

組織内ネットワークのトラフィックの可視化と 異常検出支援ツールの開発

脇村 亜衣^{1,a)} 青木 茂樹^{1,b)} 宮本 貴朗¹

概要：ネットワーク管理者は、組織内ネットワークの状態を正しく速やかに把握することが求められている。本研究では管理者を支援するため、組織内ネットワークのトラフィックを可視化するツールを開発した。本ツールではサブネット間のトラフィックを、三次元のアニメーションで単位時間ごとに地図上に描画する。また、カルマンフィルタを用いて、各サブネット間やポート番号ごとのトラフィック量などの特徴量の遷移を推定し、推定値と実測値が大きく異なった場合を変化点として検出する。そして、変化点を検出した際にはアニメーションでの描画に加え、変化を検出した特徴量のグラフを提示する。管理者はツールの出力を確認することで、トラフィックに発生した異常を容易に把握することができる。実験では大阪府立大学のキャンパスネットワークに、本ツールを適用して有効性を確認した。

キーワード：トラフィックの可視化, 変化点検出, 組織内ネットワーク

Development of Tool for Traffic Visualization and Abnormality Detection in Intranet

AI WAKIMURA^{1,a)} SHIGEKI AOKI^{1,b)} TAKAO MIYAMOTO¹

Abstract: Network administrators are required to grasp status of Intranet correctly and quickly. In this research, in order to support network administrators, we developed a tool to visualize traffic of Intranet. This tool express traffic between subnetworks by animating three-dimensionally on map per unit time. Also, it uses Kalman filter to detect change points of the traffic volume. This tool detects change point when estimated value and measured value are largely different. When change point is detected, the traffic volume detected change point is outputted as graphs, in addition to drawing traffic by animation. By checking the outputs of this tool, network administrators can grasp the transition of traffic easily. For experiment, we checked effectiveness of this tool by using the campus network of Osaka Prefecture University.

Keywords: Network Visualization, Change Point Detection, Intranet

1. はじめに

近年、我々の利用するネットワーク環境は複雑化を続けている。ネットワーク管理者は組織内のトラフィックの状況を迅速かつ正確に把握することが求められるが、膨大なデータの分析は文字と数値だけでは困難である。そのため

昨今、ネットワークトラフィックを視覚的に把握可能な可視化システムの開発が進んでいる。可視化システムは、疎通確認や障害検知、輻輳の把握や設定ミスの検出などを支援するために用いられ、可視化システムを導入することにより、組織内ネットワーク管理の効率化が期待される。

ネットワーク可視化の代表的なシステムとして、独立行政法人情報通信研究機構 (NICT) が開発した NICTER[1] と NIRVANA 改 [2] が挙げられる。NICTER は、広域ネットワークにおけるセキュリティインシデントの迅速な状況把握と原因追究を目的として開発された。ダークネットに

¹ 大阪府立大学 大学院人間社会システム科学研究科
Graduate School of Humanities and Sustainable System Sciences, Osaka Prefecture University

a) swa01331@edu.osakafu-u.ac.jp

b) aoki@kis.osakafu-u.ac.jp

届くパケットを三次元空間または世界地図上に可視化するだけでなく、観測したマルウェアの分析を行い、駆除ツールの自動生成と配布などを実行する。NIRVANA 改は、組織内ネットワークを表現した画面上にライブネットのトラフィックを可視化することに加え、各種セキュリティ機器からのアラート集約を行うサイバー攻撃統合分析プラットフォームである。

本研究では、管理者がネットワークの状況を適確に把握するために有用な、組織内ネットワークのトラフィックの可視化と異常検出を支援するツールを開発した。このツールでは、sFlow を用いて取得した情報を用いて、サブネット間のトラフィックを単位時間毎に地図上へ三次元的にアニメーションで描画する。また、アニメーションを常時監視していなくてもトラフィックに変化があった際には管理者に通知できるよう、カルマンフィルタを用いた変化点検出を行う。変化点を検出した場合にはグラフで出力し、アニメーションで確認したい場合は後からでもデータを参照して再生することができる。

管理者は、本ツールの出力するアニメーションとグラフを確認することで、トラフィックの変化の状態を把握することができる。また、組織内ネットワーク内にマルウェアに感染した端末が存在する場合などには、通常とは異なるネットワークの状態にあることを容易に判断できる。なお、本ツールでは出力に関する設定を GUI で行えるようにしており、管理者は直感的な操作で描画する情報を取捨選択することができる。

2. 既存の可視化手法

ネットワークの可視化に関する既存手法として、The Multi Router Traffic Grapher(MRTG)[3] や RRD-TOOL[4], IP Matrix[5] などが提案されている。MRTG と RRDTOOL は個々の端末におけるネットワークの利用状態を確認するためには有効であるが、ネットワークトラフィック全体を把握することは難しい。また、IP Matrix はサイバー攻撃の一種であるマルウェアの検出に焦点を絞った可視化手法であるため、組織内ネットワークのトラフィック全体の可視化を目指す本研究の手法としてそのまま適用することは難しい。

また、F. Mansman らの研究 [6] ではホストとサーバの活動状況の可視化を目的としたシステムの構築がなされており、R. Ball らが開発した VISUAL[7] では内部ホストと外部ホスト間の通信の可視化が行われている。これらの研究は主に各ホストの負荷や通信状況を提示するものであり、ネットワークトラフィックにおける異常の自動検出には至っていない。J. McPherson らが開発した PortVis[8] ではポート番号ごとの通信を可視化しており、ウイルス、トロイの木馬、ワームのような悪意のあるプログラムの特徴がポート番号に現れた場合の異常検出に有効である。ただ

し、IP アドレス・ポート番号の情報量が膨大になるため、可視化する要素を取捨選択する技術が必要であると考えられる。また、Y. Livnat らの研究 [9] では異常や攻撃を可視化するシステムが開発されているが、後述する NIRVANA 改 [2] と同様に各種セキュリティ機器からのアラートを集約して可視化するものであり、異常をシステム内部で検出している訳ではない。さらに [6]~[9] の研究では各情報を図形によって可視化しているため、その意味を直感的に読み取ることが難しくなっている。

ネットワーク全体の状況を可視化する代表的なシステムとして、NICTER[1] と NIRVANA 改 [2] について概説する。

2.1 NICTER

NICTER とは、インターネット上で時々刻々と発生しているセキュリティインシデントへの迅速な対応を目指したサイバー攻撃観測・分析システムである。ダークネットと呼ばれる未使用 IP アドレス空間に届くパケットを三次元空間または世界地図上に表示するとともに、その原因と考えられるマルウェアを特定し、検体の分析を行い、駆除ツールの自動生成と配布などを実行する。この手法はダークネットの観測と可視化に最適化されている。

2.2 NIRVANA 改

NIRVANA 改は、組織内ネットワークを流れる通信のリアルタイムでの観測・分析や、各種セキュリティ機器からのアラート集約を実現するサイバー攻撃統合分析プラットフォームである。可視化システムは NICTER を応用しており、リアルタイムに可視化された組織内ネットワークの中から、サイバー攻撃に関連した異常な通信を検知し、その通信の送信元の直上にアラート表示する機能を有している。アラート表示はファイアウォールや侵入検知システムなどの各種セキュリティ検知・防御システム (FireEye[10] や A10 Thunder ADC[11] など) からの警告を集約することによって実現している。

3. 提案手法

本研究では、ネットワークトラフィックの可視化と異常検出を同時に行い、またトラフィックの状況を実際の地図上に描画することで直感的な把握が可能となる可視化・異常検出システムを提案する。図 1 は、本ツールによって再現した異常なトラフィックが発生した際の描画例である。複数のサブネットから特定の棟に割り当てられたサブネットに向けて DDoS 攻撃が発生しており、赤色の大きな立体が連続的に対象となる棟へ斜方投射されている。このように視覚的に情報を得られる他、カルマンフィルタによって変化点を検出して提示することで、管理者がネットワークの異常を速やかに確認できるよう支援する。また、本ツ

表 1 表現方法と可視化パラメータ

表現方法	可視化パラメータ
トラフィック単位	*IP アドレス&ポート番号, IP アドレス, サブネット
出発点と着地点	送信元サブネットと宛先サブネット
立体の種類 (立方体/球体)	TCP/UDP
立体の色	*送信元サブネット, 宛先サブネット, 単位時間あたりの合計パケットサイズ, 単位時間あたりのパケット数, カルマンゲイン各種
立体の大きさ	*単位時間あたりの合計パケットサイズ, 単位時間あたりのパケット数, カルマンゲイン各種
斜方投射の高さ	*単位時間あたりのパケット数, 単位時間あたりの合計パケットサイズ, カルマンゲイン各種, **送信元ポート番号, **宛先ポート番号

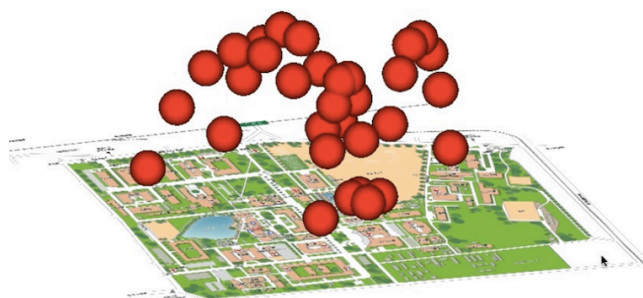


図 1 異常なトラフィックの描画例

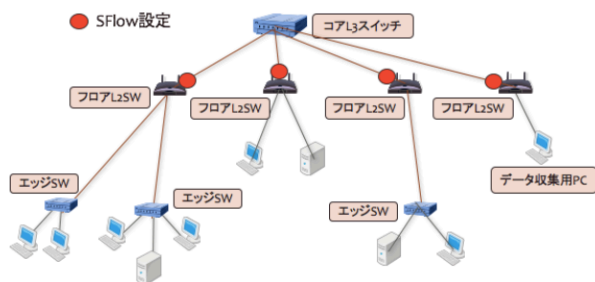


図 2 トラフィックデータの収集

ルは GUI による入力で直感的な操作ができ、管理者が得たい情報に合わせて可視化の表現方法を変更したり、得たい情報のみを抽出した描画が可能である。

3.1 トラフィックデータの収集

可視化対象となるネットワークのフロアスイッチのアップリンクポートにて sFlow を用い、トラフィックデータを収集する (図 2 参照)。sFlow とは、スイッチとルータを含むネットワークトラフィックのパケットをリアルタイムにモニタリングするプロトコルであり、指定したインタフェースにおいて一定のサンプリングレートにてデータの収集を行う。データ構成は観測時刻, 送信元 IP アドレス・ポート番号, 宛先 IP アドレス・ポート番号, 送信元 MAC アドレス, 宛先 MAC アドレス, Ethernet タイプ, プロトコル, フラグ, TOS, TTL, ウィンドウサイズ, パケット長である。

3.2 可視化手法

トラフィックデータから単位時間毎に可視化パラメータを抽出し、情報を画像上の送信元座標から宛先座標に向けて立体を斜方投射するように描画することで三次元的に表示する。トラフィックの状態の表現手法として、パケットの送信元と宛先を示す立体の発着点に加え、各トラフィックの状態を表す立体の種類・大きさ・色、また斜方投射の高さを用いる。各表現方法に対応する可視化パラメータは、表 1 に示す通りである。複数の可視化パラメータが含まれている表現方法については、管理者が GUI による操作で自由に変更が可能である。初期値としては*印の項目を設定している。なお、トラフィック単位を“IP アドレス&ポート番号”以外に設定している場合、**印の可視化パラメータは選択できない。

なお本ツールでは、カルマンフィルタを用いてトラフィックの推定を行い、観測値と推定値の間にどれ程の相違があるのかを示す「カルマンゲイン」を可視化パラメータのひとつとして扱う。次節でカルマンフィルタによる変化点検出手法について述べる。

3.3 カルマンフィルタによる変化点検出

カルマンフィルタとは、誤差のある観測値を用いて、ある動的システムの状態を推定するための、無限インパルス応答フィルタの一種である。本研究ではカルマンフィルタを用いたトラフィック推定を行い、推定値と実際の観測値を比較することで変化点を検出し、変化点を検出した際に管理者へ情報を通知する。

変化点検出の対象は、「各通信における単位時間あたりの合計パケットサイズ, パケット数, 使用ポート番号の種類数, 使用 IP アドレスの種類数 (4 次元)」と「主要なポート番号 16 種類を使用している通信の単位時間あたりの合計パケットサイズ (16 次元)」の 2 種類である。サブネット間の通信と各ポート番号の通信を同時に監視し、それぞれの変化点について情報を得ることで、ネットワークに発生する異常をより容易に特定し把握できる。なお、監視するポート番号は一般的によく用いられる「21(FTP), 22(SSH), 23(TELNET), 25(SMTP), 53(DNS), 80(HTTP), 110(POP3), 123(NTP), 143(IMAP), 161(SNMP), 194(IRC), 443(HTTPS),

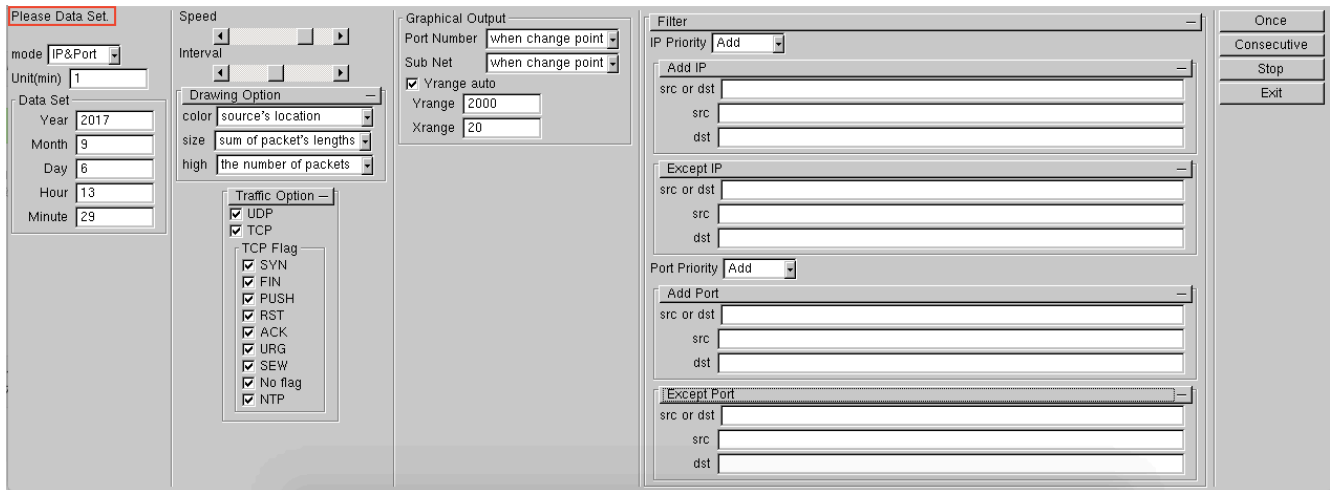


図 3 操作パネル

465(SMTPS), 587(SMTP), 993(IMAPS), 995(POP3S)」の 16 種類とする。

時刻 t において観測した値を Y_t とし、カルマンフィルタによって予測した時刻 t の推定値を X_t とする。以下の線形方程式を用いて、 X_{t+1} と X_t を関連付ける状態方程式と、 Y_t と X_t を関連付ける観測方程式を構築する。

$$\begin{cases} X_{t+1} = C_t X_t + V_t \\ Y_t = X_t + W_t \end{cases} \quad (1)$$

ここで C_t は時間遷移に関する線形モデルであり、 V_t と W_t はそれぞれ測定ノイズと状態ノイズである。これらのノイズは互いに無相関かつ正規分布に従うと仮定する。式 (1) に基づき、時刻 t における X_{t+1} と共分散行列 P は式 (2) のように推定できる。なお、 Q_t は状態方程式のノイズの分散を示す。

$$\begin{cases} \hat{X}_{t+1} = C_t \hat{X}_t + V_t \\ P_{t+1} = C_t P_t + Q_t \end{cases} \quad (2)$$

予測ステップで推定した値 $\hat{X}_{t+1}$ と実際の観測値 Y_{t+1} の組み合わせを使用し、状態の推定値と誤差の共分散を式 (3) で更新する。

$$\begin{cases} \hat{X}_{t+1} = \hat{X}_{t+1} + K_{t+1}(Y_{t+1} - \hat{X}_{t+1}) \\ P_{t+1} = (1 - K_{t+1})P_{t+1} \end{cases} \quad (3)$$

K_{t+1} はカルマンゲインと呼ばれ、次式を用いることで導出できる。なお、 R_t は観測方程式のノイズの分散を示す。

$$K_{t+1} = \frac{P_{t+1}}{P_{t+1} + R_t} \quad (4)$$

本手法では可視化パラメータとして、式 (4) のカルマンゲイン K_{t+1} を用いる。また、 K_{t+1} の各要素が閾値 0.8 以上となった時に変化点を検出したと判断し、対象となるサブネットやポート番号に関するグラフを出力する。変化点発生時刻を時刻 t とすると、時刻 $t-4$ の状態まで遡って

時系列順に並べて出力することで、どのような変化があったのかを管理者に提示する。

3.4 可視化の GUI

管理者にとって分かりやすく直感的な入力と設定を支援し、得たい情報のみを可視化するため、GUI による様々な操作を可能としている。操作パネル (図 3) は可視化画面の下側に表示されており、図の例ではロールアウトが全て開いているが、初期設定では閉じている。以下でツールが実装している機能について解説する。

- (1) “mode” では描画するトラフィックの単位についてリストボックスで設定する。
- (2) “Unit(min)” では単位時間を 1 秒単位でテキストボックスにて設定する。
- (3) “Data Set” では描画するトラフィックの年月日と時刻をテキストボックスで入力する。ツール起動時には現在時刻が自動的に設定される。
- (4) “Speed” と “Interval” はスクロールバーを操作することにより、それぞれ「斜方投射の速度」「立体を投射する間隔」を設定する。これら二つを組み合わせることにより、描画の速さを自由に変更可能である。
- (5) “Drawing Option” は前節で述べた可視化パラメータをリストボックスにより設定する。
- (6) “Traffic Option” は描画する通信の種類をチェックボックスで設定する。TCP 通信は SMB の他、各種フラグによる設定が可能である。初期値では全てがチェックされている。
- (7) “Graphical Output” ではグラフ出力についての設定を行う。変化点検出時でなくとも常にグラフを出力、あるいは常に出力しないよう設定したり、グラフ出力の際の横軸と縦軸の上限値についてもテキストボックスで設定できる。
- (8) “Filter” ではテキストボックスに直接 IP アドレスや



図 4 アニメーションの描画例

ポート番号を入力することにより、指定した通信だけを描画または省く設定を行う。

- (9) “Once” と “Consecutive” は描画を開始するためのボタンである。前者は設定された時刻のトラフィックを一回のみ描画し、後者は設定された時刻から最新の時刻までを連続的に描画する。なおトラフィック描画中は、赤枠で示す箇所の “Please Data Set.” の文字列が描画中のトラフィックの年月日・時刻に切り替わり、指定されたデータが存在しない場合は “Data Set Error.” の文字列が表示される。現在の描画を止めるためには “Stop” のボタンを、ツールを終了させるためには “Exit” のボタンをクリックする。

以上のパネル操作に加え、画面上でマウスをドラッグすることによる画像回転や、キーボード入力による画像の拡大縮小・軸移動の操作ができ、管理者が画像の中で注目したい箇所へ容易に焦点を当てることが可能である。

4. 実験と考察

4.1 実験環境

提案手法の有効性を確認するため、大阪府立大学のキャンパスネットワークにおいて実験を行った。実験に用いたのは 2017 年 7 月 27 日から 2018 年 1 月 10 日の期間に sFlow で採取されたデータであり、サンプリングレートは 1/512 パケットである。

4.2 実験結果

可視化結果の例を、図 4～図 14 に示す。図 4 では、描画の各表現方法は表 1 で述べた初期値の設定である。この場面では 4 箇所のサブネットから送信されたトラフィックが観測されているため、合計 4 色の立体が描画されている。一番上に位置する赤色の球体で示される通信は合計パケットサイズ・パケット数ともに他の通信と比べて大きかったため、図中において他の立体よりも大きく、斜方投射の高さも高く表現されている。このように、初期値ではトラフィック量の大きな通信ほど目立って描画される。

図 5 は、図 4 と同時刻のトラフィックを表現方法を変更して描画した図である。立体の色をパケット数、大きさ

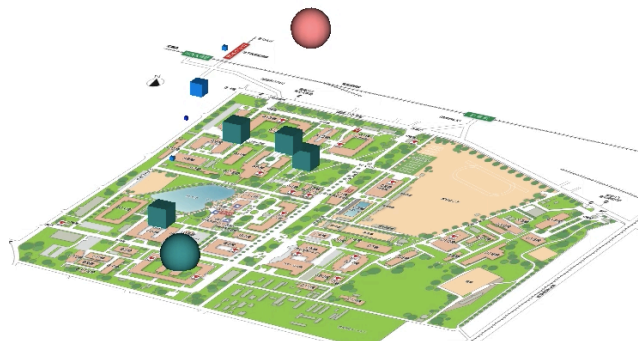


図 5 表現方法の変更例

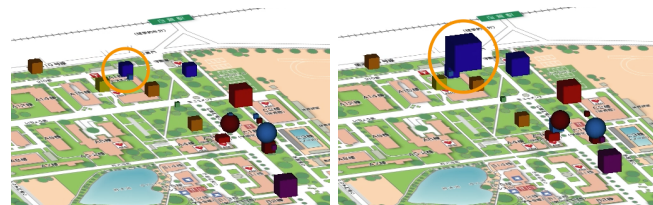


図 6 トラフィック単位別の比較例

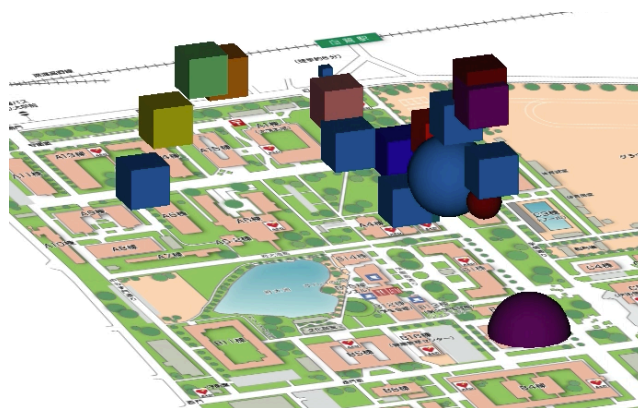


図 7 単位時間を変更した描画例

を合計パケットサイズ、斜方投射の高さを送信元のポート番号に設定した。図 4 では一番上に位置していた赤色の球体で示される通信がここでは桃色となっており、他の通信はパケット数が少なかったため寒色となっている。また、斜方投射の高さがポート番号の数字の大きさに比例するため、様々な高度で立体が飛び交っていることから様々なプロトコルが利用されていることが分かる。

図 6 では、同一時刻・同一単位時間 (1 分) の状態についてトラフィック単位を変更して描画している。左図が IP アドレス & ポート番号で設定しているのに対し、右図は IP アドレスで設定しているため、同一 IP アドレス間の通信であれば使用ポート番号に関わらずひとつに纏めて描画される。そのため、図の橙色の丸で示した立方体が右図では大きくなっており、該当 IP アドレスの通信が複数のポート番号を用いて多く発生していたことが分かる。

図 7 では、単位時間を 100 分とし、トラフィック単位をサブネットに設定している。同一サブネット間の通信であれば IP アドレス・ポート番号に関わらずひとつに纏めて

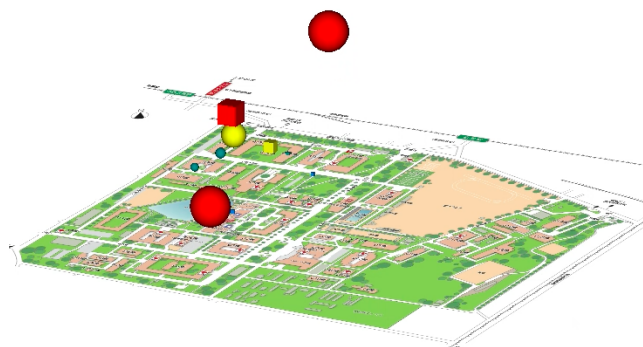


図 8 ネットワーク障害発生時 (11:21) の描画例



図 9 ネットワーク障害発生時 (11:27) の描画例

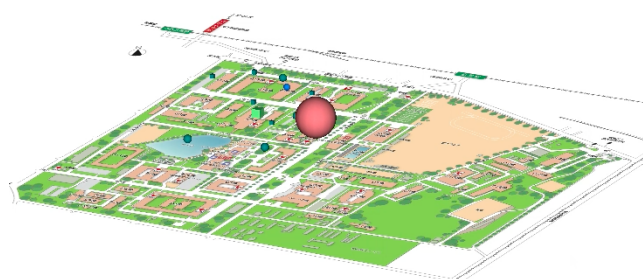


図 10 ネットワーク障害発生時 (11:33) の描画例

描画されるため、1時間を超える単位時間のトラフィック量であってもサブネット単位で短く簡潔に可視化できる。

図 6～図 7 のように、単位時間を短く設定しポート番号まで分類して詳細に可視化したり、その中でもホスト間の通信に注目したい場合は IP アドレス単位で可視化したり、長めの単位時間に対してはサブネット単位で組織単位の可視化を行うなどの設定を適用することにより、一度に得られる情報量を管理者がいつでも自由に決定・変更できる。

また 2017 年 8 月 17 日のデータにおいて、11 時 11 分と 11 時 18 分から特定のサブネット間のトラフィック量が膨大になり、その後全てのサブネット間の通信が 11 時 24 分～11 時 31 分まで途絶えるという異常をカルマンフィルタによって検出した。11 時 31 分以降は通常のトラフィック量に戻っている。図 8～図 10 まではアニメーションによる再現の様子であり、図 8 は 11 時 21 分、図 9 は 11 時 27 分、図 10 は 11 時 33 分の描画である。これらの描画は、その時間帯にネットワーク障害が発生し全てのネットワー

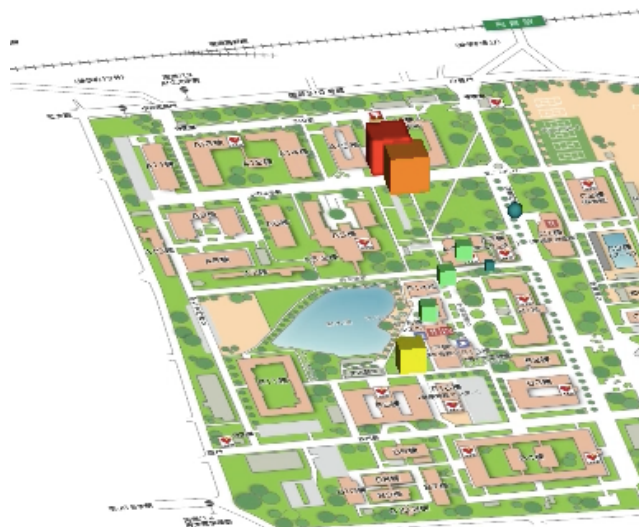


図 11 変化点検出の描画例

クが接続不可能となり、やがて回復したという状況を表現している。また、11 時 10 分～11 時 40 分までの 30 分間において変化点は 18 点検出されており、一連の異常を検出できた他、トラフィック量が突然膨大になった通信を特定することができた。

図 11 は、立体の色を合計パケットサイズについてのカルマンゲインに設定した描画例である。ここで赤色として描画されている通信は閾値を超えており、トラフィックの変化点として検出されている。この通信に関し、時刻 t から時刻 $t-4$ の状態まで遡って時系列順にグラフ出力したものが図 12 である。サブネット No.1～サブネット No.10 までのサブネットから、今回変化点が検出されたサブネット No.5 への通信の各パケットサイズが棒グラフとして出力されている。ここで、グラフ上の 201708111703 は、2017 年 8 月 11 日 17 時 3 分を表している。今回は時刻 t 、つまり 2017 年 8 月 11 日 17 時 3 分の時点でサブネット No.2 からのパケットサイズが急激に増加したことが分かる (図の赤丸で囲った部分)。このように変化点を検出した場合、どの通信がどのように変化したのかを、アニメーションだけでなくグラフとして視覚的に把握することができる。

図 13 は、送信ポート番号ごとのトラフィック量について変化点を検出し、出力したグラフの例である。グラフの横軸は時刻を示し、縦軸は合計パケットサイズを示している。このグラフが出力された 2017 年 11 月 22 日 15 時 45 分において、22 番ポート (SSH) を用いたトラフィックが急増し、993 番ポート (IMAPS) を用いたトラフィックも増加していることが分かる。このように、変化点を検出したポート番号のトラフィックを重ねてグラフ出力すること

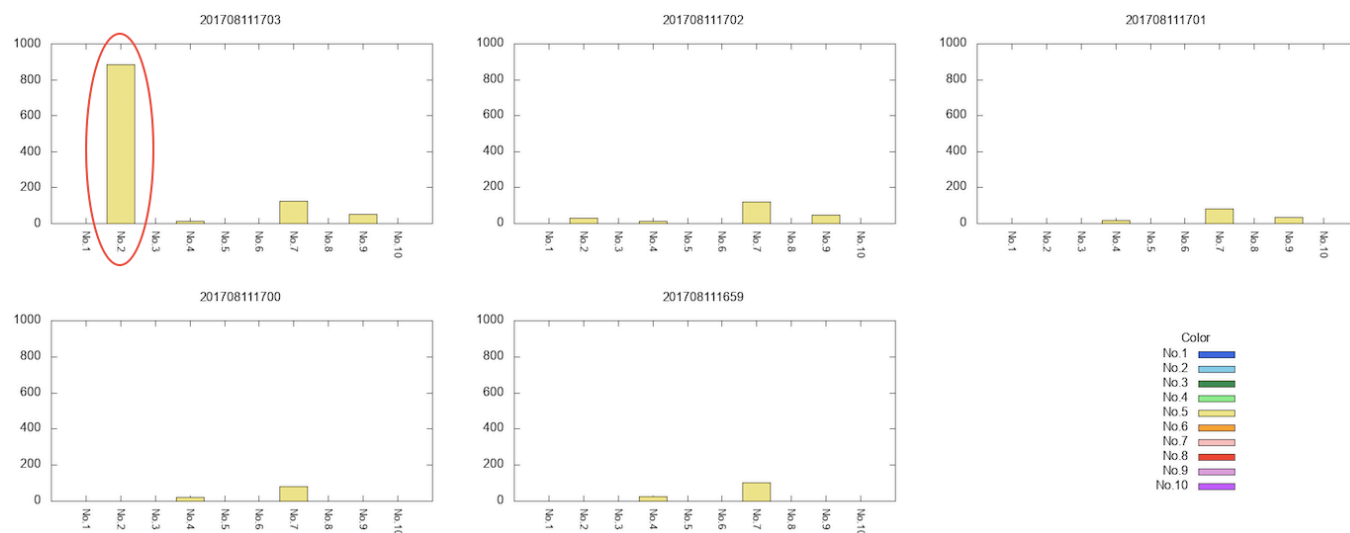


図 12 変化点検出のグラフ出力例 (サブネット)

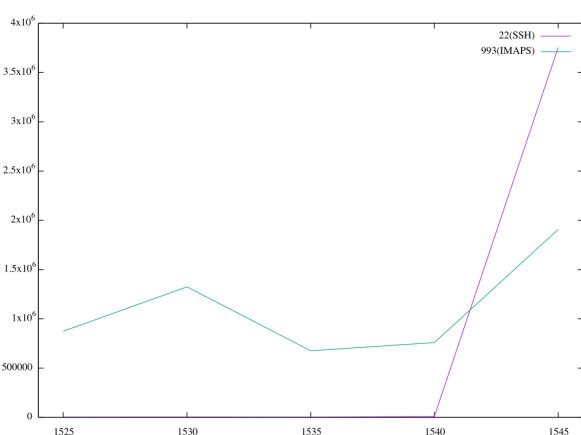


図 13 変化点検出のグラフ出力例 (ポート番号)

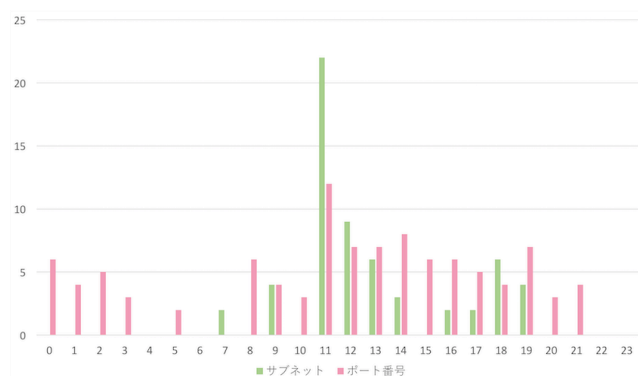


図 14 変化点の検出数

で、複数のポート番号に特徴的な挙動を示すマルウェアの感染などを特定できると考えられる。

図 14 には、前述のネットワーク障害が発生した 2017 年 8 月 17 日において検出した変化点の数を、1 時間毎に棒グラフで示している。横軸が観測時刻、縦軸は検出した変化点の数を表す。緑色で示すグラフがサブネット間のトラフィック量について検出した数、桃色で示すグラフがポ

ト番号ごとのトラフィック量について検出した数である。サブネット間のトラフィック量については、前述のネットワーク障害が発生した 11 時に、他と比べて倍以上となる 22 点の変化点が検出されている。この内 17 点がネットワーク障害の影響によるものであり、他 5 点はファイル転送の通信についての検出である。また、ポート番号ごとのトラフィック量についても 11 時に変化点の数が最多となっており、その後も 22 番ポート (SSH)、25 番ポート (SMTP)、993 番ポート (IMAPS) について一定の変化点を検出し続けた。なお、この日はサブネットについてもポート番号についても、毎秒 500Byte 以下の少ないトラフィック量の通信について変化点を検出していた。8 月 17 日は大学が夏休み期間中であり全体的にトラフィック量が少なかったため、相対的に大きな変化として検出されたものと考えられる。

このように、正常な通信であってもトラフィックの流量が大きく変化すると変化点として検出されるが、ネットワーク全体に影響を及ぼすような障害が発生した場合には変化点が数多く検出されるため、異常を判断しやすい。今後、検出した変化点が正常か異常かを自動的に判断する手法も考案していく予定である。

4.3 考察

2017 年 8 月 17 日に発生したネットワーク障害を、カルマンフィルタによって検出することができた。変化点検出後のグラフ出力によってネットワーク障害の影響を大きく受けた通信を特定できた他、アニメーションによってネットワーク障害時のトラフィック全体の様子を再現することができた。ネットワーク全体に影響を及ぼすような障害発生時に検出する変化点は、正常通信時に検出する変化点よりも数が多くなるため、異常の発生を容易に判断できる。今後は、検出した変化点を自動的に正常か異常か判断でき

表 2 Change Finder との変化点検出数の比較

Change Finder(127)		
カルマンフィルタ (162)	○	×
	○	×
	113	49
	×	14
		—

る検出手法について検討したい。

表 2 は、前述のネットワーク障害が発生した日において検出した変化点の数を、本手法と Change Finder[12] で比較した結果である。単位時間を 5 分として各サブネット間の通信とポート番号 16 種類の通信について変化点を検出した結果、本手法は合計 162 点、Change Finder は合計 127 点の変化点を検出した。うちカルマンフィルタと Change Finder が、同じ時間帯に同じ通信について検出した変化点は合計 113 点だった。また、カルマンフィルタで検出して Change Finder では検出されなかった変化点は 49 点あり、これは Change Finder が規則性のある変化を定常状態と学習しながら検出を行うため、一定の変化を継続するトラフィックに対して検出する変化点が減少したためと考えられる。一方、Change Finder で検出してカルマンフィルタでは検出されなかった変化点 14 点は、主にネットワーク障害が回復した後のトラフィックについてであった。今後、両手法で検出した変化点の差異を分析し、どのような場合に本手法で有効な変化点が検出できているのかを検証する予定である。

今回の実験では、開発したツールによってサブネット間のトラフィックを視覚的に分かりやすく把握可能であることを確認できた。また、カルマンフィルタを用いたトラフィックの変化点検出を行い、障害等が発生した場合に、アニメーションを常時監視していなくともグラフ出力によって情報を得ることができた。各通信ごとのトラフィック量とポート番号ごとのトラフィック量について並行して変化点を検出することにより、通信変化量が多いサブネットとポート番号のトラフィック推移をそれぞれ監視し、ネットワークに発生する異常をより詳細に把握できると考えられる。

5. まとめ

本研究では、組織内ネットワークにおけるトラフィックを地図上へ三次元的に描画して可視化するとともに、sFlow を用いて取得した情報を利用してカルマンフィルタによる変化点検出を行う手法を提案した。組織内のネットワークに大きな変化があった際の情報はグラフとして出力可能であり、管理者はアニメーションを常に監視していなくとも変化を知ることができる。アニメーションによる描画は過去のデータを参照して再現することもでき、障害等の発生の様子を後からでも確認可能である。本手法を用いることで、文字と数値だけでは把握が困難であった膨大なトラフィックデータを、管理者に対して視覚的かつ直感的に分

かりやすく提示することができた。なお、本ツールは GUI による直感的な操作で、管理者が得たい情報に合わせて描画の表現方法を変更したり、得たい情報のみを抽出した描画が可能である。

今後の展開として、変化点を正常か異常か判断する手法の考案、アニメーションとグラフ以外での可視化手法の検討などが挙げられる。さらに、管理者にとって使いやすいツールを目指しているため、GUI の機能性の更なる向上や便利なオプションの追加など、より管理者の目線に立った新たな機能の検討が考えられる。

参考文献

- [1] 鈴木 和也, 馬場 俊輔, 和田 英彦, 中尾 康二, 高倉 弘喜, 岡部 寿男: 迅速な障害対応を支援するトラフィック可視化システムの構築と評価, 電子情報通信学会 (B), Vol.J92-B, No.7, pp.1072-1083, 2009.
- [2] 鈴木宏栄, 衛藤将史, 井上大介: ネットワークトラフィック可視化システム NIRVANA の開発と評価, 情報通信研究機構, Vol.57, pp.63-80, 2011.
- [3] MRTG-Tobi Oetiker' s MRTG-The Multi Router Traffic Grapher: <http://oss.oetiker.ch/mrtg/>, 2016 年 2 月 29 日 13:12.
- [4] RRDtool-About RRDtool: http://www.a10networks.co.jp/lp_thunder-adc/, 2016 年 2 月 29 日 13:12.
- [5] 大野一広, 小池英樹, 小泉芳: IPMatrix:広域ネットワーク監視のための視覚化手法, 情報処理学会論文誌, vol.47, no.4, pp.1077-1086, 2006.
- [6] F. Mansman, L. Meier, and D.A. Keim: Visualization of Host Behavior for Network Security, Proc. Workshop Visualization for Computer Security (VizSEC ' 07), pp. 187-202, 2008.
- [7] R. Ball, G.A. Fink, and C. North: Home-Centric Visualization of Network Traffic for Security Administration, Proc. ACM Workshop Visualization and Data Mining for Computer Security, pp. 55-64, 2004.
- [8] J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, and M. Christensen: PortVis: A Tool for Port-Based Detection of Security Events, Proc. the ACM Workshop Visualization and Data Mining for Computer Security, pp. 73-81, 2004.
- [9] Y. Livnat, J. Agutter, S. Moon, R. Erbacher, and S. Foresti: A Visualization Paradigm for Network Intrusion Detection, Proc. Sixth Ann. IEEE SMC Information Assurance Workshop (IAW ' 05), pp. 92-99, 2005.
- [10] FireEye: <https://www.fireeye.jp/>, 2017 年 8 月 23 日 16:49.
- [11] A10 Thunder ADC: <http://www.a10networks.co.jp/products/thunderseries/thunder-adc.html>, 2017 年 8 月 23 日 16:49.
- [12] Jun-ichi Takeuchi, Kenji Yamanishi: A Unifying Framework for Detecting Outliers and Change Points from Time Series, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 18, NO. 4, APRIL 2006.
- [13] Augustin Soule, Kave Salamatian, Antonio Nucci, Nina Taft: Traffic Matrix Tracking using Kalman Filters, ACM SIGMETRICS Performance Evaluation Review - Special issue on the First ACM SIGMETRICS Workshop on Large Scale Network Inference (LSNI 2005) Homepage archive Volume 33 Issue 3, December 2005.