第Ⅴ部

ネットワーク管理とセキュリティ

第 5 部
ネットワーク管理とセキュリティ

第 1 章　はじめに

netman ワーキンググループでは、SNMP 技術を基盤とし、ネットワーク管理に関わる研究活動を行っている。

本年度は、ネットワークのパッシブモニタリング技術に関する研究を行った。第 2 章ではセキュリティ管理のための IP パケットトレース技術を、第 3 章では昨年度より継続して開発を行っているマルチプロトコル対応トラフィック情報収集モニタ (CpMonitor) における VLAN 対応技術について報告する。第 4 章では、現在配備を進めている IPv6 ネットワーク対応パッシブモニタの概要と今後の研究活動予定について報告する。

また、本報告の他に、ワーキンググループ内で、SNMPv3 技術の普及を目的とした解説記事の執筆を行い、(株)IDG ジャパンから出版されている月刊誌 “Network World” の 2003 年 1 月号に「トラブル解決のための SNMP 徹底活用術」(p.70–105) として掲載された。こちらも参照いただければ幸いである。

第 2 章　An Architecture IP Packet Tracing

2.1 概要

IP データグラムの通過経路を把握することはインターネットの管理やセキュリティの観点から興味深い技術である。しかし、IP データグラムの通過経路を決定することは非常に難しい問題である。もし、データグラムのソースアドレスが既知であるならば、そのパケットの通過経路は “追跡” できるかもしれない。しかし、インターネットでは経路情報は動的に変化するので、“追跡” 結果はあくまでも一つの可能性を示しているにすぎない。また IP データグラムのソースアドレスが “偽造” されているならば問題はさらに複雑なものとなる。そのような状況では、パケットには通過経路について何の情報を含んでいない。

IP データグラムの通過経路の追跡技術は、ネットワーク上に配置された幾つかのネットワークモニタの維持を伴う。各モニタは、“観測” したデータグラムのレコード値を記録保持する。エージェントはモニタに管理されているレコード値にアクセスすることにより、特定の IP データグラムがそのモニタが観測しているネットワークを実際に通過したかどうかを正確に把握する。十分な数のモニタの配置により、ネットワークに沿った実際のパケット通過経路の追跡は可能であるだろう。

上記に挙げた技術は、ネットワークを通過する IP データグラムの内容が変化しないことを仮定している。しかし、IP データグラムの Time-to-live 値は、ネットワークホップを通過するごとに 1 ずつ減らされ、その際にヘッダのチェックサム値も再計算されように、一般的に IP データグラムは変更されている。他にも変更されるフラグ値やオプション値も存在し、NAT ルータでは、ソースアドレスやデスティネーションアドレスも変更される。しかし、IP データグラム中のそれらの部分をマスクしても十分な大きさの不変部分が得られるので、大きな問題にならない。

ここでは、インターネット上における IP データグラムの追跡用アーキテクチャについて説明する。

2.2 アーキテクチャ

[84] では、Packet Tracing のアーキテクチャの構成要素に “monitor”、“packet record bases”、“packet record agents”、“packet trackers” を挙げている。

Packet Monitor(PM)
ネットワークのある特定箇所で IP データグラムを観測している要素。

Packet Record Base(PRB)
Packet Monitor によって観測された各 IP データグラムのレコード値を保存する要素。この際 PRB に格納されるパケットレコード値は、特定のパケッ

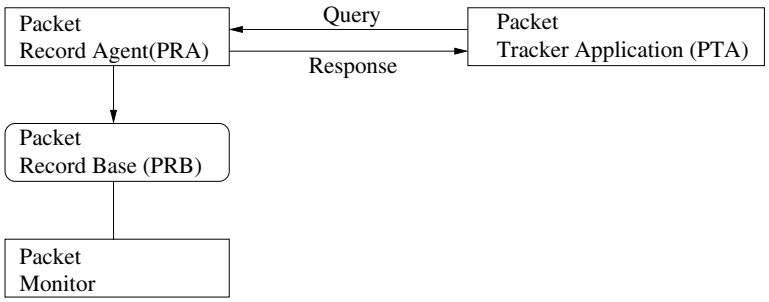

図 2.1.　Packet tracing architecture.

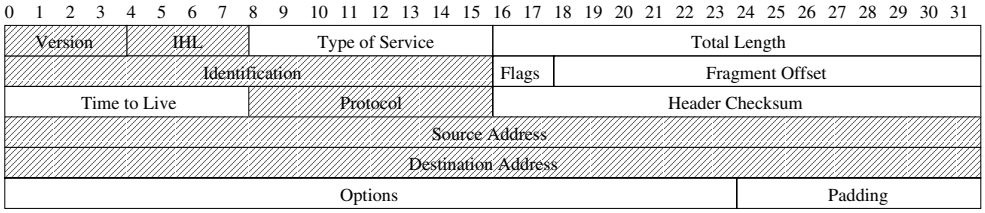

図 2.2.　IPv4 ヘッダ中のパケットレコード値算出に用いられたフィールド

トが PRB を通過したかしないかを決定づけるために十分な形式を取らなければいけない。

Packet Record Agent(PRA)

IP データグラムについての検索を受けつける要素。PRA は、PRB の中のパケットレコード値を参照して検索対象の IP データグラムが存在していたかどうかを答える。

Packet Tracker Application(PTA)

IP データグラムの経路を追跡するアプリケーション。PRA に問いあわせを行う。エージェントからの返答が、そのような IP データグラムのパケットレコードは存在しないという返答があれば、アプリケーションは追跡対象 IP データグラムが、該当 PRA の観測箇所を通過しなかったと判断する。

2.3 実装

WIDE 研究会、2002 年の秋の合宿では、このアーキテクチャに基づいた SNMP ベースによる実装の評価実験を行った。以下に実装内容と外部への貢献について説明する。

Packet Monitor(PM)

FreeBSD 4.5-RELEASE 上で libpcap を用いてトラフィックを監視した。

Packet Record Base(PRB)

libpcap で補足したパケットのハッシュ関数を用いてパケットレコード値を算出した。ハッシュ関数を適応した箇所はルータにより変更されない IP ヘッダのフィールド箇所 (図 2.2 の網かけ箇所) を抽出し、それを元にハッシュ関数 (MD5) を適応し、その結果を FreeBSD 4.5-RELEASE の db を用いて格納した。この際、PM からの指定でパケットレコード値の格納上限個数を指定できるようにした。

Packet Record Agent(PRA)

Query 用のプロトコルには SNMP を採用し、ucd-snmp 4.2.5 の snmpd に新たな MIB を実装した。

Packet Tracker Application(PTA)

追跡時刻測定用として ucd-snmp 4.2.5 の snmptrapd を用いた。またグラフィカルなアプリケーションを Java 上で実装した (図参照)。

また PTA にどのパケットを追跡すればいいかを通知するアプリケーションにオープンソフトウェアの侵入検知ソフト snort(`http://www.snort.org/`) を用いた。この機能は、現在の snort のバージョン snort 1.9.0 の SNMP アウトプットプラグインにフィールドバックされている。snort 1.9.0 の SNMP アウトプットプラグインはパケットレコード値を算出する機能が実装されており、SNMP Trap を用いてパケットレコード値を送信することが可能である。SNMP アウトプットプラグイン用 MIB として Snort-CommonMIB.txt と SnortIDAlertMIB.txt が用意されていて、この中でパケットレコード値通知用の管理オブジェクトが定義されている (図 2.3 参照)。

SnortIDAlertMIB.txt より抜粋

```
sidaSensorHashAlgorithm OBJECT-TYPE
     SYNTAX INTEGER {
            other      (1),
            md5        (2),
            sha1       (3),
            ripeMd160  (4)
     }
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
       " The hash algorithm used to compute the hash value part
         in sidaAlertPacketPrint.
       "
     ::= { sidaSensorEntry 11}

sidaAlertPacketPrint OBJECT-TYPE
     SYNTAX SnmpAdminString
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
       " The print of the invariant part of the packet that
         triggered the event. The hashing algorithm is specified
         in sidaSensorHashAlgorithm.
         The print has the following format
             <The hash value generated by sidaSensorHashAlgorithm> ':'
             <The length of the packet that was hashed>             ':'
             <The TTL of the packet>
              NULL string termination character
         The hash value is represented in hexadecimal notation.
       "
     ::= { sidaAlertEntry 31}
```

図 2.3. パケットレコード値を通知する管理オブジェクトの定義

2.4 評価

2002 年秋合宿実験ネットワーク上で本実装の評価実験を行った。実験結果に関しては第 XXI 部 大規模な仮設ネットワークテストベッドの設計・構築とその運用「SNMP による Hash-based 技術実装実験」に記載されている。

第 3 章 CpMonitor の VLAN 対応化と実用に向けた取り組み

3.1 はじめに

CpMonitor は、ネットワークトラフィックの観測を受動的に行い、計測したトラフィック情報を

```
# CpMonitor Configuration
# -----------------------
# preprocessor CGpMonitor: <statsType> <statsSpec> port <port list>
#   where, <statsType>  = vlan
#          <statsSpec> is a space separated list of specifications
#           for which the statistics must be collected.
#           e.g. when statsType = vlan
#                     statsSpec specifies the vlanIDs for which statistics
#                     will be collected
#                The format for the statsSpec is
#                [statsID@]<spec>
#                The optional statsID specifies the index that will be
#                assigned by the snmp agent.
#           <port list> is a space separated list of port numbers for
#           which you want the stats.
#           Note-1 0 is the reserved vlanID for the total.
#           Note-1 '*' is the wild card signifying all vlanIDs or,
#                  all port numbers (in the range 0-1023) depending
#                  on the preceding token.
```

図 3.1.　バックエンド設定方法

```
# VLAN 番号 100 のトラフィックを観測する場合
preprocessor CGpMonitor: vlan 100 port *
# すべての VLAN トラフィックを番号別に観測する場合
preprocessor CGpMonitor: vlan * port *
```

図 3.2.　VLAN のトラフィックを観測する設定例

SNMP(CpMonitor MIB) により管理アプリケーションに提供する、マルチプロトコル対応トラフィック情報収集エージェントである。

本年度我々は、2001 年度に開発したプロトタイプをより実用的にするための取り組みとして、広く利用されている 802.1Q VLAN を解釈する機能の追加を行った。

また 2002 年秋合宿やネットワークや東京 NOC への設置を行い、実際の利用に際しての試験を行った。

3.2 802.1Q VLAN 対応

2002 年秋合宿ネットワークでの試験結果を元に、バックエンド部の動作を規定するルールの拡張、MIB の拡張を行った。

3.2.1 バックエンド部の拡張

次のように、802.1Q VLAN に関する設定項目を追加した。

実際の設定例を図 3.2 に示す。

3.2.2 フロントエンド部の変更

今後の開発を容易にし、IPv6 トランスポートにも対応するため UCD-SNMP バージョン 4 ベースから NET-SNMP バージョン 5 ベースへの移行を行った。現在最新版の 5.0.6 をベースとして利用している。

3.2.3 拡張された CpMonitor MIB

拡張により増加した情報を扱うため、拡張された MIB を図 3.3 に示す。VLAN 番号や将来的にその他の情報で分類するためにインデックスが増加していること、デバッグ・性能評価のためのカウンタが追加されたことが大きな変化である。

本 MIB を実装した CpMonitor エージェントに対する snmpwalk の結果を図 3.4 に示す。

MO「cpmStatsID」が今回拡張されたインデックスである。「cpmStatsType」が「vlan」のものが VLAN 部分である。また、「Total」は VLAN・非 VLAN を含めた全体である。インデックスに対する VLAN 番号は「cpmStatsDescr」の値として示される。

結果として VLAN 番号毎にインデックスが振られ、それぞれの VLAN のトラフィック情報が別々に、「Total」として非 VLAN 部分も含めた全体のトラフィック情報が取得できているのがわかる。

またポート番号別アプリケーショントラフィック情報も、VLAN 番号毎に分別して情報を取得できている様子が示されている。

```
CYSOLS-PASSIVE-MONITOR-MIB DEFINITIONS ::= BEGIN
 IMPORTS
   MODULE-IDENTITY,  Counter32, Integer32,
   OBJECT-TYPE
              FROM SNMPv2-SMI
   DateAndTime
              FROM SNMPv2-TC
   MODULE-COMPLIANCE, OBJECT-GROUP
              FROM SNMPv2-CONF
   SnmpAdminString
              FROM SNMP-FRAMEWORK-MIB
   InetAddressType, InetAddress
              FROM INET-ADDRESS-MIB
   InterfaceIndex
              FROM IF-MIB
   cysolExp
              FROM CYSOL-MIB;

cpMonitorMIB MODULE-IDENTITY
    LAST-UPDATED "200203010000Z"        --   1st March 2002
    ORGANIZATION "Cysols.com"
    CONTACT-INFO
   "                      Glenn Mansfield Keeni
                 Postal: Cyber Solutions Inc.
                         6-6-3, Minami Yoshinari
                         Aoba-ku, Sendai, Japan 989-3204.
                    Tel: +81-22-303-4012
                    Fax: +81-22-303-4015
                 E-mail: glenn@cysols.com

    Support Group E-mail: mibsupport@cysols.com"

    DESCRIPTION
            " The MIB for passive monitoring."
-- Revision History
--       010302 : TimeStamps changed to DateAndTime format
--                cpmStatsStartTimeStamp
--                cpmApStatsStartTimeStamp

         ::= { cysolExp 1 }

--  cpmMonitors: The Table of Monitors. Each row represents a Monitor.
--               the monitor configurations may allow for interface
--               per monitor. It may have zero or more filters too.
--  cpmMonitorID is the key to the table.

 cpmMonitorTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF CpmMonitorEntry
     MAX-ACCESS  not-accessible
     STATUS  current
     DESCRIPTION
       " Each row of this table contains information
         about an alert indexed by cpmMonitorID."
     ::= { cpMonitorMIB 1 }

--
--  The sensor static objects
--
```

図 3.3. CpMonitor MIB (抜粋)

```
    cpmMonitorEntry OBJECT-TYPE
        SYNTAX  CpmMonitorEntry
        MAX-ACCESS  not-accessible
        STATUS  current
        DESCRIPTION
          " Entry containing information pertaining to
            a snort sensor."
        INDEX { cpmMonitorID }
        ::= { cpmMonitorTable 1 }

    CpmMonitorEntry ::= SEQUENCE {
        cpmMonitorID
                    Integer32,
        cpmMonitorDescription
                    SnmpAdminString,
        cpmMonitorVersion
                    SnmpAdminString,
        cpmMonitorLocation
                    SnmpAdminString,
        cpmMonitorAddressType
                    InetAddressType,
        cpmMonitorAddress
                    InetAddress,
        cpmMonitorInterfaceIndex
                    InterfaceIndex
    }

(中略)
    --  cpmStats: The Table of Stats. Each row represents Stats corresponding
    --  to an interface.
    --  cpmMonitorID cpmStatsID form the key to the table.

     cpmStatsTable OBJECT-TYPE
         SYNTAX  SEQUENCE OF CpmStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
         DESCRIPTION
           " Each row of this table contains information
             about an alert indexed by cpmMonitorID and cpmStatsID."
         ::= { cpMonitorMIB 2 }

     cpmStatsEntry OBJECT-TYPE
         SYNTAX  CpmStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
         DESCRIPTION
           " Entry containing information pertaining to
             an alert."
         INDEX { cpmMonitorID, cpmStatsID}
         ::= { cpmStatsTable 1 }

     CpmStatsEntry ::= SEQUENCE {
        cpmStatsStartTimeStamp
                  DateAndTime,
        cpmStatsV4IpDgrams
                  Counter32,
        cpmStatsV4IpOctets
                  Counter32,
```

図 3.3.　(Continued.)

```
        cpmStatsV4UdpDgrams
                  Counter32,
        cpmStatsV4UdpOctets
                  Counter32,
        cpmStatsV4TcpSegs
                  Counter32,
        cpmStatsV4TcpOctets
                  Counter32,
        cpmStatsV4IcmpMsgs
                  Counter32,
        cpmStatsV4IcmpOctets
                  Counter32,
        cpmStatsV6IpDgrams
                  Counter32,
        cpmStatsV6IpOctets
                  Counter32,
        cpmStatsV6UdpDgrams
                  Counter32,
        cpmStatsV6UdpOctets
                  Counter32,
        cpmStatsV6TcpSegs
                  Counter32,
        cpmStatsV6TcpOctets
                  Counter32,
        cpmStatsV6IcmpMsgs
                  Counter32,
        cpmStatsV6IcmpOctets
                  Counter32,
        cpmStatsRecvs
                  Counter32,
        cpmStatsDrops
                  Counter32,
        cpmStatsID
                  Integer32,
        cpmStatsDescr
                  SnmpAdminString,
        cpmStatsType
                  INTEGER
    }

(中略)

    --  cpmApStats: The Table of Stats. Each row represents Stats corresponding
    --  to an interface.
    --  cpmApStatsID is the key to the table.

     cpmApStatsTable OBJECT-TYPE
         SYNTAX  SEQUENCE OF CpmApStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
         DESCRIPTION
           " Each row of this table contains information
             about an alert indexed by cpmMonitorID, cpmStatsID and cpmApStatsID."
         ::= { cpMonitorMIB 3 }
     cpmApStatsEntry OBJECT-TYPE
         SYNTAX  CpmApStatsEntry
         MAX-ACCESS  not-accessible
         STATUS  current
```

図 3.3. (Continued.)

```
        DESCRIPTION
          " Entry containing information pertaining to
            an alert."
        INDEX { cpmMonitorID, cpmStatsID, cpmApStatsPortNo}
        ::= { cpmApStatsTable 1 }

    CpmApStatsEntry ::= SEQUENCE {
        cpmApStatsPortNo
                  Integer32,
        cpmApStatsStartTimeStamp
                  DateAndTime,
        cpmApStatsV4IpDgrams
                  Counter32,
        cpmApStatsV4IpOctets
                  Counter32,
        cpmApStatsV4UdpDgrams
                  Counter32,
        cpmApStatsV4UdpOctets
                  Counter32,
        cpmApStatsV4TcpSegs
                  Counter32,
        cpmApStatsV4TcpOctets
                  Counter32,
        cpmApStatsV6IpDgrams
                  Counter32,
        cpmApStatsV6IpOctets
                  Counter32,
        cpmApStatsV6UdpDgrams
                  Counter32,
        cpmApStatsV6UdpOctets
                  Counter32,
        cpmApStatsV6TcpSegs
                  Counter32,
        cpmApStatsV6TcpOctets
                  Counter32
    }
```

(後略)

図 3.3.　(Continued.)

3.3 2002 年秋合宿ネットワークでの利用

VLAN 対応 CpMonitor のプロトタイプを設置し、デバッグに必要な情報収集を行った。また、インフラチームとの協力により、合宿地の地上線側のトラフィック情報収集・提供を行った。

公開を行った情報は次の通り。

- IPv4
  - 総トラフィック量 (上下別)
  - UDP/TCP/ICMP それぞれのトラフィック量
  - FTP/SSH/HTTP/SNMP/HTTPS/CVS それぞれのアプリケーション別トラフィック量
- IPv6
  - 総トラフィック量 (上下別)
  - UDP/TCP/ICMP それぞれのトラフィック量
  - FTP/SSH/HTTP/SNMP/HTTPS/CVS それぞれのアプリケーション別トラフィック量

公開の様子を図 3.5 に示す。安定したデータ提供、アプリケーション別のトラフィック情報の提供はオペレータに好評であった。

3.4 東京 NOC への導入

2002 年 12 月、VLAN 対応の CpMonitor を、WIDE バックボーンネットワークから東京 NOC への入り口に、イーサネットタップキットを介して接続した。現在、観測情報公開に向けた準備作業中で

```
cpmMonitorID.1 = INTEGER: 1
cpmMonitorID.2 = INTEGER: 2
(中略)
cpmStatsV4IpDgrams.1.0 = Counter32: 4184894
cpmStatsV4IpDgrams.1.4 = Counter32: 203254
cpmStatsV4IpDgrams.1.18 = Counter32: 262
cpmStatsV4IpDgrams.1.21 = Counter32: 5246
cpmStatsV4IpDgrams.1.31 = Counter32: 17461
cpmStatsV4IpDgrams.1.38 = Counter32: 3956917
cpmStatsV4IpDgrams.1.41 = Counter32: 85
cpmStatsV4IpDgrams.1.49 = Counter32: 1669
(中略)
cpmStatsID.1.0 = INTEGER: 0
cpmStatsID.1.4 = INTEGER: 4
cpmStatsID.1.18 = INTEGER: 18
cpmStatsID.1.21 = INTEGER: 21
cpmStatsID.1.31 = INTEGER: 31
cpmStatsID.1.38 = INTEGER: 38
cpmStatsID.1.41 = INTEGER: 41
cpmStatsID.1.49 = INTEGER: 49
(中略)
cpmStatsDescr.1.0 = STRING: Total
cpmStatsDescr.1.4 = STRING: 4
cpmStatsDescr.1.18 = STRING: 18
cpmStatsDescr.1.21 = STRING: 21
cpmStatsDescr.1.31 = STRING: 31
cpmStatsDescr.1.38 = STRING: 38
cpmStatsDescr.1.41 = STRING: 41
cpmStatsDescr.1.49 = STRING: 49
(中略)
cpmStatsType.1.0 = INTEGER: total(1)
cpmStatsType.1.4 = INTEGER: vlan(2)
cpmStatsType.1.18 = INTEGER: vlan(2)
cpmStatsType.1.21 = INTEGER: vlan(2)
cpmStatsType.1.31 = INTEGER: vlan(2)
cpmStatsType.1.38 = INTEGER: vlan(2)
cpmStatsType.1.41 = INTEGER: vlan(2)
cpmStatsType.1.49 = INTEGER: vlan(2)
(中略)
cpmApStatsPortNo.1.38.1 = INTEGER: 1
cpmApStatsPortNo.1.38.2 = INTEGER: 2
cpmApStatsPortNo.1.38.3 = INTEGER: 3
cpmApStatsPortNo.1.38.7 = INTEGER: 7
cpmApStatsPortNo.1.38.10 = INTEGER: 10
(中略)
cpmApStatsStartTimeStamp.1.38.1 = STRING: 2003-2-3,18:55:43.0,-9:0
cpmApStatsStartTimeStamp.1.38.2 = STRING: 2003-2-3,19:4:26.0,-9:0
cpmApStatsStartTimeStamp.1.38.3 = STRING: 2003-2-3,19:4:26.0,-9:0
cpmApStatsStartTimeStamp.1.38.7 = STRING: 2003-2-3,19:19:48.0,-9:0
cpmApStatsStartTimeStamp.1.38.10 = STRING: 2003-2-3,19:42:26.0,-9:0
(中略)
cpmApStatsV4IpDgrams.1.38.1 = Counter32: 16
cpmApStatsV4IpDgrams.1.38.2 = Counter32: 30
cpmApStatsV4IpDgrams.1.38.3 = Counter32: 5
cpmApStatsV4IpDgrams.1.38.7 = Counter32: 25
cpmApStatsV4IpDgrams.1.38.10 = Counter32: 4
(後略)
```

図 3.4. CpMonitor に対する snmpwalk の結果 (抜粋)

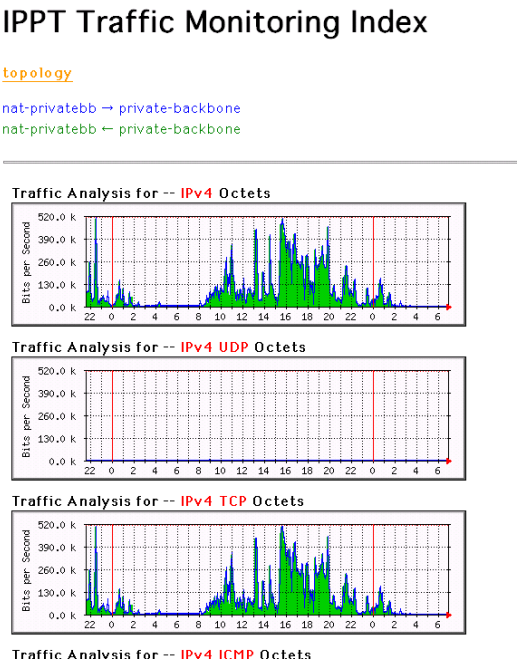

図 3.5.　合宿地のトラフィック情報公開

ある。

観測対象は広域イーサネット技術により構築されたネットワークであり、その観測のためにも 802.1Q VLAN 対応が不可欠であったといえる。

3.5 まとめ

本年度我々は、2001 年度に開発したプロトタイプをより実用的にしていくための取り組みをすすめ、広く利用されている 802.1Q VLAN を解釈する機能の追加を行った。

また 2002 年秋合宿やネットワークや東京 NOC への設置を行い、実際の利用に際しての試験を行うとともに、802.1Q VLAN 対応が必要かつ重要な拡張であったことを確認した。

第 4 章　次世代ネットワーク (JGN IPv6) の管理に関する研究

4.1 目的

IPv6 による次世代インターネットの普及を促し、その潜在能力を最大限引き出すためには、ネットワーク管理技術も次世代に対応した新しい技術が必要になる。しかし、IPv6 ネットワークにおける管理技術の整備は大幅に立ち遅れており、早急な整備が必要となっている。そこで我々は、昨年度 IPv4/IPv6 ネットワークに対応した SNMP エージェントとパッシブ型ネットワーク情報収集プローブの研究開発を行った。この研究開発により、管理プロトコルの IPv6 対応が実現し、IPv6 管理の基盤技術の確立と、これまでできていなかった IPv6 情報の汎用的な収集方式を確立し、結果として IPv6 ネットワーク上での IPv6 プロトコルのみによる管理と IPv6 トラヒック情報の詳細な収集が実現した。(図 4.1)

そこで我々は、昨年度の研究成果を基に、広域にわたる IPv6 ネットワークを安全で効率的かつ容易に管理できる管理技術の確立を目的とし、IPv6 セキュリティ管理技術の確立と監視システムのスケーラビリティの向上、IPv4 に比較して複雑性が増している IPv6 の構成管理を容易にする技術についての研究を継続して行っている。具体的には、(1) パッシブ型ネットワーク情報収集プローブのセキュリティ関連機能の拡張、および、インターネット標準プロトコルである SNMP と LDAP (ディレクトリアクセスプロトコル) を基盤とした　(2) 分散配置されたネットワーク情報収集プローブの活用技術、さらに (3)IPv6 ネットワークマップの自動生成と活用技術に関する研究開発を実施し、JGN IPv6 ネットワーク上で実運用を行い、その評価を行う計画である。

本研究における技術課題を以下に挙げる。

パッシブ型ネットワーク情報収集プローブのセキュリティ関連機能の拡張として

(a) IPv6 セキュリティ関連情報収集機能の実現

を、分散配置されたネットワーク情報収集プローブに関して次の 3 点を、

(b) 分散プローブにより収集された情報の、安全で効率的な管理技術

(c) 分散プローブの、安全で効率的な管理制御技術

(d) 分散プローブによる、セキュリティ関連情報の収集利用技術

また、ネットワークマップ技術に関して次の 2 点の研究開発を実施する。

(e) IPv6 ネットワークマップをベースとした情報提供ユーザインタフェース (UI)

(f) IPv6 ネットワークマップの自動生成技術

4.2 本研究の位置づけと方向性

本研究は、昨年度の研究成果である IPv6 パッシブ

図 4.1. パッシブ型ネットワーク情報収集プローブにより観測された IPv6 トラヒック

ネットワークモニタリング技術を拡張することにより、現在技術的に欠けている IPv6 セキュリティ管理を可能とする。さらに、IPv6 に対応し、SNMP と LDAP を基盤とした分散型ネットワーク管理情報の収集管理技術を実現することにより、世界に先駆けて広域 IPv6 ネットワークの管理システムの実現を可能とする。さらに、IPv6 ネットワークマップの自動生成技術を実現することにより、ユーザの利便性の大幅な向上や、ネットワークマップ情報をもとにしたより高度な管理アプリケーションの実現や、新しいタイプのネットワークアプリケーションの実現に新たな道を開くものである。

IPv6 の導入によりインターネットの構成要素および管理対象は飛躍的に増大する。従ってそれらを管理運用するためには分散化された情報収集管理技術と、ネットワークマップを基とした統合情報提供技術が鍵となる。本研究開発によって確立される技術は、IPv6 ネットワークの安全性を高め、また、IPv6 ネットワーク上で稼動するアプリケーションが管理情報を積極的に利用するための基盤技術となる。その結果、IPv6 ネットワークの普及をいっそう促し、また、より積極的に管理情報を活用する、ネットワークへの親和性を高めた高度なアプリケーションを実現できるようになる。

研究開発された標準技術は早い段階でインターネットの開発者グループに対して公開することを予定しており、また、それらを用いた応用技術は、本研究開発で利用する JGN IPv6 ネットワークを通じて運用，評価、開発し、大規模な検証を行う。

4.3 研究の計画・方法

本研究では、既存のインターネット標準技術、および我々が既に積極的に貢献を行っている snort や net-snmp 等のオープンソース技術を活用するとともに、研究成果を積極的に公開し、必要に応じ IETF において積極的に提案活動を行っていく。snort は、製品化されている侵入検知システムと比較しても、その性能と機能面で優位な点が多く、オープンソースの物としてはインターネット上でもっとも多く利用されている IDS パッケージである。また、net-snmp も、インターネット上でもっとも広く用いられている SNMP エージェントパッケージで、UNIX 系をはじめ Windows や MacOS-X 等、様々な OS 上で動作する。

本研究開発では、以下の 6 つの技術課題について研究開発を行う。

(1) IPv6 セキュリティ関連情報収集機能の実現

昨年度研究開発を行ったパッシブ型ネットワーク情報プローブに対して、セキュリティ関連情報の収集機能を付加する。具体的には、IPv6 ア

ドレスに対応する端末の物理アドレス (MAC アドレス等) の収集機能や、シグネチャベースによる侵入検知情報収集機能の開発を行う。

(2) 分散プローブにより収集された情報の、安全で効率的な管理技術

飛躍的に増大する管理対象と管理項目を効率的に管理するための収集情報管理技術を研究開発する。IPv6 に対応した SNMP プロトコルと LDAP プロトコルを基にし、自律的にネットワーク管理情報を収集し蓄積するエージェントを配置する。そしてそれらエージェントが収集管理するネットワーク管理情報をディレクトリ技術を用いて検索可能とすることにより、情報収集の負荷を分散し、ネットワーク管理システムの負荷の軽減や、ユーザアプリケーションからの積極的な管理情報の利用を可能とする。

(3) 分散プローブの、安全で効率的な管理制御技術

ネットワーク情報収集プローブは、それ自身の機能や構成を柔軟に変更することが可能となっている。この機能をより強力に活用するために、分散配置された多数のプローブ管理技術や、安全な制御技術の研究開発を行う。具体的には、遠隔から、もしくは自律的に、複数プローブの一括設定変更や起動，停止、プローブの機能試験等を行うための技術の研究開発を行う。

(4) 分散プローブによる、セキュリティ関連情報の収集利用技術

セキュリティ関連情報に適した情報収集技術の研究開発を行う。セキュリティ関連情報は、一般的なネットワーク管理情報と比較し、より即時性が求められるため、即時性をみたしかつ効率的な情報提供を行える通知技術と収集利用技術が必要となる。具体的には、IPv6 に対応した SNMP Inform 技術の活用と、即時性をもとめるアプリケーションとの API の研究開発を実施する。

(5) IPv6 ネットワークマップをベースとした情報提供ユーザインタフェース (UI)

IPv6 ネットワークマップをもとにした、一般ユーザにも利用のしやすいネットワーク情報提供ユーザインタフェースの開発を行う。具体的には、MRTG にあるネットワーク情報の閲覧性や時系列的ネットワーク情報の不可逆性等の問題を解決したネットワーク情報可視化ツールの研究開発と、一般ユーザが利用しやすい、ネットワークマップを基にしたネットワーク情報閲覧 GUI の研究開発を行う。

(6) IPv6 ネットワークマップの自動生成技術

IPv6 ネットワーク上からオンラインで収集可能なネットワーク情報を基に、IPv6 ネットワークマップを自動生成する技術の研究開発を行う。具体的には、IPv6 ルーティングに関する SNMP MIB 情報や、プローブにより収集されるリンクレイヤ情報、DNS やルーティングレジストリ DB 等から情報を収集分析し、ネットワークマップの基本情報である IPv6 サブネットの情報やサブネット間の接続情報、AS レベルでの IPv6 ネットワーク接続情報や AS パス情報を生成する技術の研究開発を行う。

本研究開発は、JGN IPv6 ネットワーク上で、それぞれの技術課題の成果を実際に運用評価し、その結果を研究成果にフィードバックしながらすすめる。この運用実験のために、平成 14 年度末より、以下の研究設備の導入を計画している。

(a) ネットワークモニタ装置

パッシブ型ネットワーク情報収集プローブ。本研究開発では、IPv6 セキュリティ関連情報収集機能の拡張を行う。JGN IPv6 ネットワーク内の NOC に設置し、計 30 のサブネットの情報収集を行う予定である。

(b) ネットワーク情報収集装置

ネットワーク管理情報の自立的収集機能をもち、本研究開発における分散配置されたネットワーク情報収集プローブの活用技術の運用実験に用いる。JGN IPv6 ネットワーク内の 4 つのサイト (大手町 IPv6 システム運用技術開発センター，岡山 IPv6 システム検証評価センター，京都大学，東北大学) に設置する。

これらの設備の設置例として、図 4.2 に、TAO 東北大学分室内の JGN IPv6 ネットワーク東北北海道 NOC における収集装置群の写真を示す。

表 4.1. IPv6 プローブ新設予定サイト

	サイト名	100baseTX	1000base-SX
1	東北大学シナジーセンター	1	
2	東北大学電気通信研究所	2	
3	名古屋大学	2	1
4	ソフトピアジャパン	3	
5	京都大学		1
6	広島大学	2	1
7	広島市立大学	1	
8	九州大学	8	1
9	佐賀大学	3	
10	TAO 北九州リサーチセンター	3	
11	TAO 大手町 IPv6 システム運用技術開発センター	2	1
12	TAO 岡山 IPv6 システム検証評価センター		1
13	堂島	2	1
14	TAO 高知通信トラヒックリサーチセンター	3	
15	大阪大学	2	1
16	東京大学	-	-

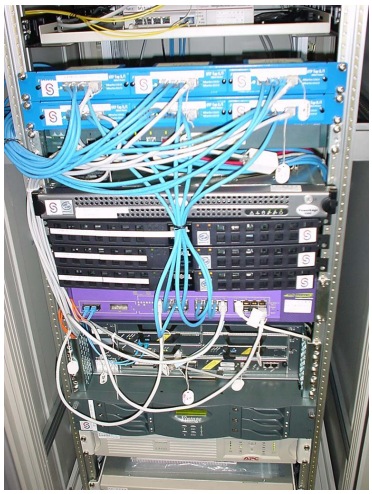

図 4.2. ネットワークモニタ装置と情報収集装置の様子 (TAO 東北大学分室)

4.4 現状と今後の予定

2002 年 12 月中旬よりプローブ装置の設置ポイントの選定を行い、現在、表 4.1 に示すサイトに装置の設置を進めている。各サイトにおけるモニタ対象リンクは 100baseTX および 1000base-SX イーサネットで、現時点で確定しているリンク数は合計 42 リンクとなっている。

図 4.3、4.4、4.5 に、実際に設置されるプローブ装置の例を示す。

図 4.3. 1 BOX タイプのプローブ装置

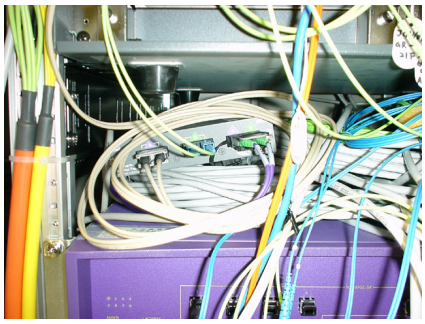

図 4.4. 1000base-SX TAP モジュール

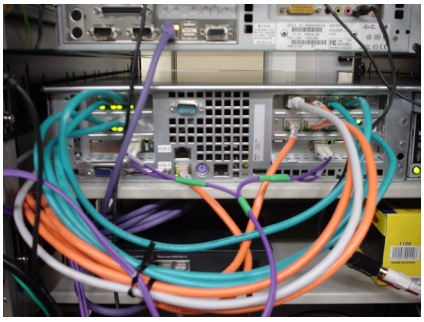

図 4.5.　2U タイプのプローブ装置例
(100baseTXx2, 1000base-SXx1)

プローブ装置の各サイトへの設置は 2003 年 3 月中には完了の予定である。今後、これらの分散プローブを用い、IPv6 トラヒック情報の JGN IPv6 ネットワーク利用者への公開なども含め、計画している研究をさらに進めていく予定である。