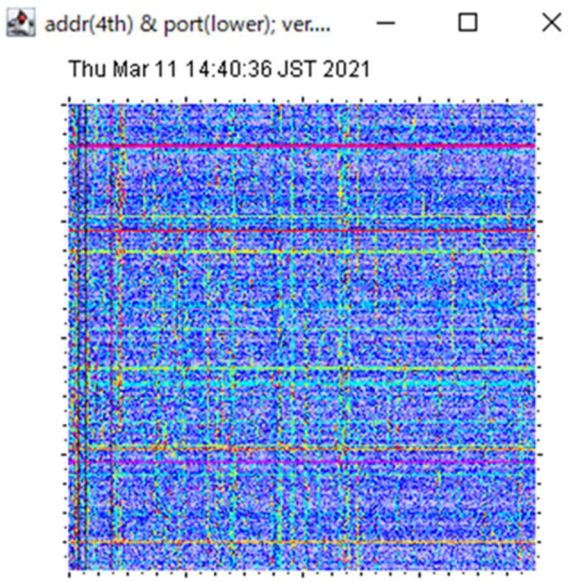
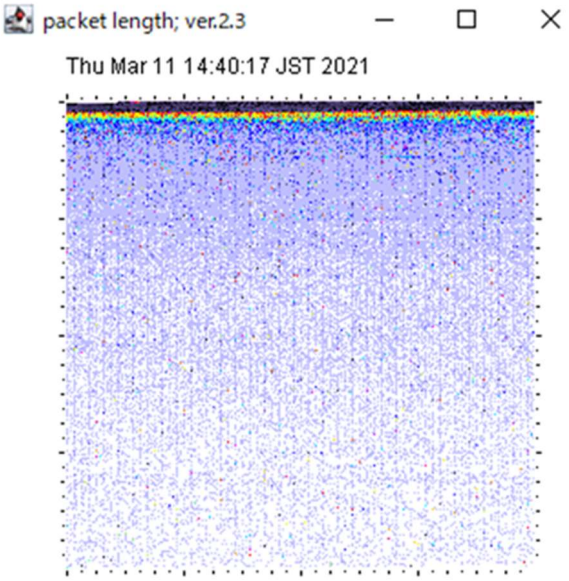


図 12 宛先アドレスとポート番号の分布

図 14 にパケット長の分布を示す. イーサネットの最大パケット長は 1500 であるが, モニタリングサーバのインターフェースカードのハードウェアオフロード機能により, TCP パケットが最大 window size になるまで結合された長大なパケットを観測している. 長大なパケットの長さが, 8 の倍数に集中しているので, この時は, Window size 拡張オプションが 3 (8 倍)の TCP コネクションで, 大量のデータの受信を行っていたことがわかる.



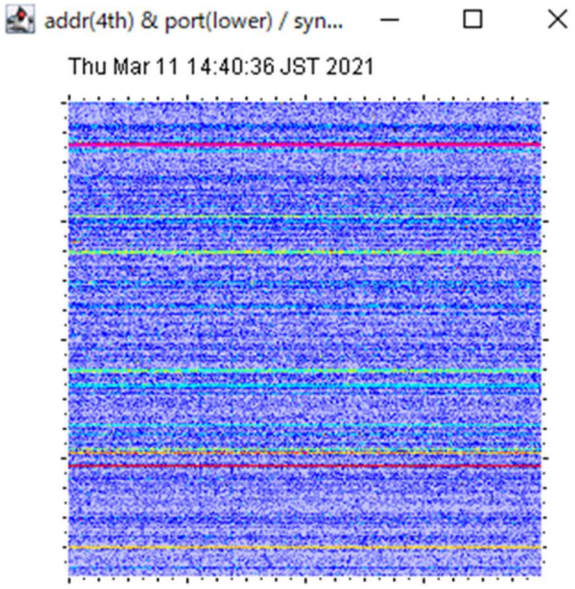


図 14 パケット長の分布

4. おわりに

パケットヘッダ情報を解析し，パケット数を計数することにより，スキャン活動の宛先 IP アドレスの分布，ポート番号の分布等をリアルタイムに可視化するリアルタイムトラフィック可視化システムについて述べた. その実行例から, **ポート番号の分布では，8000～10000，49000～55000 の 2 か所に集中していること，宛先 IP アドレスは第 4 オクテットを変化させるスキャンが大部分であることなどがわかった.**

パケットの新たな分布を調査したいとき容易に拡張できる. 本システムを使い, 新たな分布図の作成を試行することにより, ネットワークの運用管理, セキュリティ監視にさらに有用なものを見つけたい. .

参考文献

[1] 総務省, “我が国のインターネットにおけるトラヒックの集計・試算”, https://www.soumu.go.jp/menu_news/s-news/01kiban04_02000182.html, 2021.2

[2] nicter, https://www.nict.go.jp/, 国立研究開発法人情報通信研究機構,閲覧日: 2021 年 2 月 4 日.

[3] 新川拓也,山之上卓,IP アドレスとポートによる二次元平面を用いた通信トラフィックの可視化について, 情報処理学会研究報告, Vol.2006,No97,pp.31-

36,2006 年 9 月 15 日.
[4] 宇都木進,渡邉晶,TCP コネクション単位でトラフィックの可視化を行うツールの開発,情報処理学会研究報告,Vol.2009-IOT-7,No.4, pp.1-5,2009 年 10 月 9 日.
[5] tcpdump コマンド, https://www.tcpdump.org
[6] ncat コマンド, https://nmap.org/ncat/