第 10 部

ネットワークトラフィック統計情報の収集と解析

第 1 章

はじめに

WIDE NetStat WG は、広域分散環境におけるトラフィックデータの「収集」、「解析」、「保存」、「利用」等のために必要とされる技術に関する研究をおこなうことを目的に活動を行っている。また、トラフィックデータの収集・解析によって得られた情報をネットワーク管理において利用する技術に関する研究も合わせて行っている。

これまで当 WG では、WIDE Internet 上のトラフィックの統計情報の収集・解析を通して、WIDE バックボーンの利用状況を、その上を流れるトラフィックの観点から明らかにし、また、WIDE の国際回線を通した国際的な通信の広がりについての考察を行ってきた。

本報告では、92 年度の活動をもとに、

- トラフィックデータの収集手法の解説
- WIDE バックボーン上のトラフィック解析
- TIX 等における、他のネットワークプロジェクトとの間のトラフィック解析
- 国外のネットワークとのトラフィック解析
- IETF で現在提案されている、トラフィックデータの共用を行うための、共通データフォーマットについての解説

を行う。

第 2 章

NNStat による統計情報の収集について

2.1 概要

NNStat[93] は、南カリフォルニア大学の Robert Braden らにより開発された、インターネット上のトラフィックの統計情報を収拾するためのツールである。NNStat を用いる事によって、インターネット上に分散した幾つかのイーサネットセグメントやポイント・トゥー・ポイントリンクを通過する IP トラフィックのさまざまな統計情報を同時に収集する事が可能となる。また、どのような統計情報を収集するかは、インタラクティブに、もしくはコンフィギュレーションファイルにより柔軟に指定する事が可能である。

NNStat は概念的には、SAA (Statistics Aquisition Agent) と SCH (Statistics Collection Host) の 2 つの部分から構成される。

SAA(Statistics Aquisition Agent) ネットワーク上に複数個分散配置され、各々配置された場所における IP トラフィックを監視し、コンフィギュレーションファイルによる指定にしたがって、統計情報の収集を行う。statspy というプログラムがこの役割を果たす。

SCH(Statistics Collection Host) 分散配置された複数の SAA を定期的にポーリングし、各 SAA において収集された統計情報を一箇所に集め、ファイルに格納する。collect というプログラムがこの役割を果たす。

図 2.1に、SAA および SCH を用いた NNStat による統計情報の収集の基本概念を示す。

上記 SAA と SCH の関係は SNMP における management agent と management station の関係と概念的には同じものである。

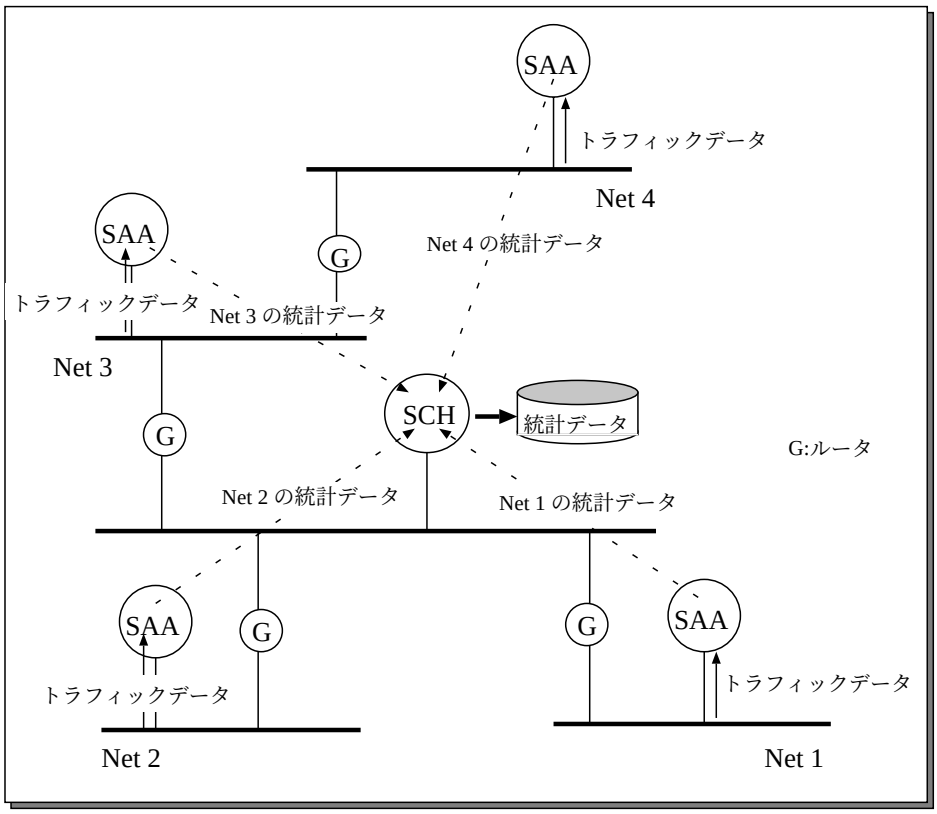

図 2.1: NNStat による統計情報収集の概念図

2.2　statspy が収集可能な統計情報

各 SAA は、イーサネットセグメントもしくはポイント・トゥー・ポイントのリンクを通過する全てのパケットを検査し、あらかじめ指定された各種条件に従って、トラフィックを分類し、統計値を計算する。

2.2.1　論理的フィールド

ここでの条件の指定では、イーサネットフレームヘッダ、IP ヘッダ、TCP および UDP ヘッダの以下のフィールドに対して、条件式を指定することができる。

フィールド名	フィールド長 (バイト数)	意味
Ether.src	6	イーサネットアドレス
Ether.dst	6	イーサネットアドレス
Ether.type	2	イーサネットタイプ
IP.version	1	IP バージョン
IP.length	2	IP データグラムのサイズ
IP.option	1	IP オプション
IP.TOS	1	TOS を表すビット列
IP.offset	2	フラグメントされている場合のオフセット値
IP.protocol	1	IP のプロトコル
IP.srchost	4	送り手の IP アドレス
IP.dsthost	4	送り先の IP アドレス
IP.srcnet	4	送り手のネットワークアドレス
IP.dstnet	4	送り先のネットワークアドレス
IP.srcsubn	4	送り手のサブネットワークアドレス
IP.dstsubn	4	送り先のサブネットワークアドレス
TCP.srcport	4	送り手の TCP ポート番号
TCP.dstport	4	送り先の TCP ポート番号
UDP.srcport	4	送り手の UDP ポート番号
UDP.dstport	4	送り先の UDP ポート番号
ICMP.type	1	ICMP の type フィールド
ICMP.code	1	ICMP の code フィールド
packet	可変	パケット全体のビット列
length	4	パケットの全長

ただし、上記の表の各フィールドは、必ずしも実際の IP データグラムのヘッダ中に存在するものばかりではなく、例えば IP.srcnet のように、実際のヘッダ中のフィールドの値を用いて計算可能な物もふくむ、論理的なものと考える。

2.2.2 オブジェクト

NNStat では、データ構造としていくつかのオブジェクトを用意しており、統計情報の収集は上記の表に示した、各 IP データグラムから得られる論理的なフィールド値を、それらのオブジェクトに代入する事により行う。

- Recorders···統計値を格納するためのオブジェクト。
  - freq-all

    与えられた 1 つのフィールド値に対する頻度情報を格納する。例えば、IP.srchost をこのオブジェクトに与えることにより、各送り手の IP アドレス毎のパケット数を数えることができる。

- freq-all-bytes

  freq-all と類似しているが、パケット数を数えるだけではなく、パケット長の累積値も同時に格納することができる。

- matrix-all

  与えられた 2 つのフィールド値に対する頻度情報を格納する。例えば、IP.srchost と IP.dsthost をこのオブジェクトに与える事により、送り手と送り先の IP アドレスの組毎のパケット数を数える事ができる。

- matrix-all-bytes

  matrix-all と類似しているが、パケット数を数えるだけではなく、パケット長の累積値も同時に格納する事ができる。

- その他 ([94] を参照)

• Filters···条件を表すオブジェクトで、条件式の一部として記述する事ができる

- eqf

  与えられたフィールド値を指定された値と比較する。

- setf

  与えられたフィールド値を指定された一連の値と比較する。eqf は setf の特別な場合と考える事ができる。

- その他 ([94] を参照)

2.2.3　コンフィギュレーション言語

上記の論理フィールド、オブジェクトと、以下に示す構文を加えたコンフィギュレーション言語を用いて、statspy に対して収集する統計情報を指定する。

コンフィギュレーション言語で使用するおもな構文には以下のものがある。

• attach 構文···オブジェクトの定義文全体

• enum 構文···フィールド値に特定の文字列を対応づける

例

```
enum {
        *port* (20 "FTP data", 21 FTP, 23 Telnet, 25 SMTP,
           37 Time, 42 Name, 43 Whois, 53 Domains,
           69 TFTP, 79 Finger, 103 X.400, 104 "X.400-SND",
           109 POP2, 111 sunrpc, 115 SFTP, 119 NetNews,
           153 SGMP, 512 exec, 513 "rwho|rlogin", 514 shell,
           515 printer, 520 RIP)
      }
```

この例では、*port*にマッチする名前をもつフィールドの値に対して、上記のような値と文字列との対応を定義する。例えば、TCP.dstport の値 23 を、Telnet という文字列によって指定可能にする。これは以下に述べる if 構文や select 構文でフィールド値に対する条件を指定する時に便利である。

- restrict 構文···簡単なアクセスコントロールを行う。

  例 restrict readwrite 192.41.197.0 0xffffff00

  この例では、この statspy プロセスに対して、192.41.197.0 のネットワーク上のホストからの読み書きのアクセスを許すことを指定する。

- subnet 構文···IP.srcsubn や IP.dstsubn フィールドによって識別可能なサブネットワークを指定する。

  例 subnet 133.160.0.0 0xfffffe0

  この例では、クラス B ネットワークである 133.160.0.0 はサブネットマスク 0xfffffe0 によってサブネットされている事を指定する。subnet 構文で指定しないかぎり IP.srcsubn や IP.dstsubn によって参照されるネットワークアドレスは、IP.srcnet や IP.dstnet によって参照されるネットワークアドレスと同一である。

- record 構文···recorders オブジェクトに対する値の代入を行う

  例 1 record IP.srchost in SOURCE-IP freq-all ;

  は、SOURCE-IP という名前の freq-all オブジェクトを定義し、各 IP.srchost 毎のパケット数を記録する。

  例 2 record IP.srcnet,IP.dstnet in MATRIX-IP matrix-all-bytes ;

  は、オブジェクトタイプ matrix-all-bytes のオブジェクト MATRIX-IP を定義し、各 IP.srcnet と IP.dstnet の組毎のパケット数を記録する。

- if 構文···C 言語等と同様の通常の条件分岐構文

- select 構文···C 言語等の switch 構文と類似した制御構造を与える構文

  例

```
select TCP.dstport selectSport {
    case ("Telnet", "Whois", "Finger", "rwho|rlogin"):
        record IP.srchost, IP.dsthost in Telnet.hosts matrix-all;

    case ("FTP data", "FTP", "TFTP"):
        record IP.srchost IP.dsthost in ftp.hosts matrix-all;

    case ("SMTP", "X.400", "X.400-SND", "NetNews"):
```

```
                record IP.srchost IP.dsthost in mail.hosts matrix-all;

            default:
                select TCP.srcport  selectDport {
                    case ("Telnet", "Whois", "Finger", "rwho|rlogin"):
                       record IP.srchost, IP.dsthost in Telnet.hosts
                                                    matrix-all;

                   case ("FTP data", "FTP", "TFTP"):
                       record IP.srchost IP.dsthost in ftp.hosts matrix-all;

                   case ("SMTP", "X.400", "X.400-SND", "NetNews"):
                       record IP.srchost IP.dsthost in mail.hosts matrix-all;
                }  #end of select TCP.srcport

        }  # end of select TCP.dstport
```

上記の例では、TCP.dstport および TCP.srcport によって、各パケットがどの TCP アプリケーションに属するかを決め、それに応じて対応する名前をもつ matrix-all タイプのオブジェクトである Telnet.hosts、ftp.hosts、mail.hosts のそれぞれに、パケットの個数を記録する。

以下に、上記の構文を用いたコンフィギュレーションファイルの例を示す。

```
#
# Enum of TCP/UDP ports
#
enum {
 *port* (20 "FTP data", 21 FTP, 23 Telnet, 25 SMTP,
     37 Time, 42 Name, 43 Whois, 53 Domains,
     69 TFTP, 79 Finger, 103 X.400, 104 "X.400-SND",
     111 sunrpc, 115 SFTP, 119 NNTP, 161 SNMP,
     162 "SNMP trap", 512 exec, 513 "rwho|rlogin",
     514 shell, 515 printer, 520 RIP)

 *IP.proto* (1 ICMP, 3 GGP, 6 TCP, 8 EGP, 12 PUP, 17 UDP,
             20 HMP, 21 "XNS-IDP", 27 RDP, 63 HELLO, 77 ND),

 *ICMP*  (0 "Echo Rep", 3 "Dest Unreach", 4 "Source Quench",
          5 "Redirect", 8 "Echo Req", 11 "Time Exc",
          12 "Param Prob", 13 "Time Stamp",
          14 "Time Stamp Rep", 15 "Info Req", 16 "Info Rep",
          17 "AddrMask Req", 18 "AddrMask Rep")
```

```
}
#
# restrict remote access to statspy
#
restrict readwrite 192.41.197.5  0xffffffff #clapton
restrict readwrite 130.69.251.15  0xffffffff #clapton
restrict readwrite 127.0.0.1 0xffffffff #localhost

subnet 133.4.0.0 0xfffffe0                 #wide-bb

attach {
 #
 #  Record all IP packets.
 #
 record IP.srcsubn,IP.dstsubn in IP matrix-all-bytes ;
 #
 # Record ICMP packets.
 #
 if IP.protocol is IP.proto eqf("ICMP") {
  record ICMP.type,ICMP.code in ICMP.types matrix-all-bytes ;
record IP.srcsubn,IP.dstsubn in ICMP matrix-all-bytes ;
        record packet in ICMP.dump bin-pkt(50) ;
 }
 #
 # Record TCP packets.
 #
 select TCP.dstport selDport {
   case ("Telnet", "rwho|rlogin"):
    {
record IP.srcsubn,IP.dstsubn in Telnet matrix-all-bytes;
    }
   case ("FTP", "FTP data"):
    {
record IP.srcsubn,IP.dstsubn in FTP matrix-all-bytes;
    }
   case ("SMTP"):
    {
record IP.srcsubn,IP.dstsubn in SMTP matrix-all-bytes;
    }
```

```
   case ("NNTP"):
    {
record IP.srchost,IP.dsthost in NNTP matrix-all-bytes;
    }
   default:
      select TCP.srcport selSport {
  case ("Telnet", "rwho|rlogin"):
    {
     record IP.srcsubn,IP.dstsubn in Telnet ;
  }
case ("FTP", "FTP data"):
  {
   record IP.srcsubn,IP.dstsubn in FTP ;
  }
case ("SMTP"):
  {
   record IP.srcsubn,IP.dstsubn in SMTP ;
  }
  case ("NNTP"):
        {
   record IP.srchost,IP.dsthost in NNTP ;
          }
    }
  }
}
```

この例では、以下のような統計情報が収集される。

オブジェクト名	統計情報
IP	IP パケットの送り手ネットワークと送り先ネットワークの組毎のパケット数とバイト数
ICMP.types	ICMP メッセージの types と code の組毎のパケット数とバイト数
ICMP	ICMP メッセージの送り手と送り先ホストの組毎のパケット数とバイト数
ICMP.dump	全ての ICMP メッセージのダンプ
Telnet	Telnet、rwho、rlogin の送り手ネットワークと送り先ネットワークの組毎のパケット数とバイト数
FTP	FTP、FTP data の送り手ネットワークと送り先ネットワークの組毎のパケット数とバイト数
SMTP	SMTP に属するパケットの送り手ネットワークと送り先ネットワークの組毎のパケット数とバイト数
NNTP	NNTP に属するパケットの送り手ネットワークと送り先ネットワークの組毎のパケット数とバイト数

2.2.4 collect による statspy へのアクセス

前節で解説したように、統計をとりたい各ネットワーク上で statspy プログラムを動かし、collect プログラムにより遠隔ホスト上で動作する statspy プログラムからおのおので収集されている統計情報を 1 箇所に集めることができる。

collect によるアクセスの種類には、基本的に次の 2 種類がある。

read 指定したオブジェクトの現在の値を読み出す。

readclear 指定したオブジェクトの現在の値を読みだし、オブジェクトの保持するデータをクリアする。

従って、collect による statspy へのアクセスの戦略としては、

- 比較的短い周期で各 statspy に対して必要なオブジェクトに対する read アクセスを行い、得られたデータをファイルに書き出す。
- 上記よりも長い周期で各 statspy に対して必要なオブジェクトを指定した readclear アクセスを行い、得られたデータをファイルに書き出す。

がとられている。

collect によって作られる統計情報のファイルは、statspy が動作しているホスト毎に、さらに、オブジェクト毎に分割されて作られる。各々のファイルに書き出される内容は、そのオブジェクトの名前、オブジェクトが定義された時刻 (統計をとり始めた時刻)、以前に readclear された時の時刻、現在の時刻、オブジェクトの内容等である。

以下には、collect により作成されるファイルの例を示す。

```
Log created on Sun Oct 11 00:05:49 1992, for host jp-tap.wide.ad.jp.
    Sample interval = 20 minutes; checkpoint interval = 60 minutes.
    Clear interval = 60 minutes.
    Object name = 'IP'.

OBJECT: IP  Class= matrix-all-bytes  [Created: 11:30:18 10-09-92]
  ReadTime: 01:00:00 10-11-92, ClearTime: 00:00:00 10-11-92 (@-3600sec)
  Total Count= 131762 (+0 orphans)
  Total Bytes= 24589322B    #bins = 812
[133.164.0.0 : 133.160.0.0]= 67616 &9263392B   (51.3%) @-0sec
[130.209.0.0 : 133.50.0.0]= 8321 &2942916B     ( 6.3%) @-257sec
[130.215.0.0 : 133.26.0.0]= 6009 &3286226B     ( 4.6%) @-596sec
[138.95.0.0 : 130.34.0.0]= 5007 &201875B       ( 3.8%) @-3sec
[134.95.0.0 : 133.9.0.0]= 4668 &256740B    ( 3.5%) @-4sec
[128.252.0.0 : 130.34.0.0]= 3952 &2040828B ( 3.0%) @-576sec
[192.42.239.0 : 130.34.0.0]= 3617 &146353B ( 2.7%) @-2229sec
                      .
                      .
                 [途中省略]
                      .
                      .

[137.39.0.0 : 130.34.0.0]= 1 &67B  ( <.1%) @-3212sec
[138.253.0.0 : 133.4.1.0]= 1 &67B  ( <.1%) @-3251sec


OBJECT: IP Class= matrix-all-bytes  [Created: 11:30:18 10-09-92]
  ReadTime: 02:00:00 10-11-92, ClearTime: 01:00:00 10-11-92 (@-3600sec)
  Total Count= 193532 (+0 orphans)
  Total Bytes= 24243313B    #bins = 825
[134.95.0.0 : 133.9.0.0]= 82161 &4518855B  (42.5%) @-19sec
[133.164.0.0 : 133.160.0.0]= 68548 &9391076B   (35.4%) @-0sec
[128.59.0.0 : 133.1.0.0]= 6649 &3435289B   ( 3.4%) @-39sec
[16.0.0.0 : 133.4.1.0]= 4438 &2252155B     ( 2.3%) @-2062sec
                      .
                      .
                 [途中省略]
                      .
```

```
                    .

[130.127.0.0 : 133.4.4.0]= 1 &69B  ( <.1%) @-3566sec
[129.24.0.0 : 133.4.2.0]= 1 &61B   ( <.1%) @-3576sec


OBJECT: IP  Class= matrix-all-bytes  [Created: 11:30:18 10-09-92]
  ReadTime: 03:00:00 10-11-92, ClearTime: 02:00:00 10-11-92 (@-3600sec)
  Total Count= 121194 (+0 orphans)
  Total Bytes= 24399400B    #bins = 823
[133.164.0.0 : 133.160.0.0]= 74265 &10174305B  (61.3%) @-0sec
[18.0.0.0 : 131.113.0.0]= 5397 &2866870B       ( 4.5%) @-438sec
[128.59.0.0 : 130.69.0.0]= 4686 &2554929B      ( 3.9%) @-62sec
[141.210.0.0 : 130.158.0.0]= 4432 &2225222B    ( 3.7%) @-597sec
                    .
                    .
                [以下省略]
                    .
                    .
```

この例では 1 時間毎にオブジェクト IP を read clear しており、例えば時刻 1:00 から 2:00 の間に 133.95.0.0 から 133.9.0.0 への IP パケットが 82161 個あり、それらのパケットによって運ばれたデータ量が 4518855 バイトあったことが分かる。

これ以外にも同様に、オブジェクト Telnet や SMTP などの統計情報のファイルも同時に作られるので、各々のファイルの内容を総合すれば、各ネットワーク間で各アプリケーションによるトラフィックが 1 時間毎にどのくらいあったかを調べることができる。

これらのファイルに書き込まれた統計情報を、perl[95] 等のスクリプトを用いる事により集計し、アプリケーション別のトラフィックの表を作ったり、グラフを描いたりすることができ、それにより、データを収集したネットワークにおけるトラフィックの傾向を把握する事ができ、構成管理やアカウント管理、場合によっては障害管理にも役立てる事ができる。

2.3 利点と欠点

以上簡単ではあるが、NNStat を用いた IP トラフィックの統計情報収集の手法に付いて概観した。以下に NNStat を用いた手法における利点と欠点をまとめる。

利点

- 統計情報収集の作業を SAA と SCH とに分散する事により、遠隔地にある複数のホストで、SAA プログラムを動作させ、1 箇所においた SCH プログラムか

ら全ての SAA において収集されている統計情報を集める事ができるので、統計データの管理が容易に行える。

- イーサネット上の全てのトラフィックを観測できるので、SAA をイーサネットセグメントに１つおいておけば、そのセグメント上の全てのホストに関わるトラフィック、および、そのセグメントを通過して行くトラフィックの全てを含む統計情報を得られる。
- コンフィギュレーション言語により、収集したい統計情報を定義する事ができ、SAA によって収集する統計情報を変えたり、あとから新たに収集する統計情報を付け加えたりする事が柔軟に行える。

欠点

- イーサネット上に現れないトラフィックを観測する事ができない。例えば、図 2.2のような構成のネットワークの場合、ネットワーク１からネットワーク４に向かうトラフィックは、ネットワーク３を通過するので、図の SAA で観測可能であるが、ネットワーク１からネットワーク２に向かうトラフィックは、ルーター１上で折り返してしまい、ネットワーク２には現れないので、図の SAA では観測不可能になる。このような場合、各ルーター上でも同様に SAA プログラムを動作させる必要があるが、ルーターが専用ルーターである場合には、それができない。特に、WIDE のようなバックボーンネットワークの場合、各 NOC におかれたルーターに複数の接続組織へのリンクが収容されている場合が多く、その様な場合には各組織毎の正確な統計情報が得られなくなる。
- SNMP を併用しているような場合、NNStat とは別に運用する必要があり、得られたデータの集計や管理に手間がかかる。

2.4　まとめ

本章では、NNStat を用いたトラフィックの統計収集手法に付いて簡単にまとめ、その利点および欠点に付いて議論した。

前節で述べたような利点を活かしたまま欠点を補って行くためには、専用ルーターにおいても、ここで概観した NNStat のような統計収集手法が実現されることが望まれる。その際、SNMP との融合に関しても考慮されるべきであり、enterprise MIB のような枠組の中で、統計情報を柔軟に定義するための手法が用意され、標準 MIB と同様に Management Station からアクセスできるようになれば理想的である。

その際に検討しなければならない課題としては以下のものがあげられる。

- 全てのパケットのトランスポートヘッダまで調べる必要があるため、統計収集を行うことが、ルーターの通常の処理に対してどの程度のオーバーヘッドを与えるか。
- 収集する統計情報を柔軟に定義するためには、どのように MIB を構成する必要があるか。

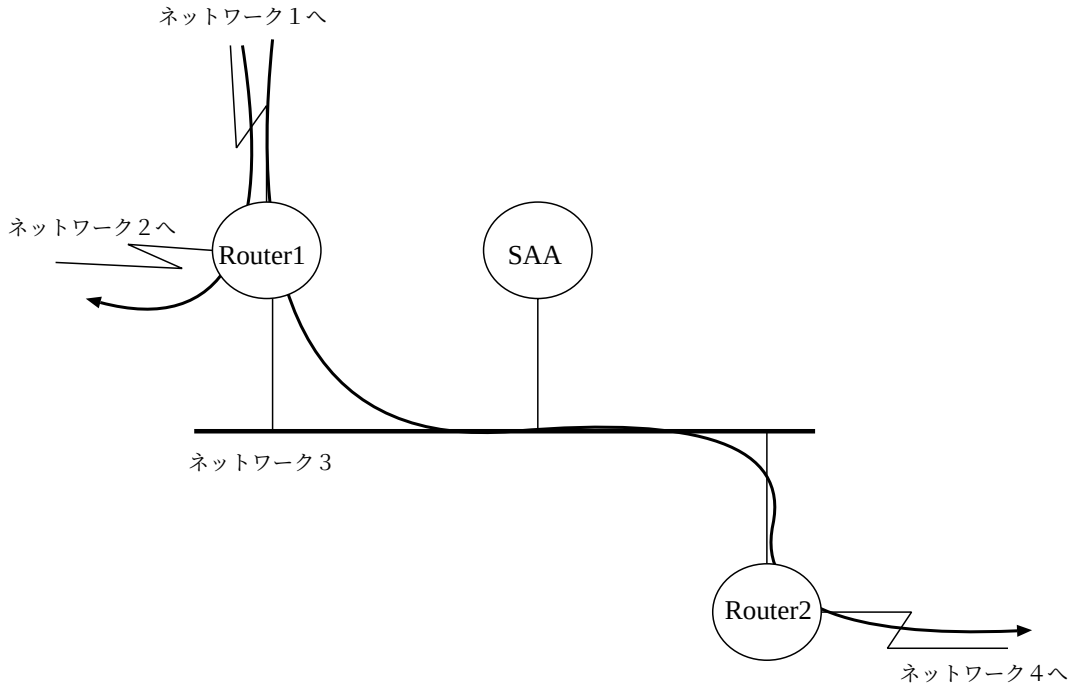

図 2.2: NNStat により観測できないトラフィック

- ある程度の量の統計データを保持するために、ルーターの通常の処理に必要なメモリ容量に較べて、どの程度余分なメモリ容量が必要となるか。

なお、今回は述べなかったが、さらに tcpdump[96] のような、各パケットの内容を解析するためのツールも専用ルーター上に実現することが今後必要になることも考えられる。

第 3 章

WIDE インターネット上のトラフィック解析

WIDE NetStat WG では、WIDE インターネットの利用状況を把握し、また将来の回線計画のための裏付けとなる情報を得るために、2 年前から 2で解説した nnstat を用いて WNOC-TYO と WNOC-SFC 間のトラフィックおよび国際回線のトラフィックデータの収集・解析をおこなっている。

また、WIDE プロジェクトでは、国内の他のネットワークとの相互接続を行なっている。現在、主な相互接続地点としては、TIX(Tokyo Internet eXchange)、WNOC-TYO、WNOC-KYO の 3 箇所である。WNOC-KYO では、TISN(Genome) との相互接続を行ない、WNOC-TYO では、SINET と JOIN との相互接続を行なっている。また、TIX では、TRAIN, TISN, JAIN との相互接続および UTnet との接続を行なっている。WIDE NetStat WG では WIDE のバックボーントラフィックと同様に、TIX や WNOC-TYO におけるネットワーク相互間のトラフィックの収集・解析も行っている。

本章では、WIDE インターネット上のトラフィックおよび各ネットワーク間のトラフィックに関する統計情報を示す。

3.1 トラフィックデータ収集環境

図 3.1に、現在の東京地域におけるネットワーク相互接続の状況およびトラフィックデータ収集環境を示す。

図中の jp-tap、wnoc-tyo、stat というホスト上で、statspy を動作させ、それぞれ国際線、WNOC-TYO と WNOC-SFC 間、TIX のトラフィックデータを収集し、stat というホストで collect を動作させ、各ホストで収集されたトラフィックデータを集め、ディスクに納めている。さらに収集されたデータは毎月 1 回 CDROM に書き込み、保存している。

現在は、

- ネットワークプロジェクト (WIDE、TISN、JAIN 等) 間
- WIDE バックボーン (現在は WNOC-TYO と WNOC-SFC 間のみ) 上
- WIDE の国際回線上

という分類で、全てのネットワークアドレス間の、

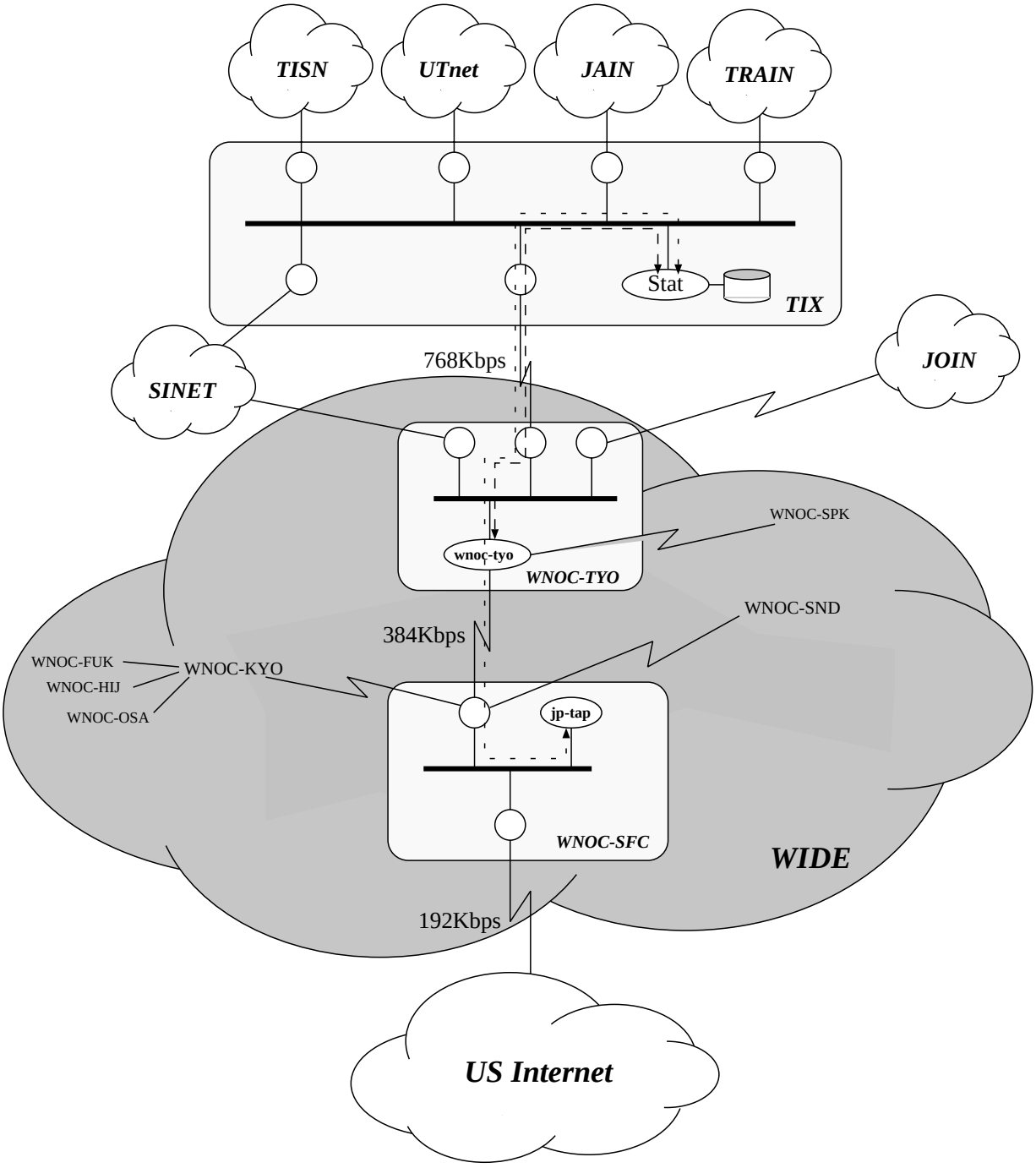

図 3.1: WIDE NetStat WG によるトラフィックデータ収集環境

- IP プロトコル別 (TCP、UDP、ICMP、IGMP 等) パケット数/バイト数
- ICMP メッセージタイプ別パケット数/バイト数
- TCP アプリケーション別 (FTP, Telnet 等) パケット数/バイト数
- UDP アプリケーション別 (DNS、NTP 等) パケット数/バイト数

に関する統計情報を、1 時間毎にまとめて集計している。

TIX 上でのトラフィックは、各ネットワークプロジェクトのネットワークを TIX に接続するために用いているルータのイーサネットアドレスを用いて、トラフィックを分類した。

以下の節では、1992 年 6 月から 1993 年 3 月までに収集されたデータに対する解析結果について述べる。

3.2　TIX における情報交換

まず始めに、図 3.2は、1993 年 2 月の各ネットワーク間で交換された情報量 (byte) を、各ネットワークのマトリックスと、TIX と交換した情報量とでグラフ化したものである。

SINET との情報交換は、元時点では、TIX としてまとめた情報しかサンプルしていないため、各ネットワークとの情報交換した値となっていない。

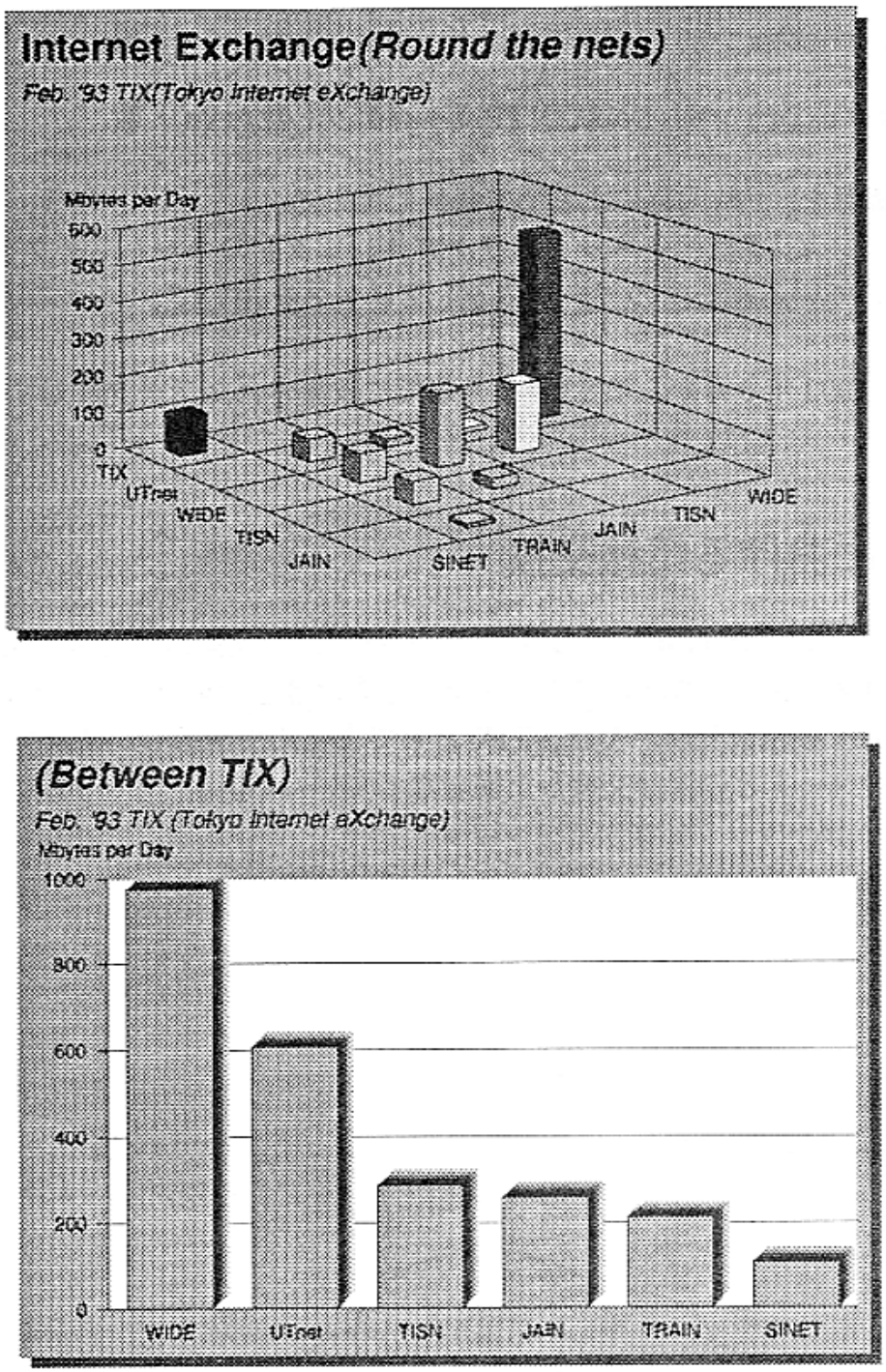

図 3.2: TIX におけるネットワークプロジェクト間のトラフィック

3.2.1　92 年度のトラフィック変動

以下に示す情報は、すべて nnstat を用いて収集したデータで、月別に一日あたり交換された情報量の変化をグラフ化している。括弧内の値は、各月で情報が収集出来た日数である。

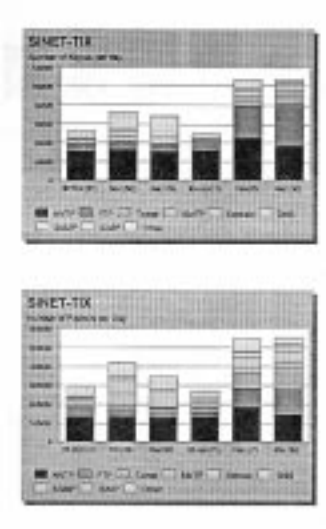

図 3.3: SINET と TIX の間のトラフィック

3.3　WONC-TYO – WNOC-SFC のトラフィック

WIDE のバックボーン中最も利用されている WNOC-TYO と WNOC-SFC 間の昨年 1 年間のトラフィックを以下に示す。

3.4　国際線のトラフィック

3.4.1　IP プロトコル別トラフィック

表 3.1は、1992 年 6 月から 1993 年 3 月までの IP プロトコル別のトラフィック量をキロバイト単位で表したものである。

一番左側の欄の括弧の中の日数は、各月に 1 日のデータが完全に収集できた日数である[1]。これを有効日数と呼ぶことにする。合計の欄が各月に国際線上を転送された IP データグラムの総データ量をあらわしている[2]。また、一日平均の欄は、月の総データ量をそ

[1]データを蓄積したディスクの容量や、collect を動作させたホストの状態によって、完全なデータが収集できないこと日があった。

[2]ただし、このデータ量にはデータリンクレベルのフレームヘッダの分は含んでいない。

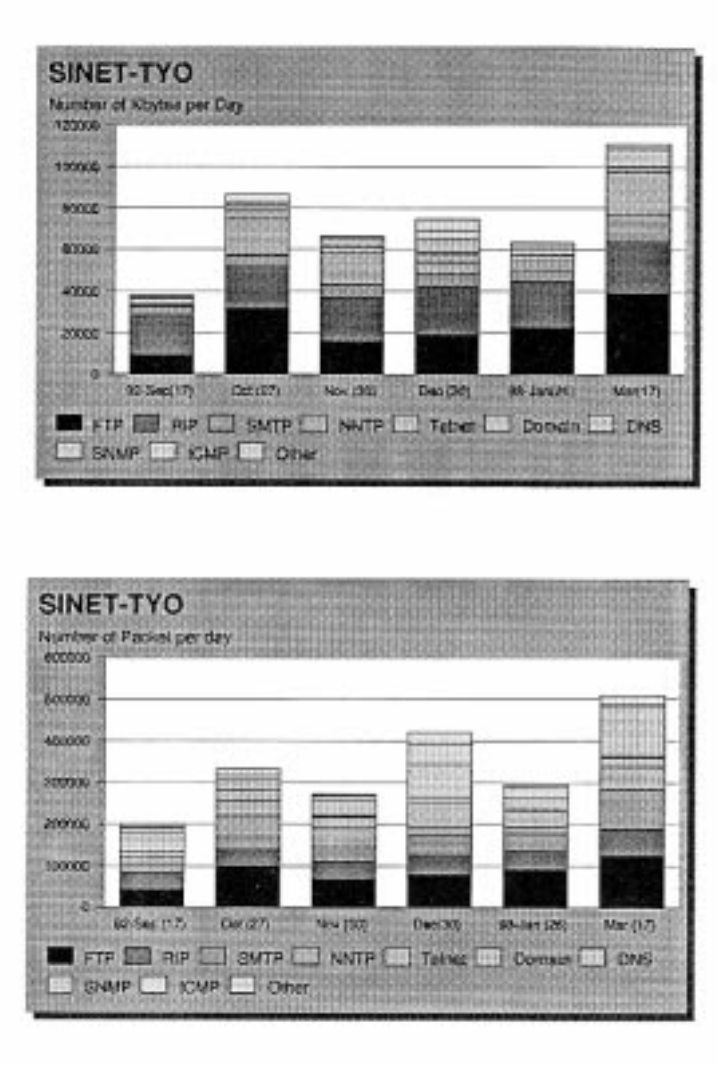

図 3.4: SINET と WNOC-TYO の間のトラフィック

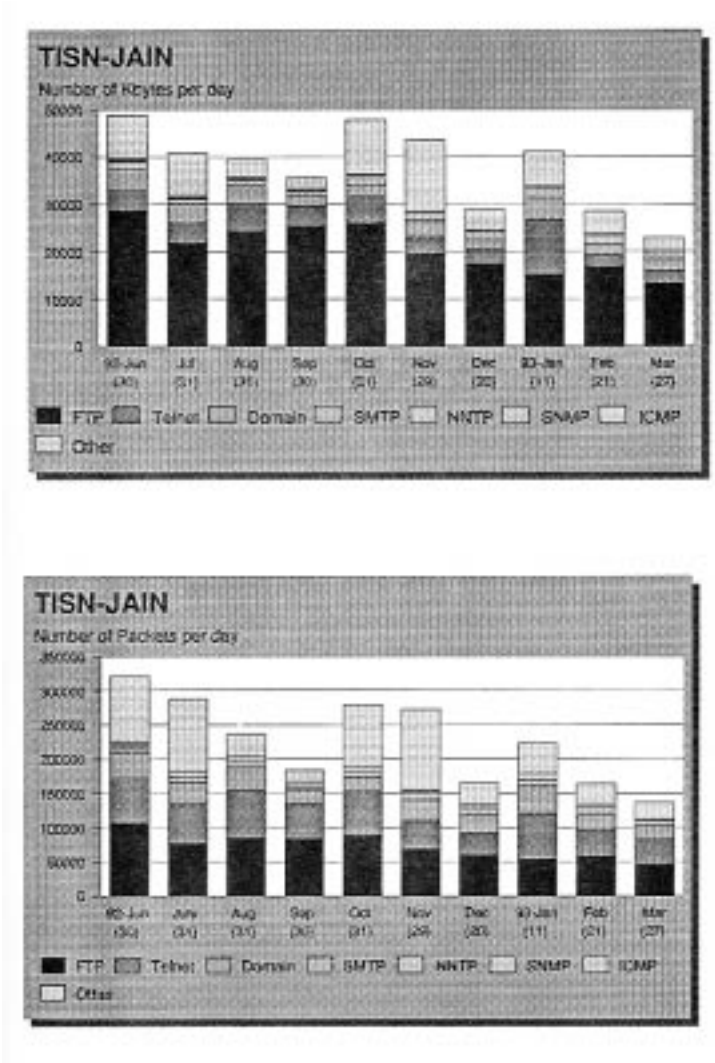

図 3.5: TISN と JAIN の間のトラフィック

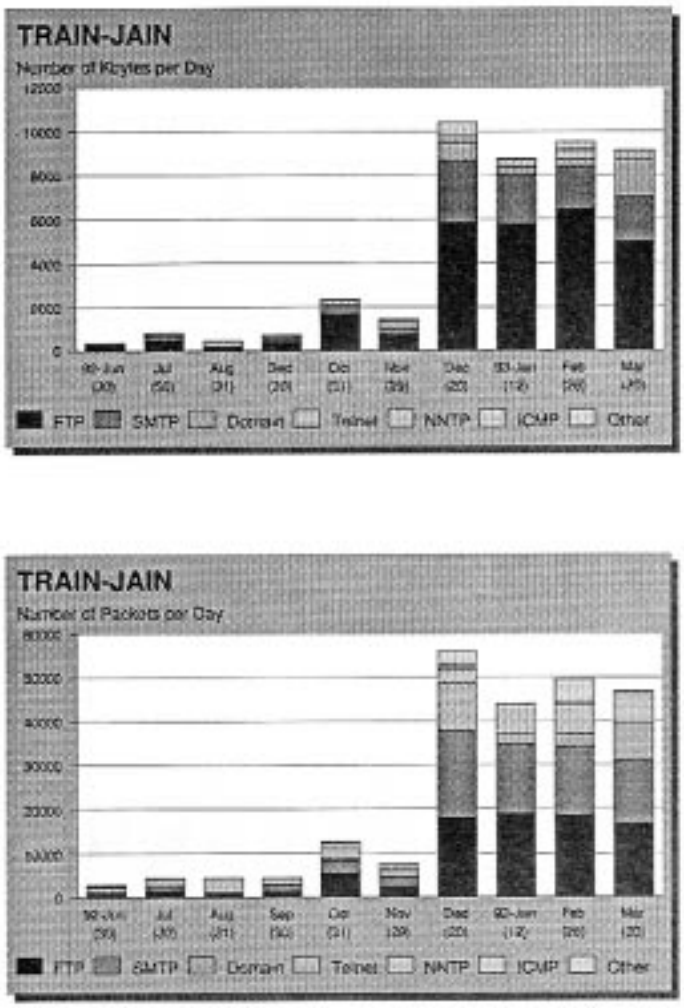

図 3.6: TRAIN と JAIN の間のトラフィック

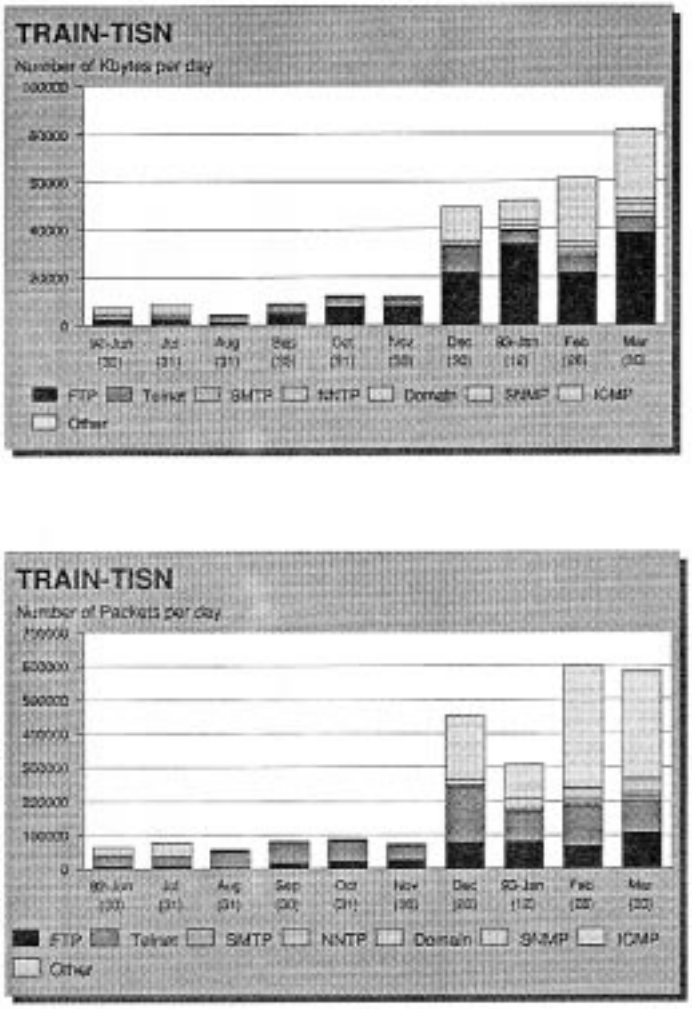

図 3.7: TRAIN と TISN の間のトラフィック

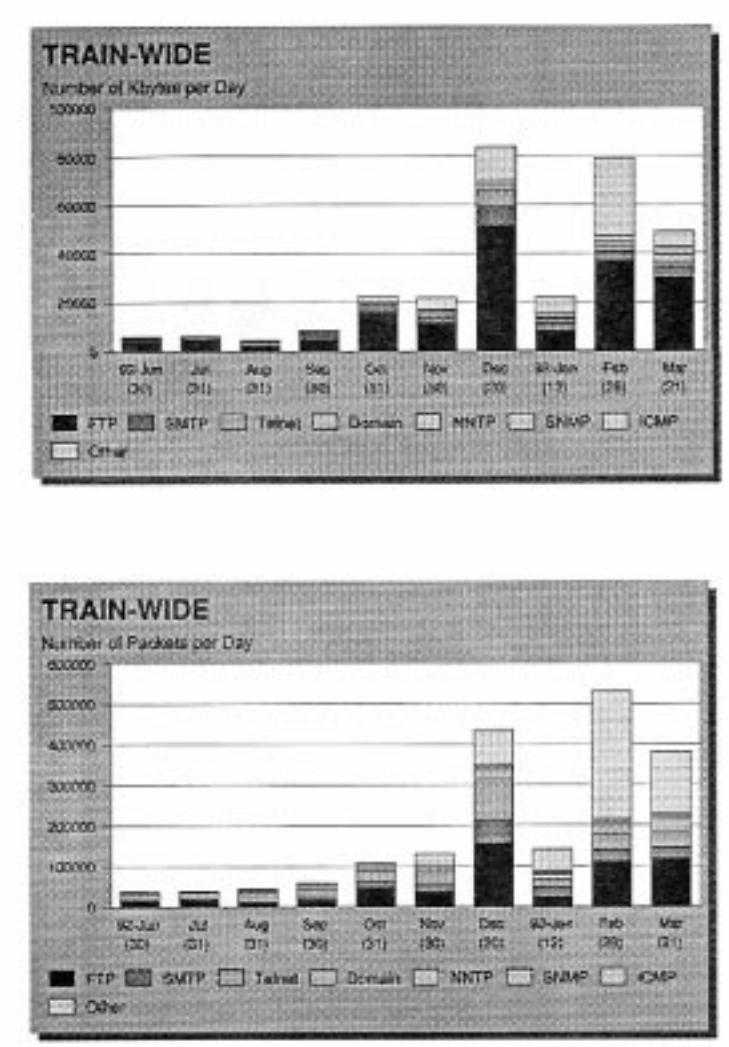

図 3.8: TRAIN と WIDE の間のトラフィック

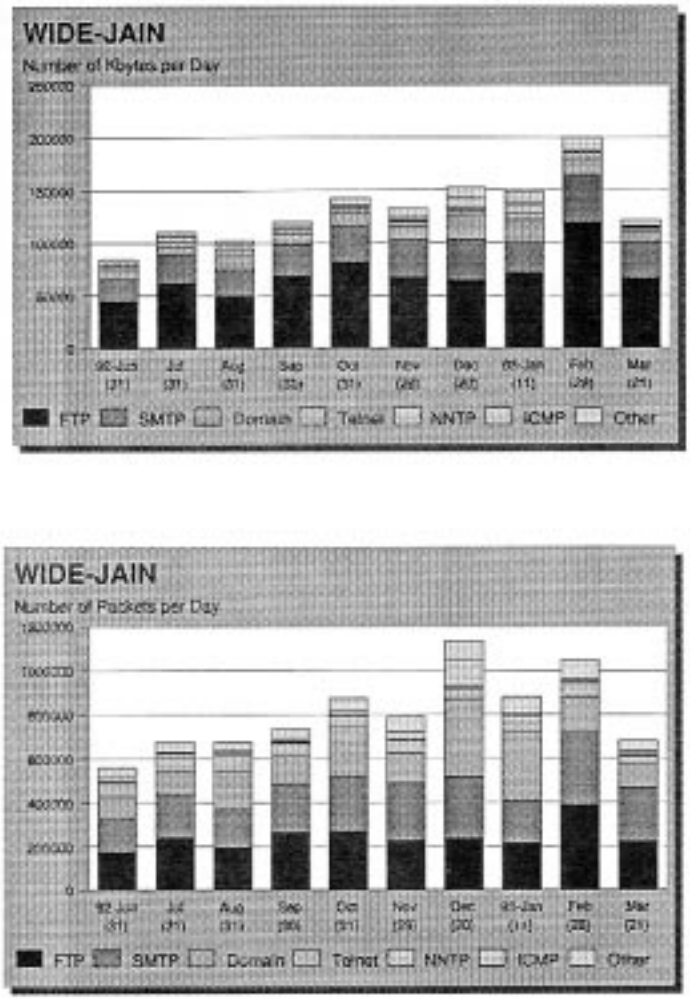

図 3.9: WIDE と JAIN の間のトラフィック

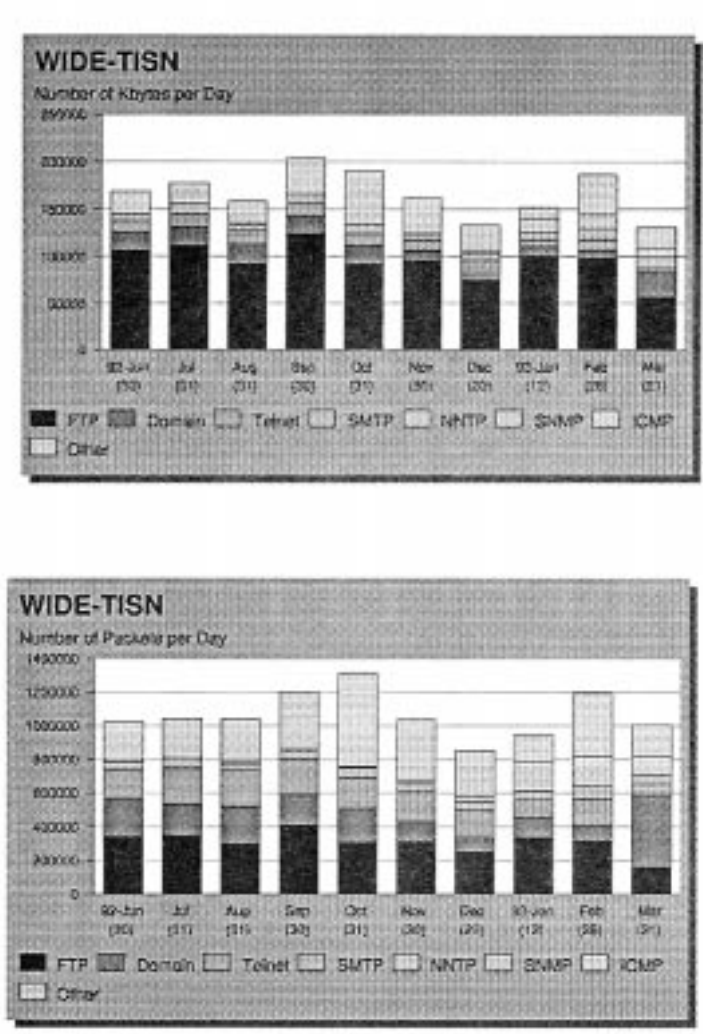

図 3.10: WIDE と TISN の間のトラフィック

の月の有効日数で割った値である。一番右の欄は、192Kbps のバンド幅をもつ国際回線上を有効日数の間に転送する事ができる最大のデータ量に対する月の総データ量の占める割合である。従って、この値は各月の 1ヵ月間の平均回線利用率を表している。

図 3.12は、表 3.1をもとに、1992 年 6 月から 1993 年 3 月までの全 IP トラフィックのうち、各プロトコルのデータ量の占める割合を表したグラフである。どちらの方向に関しても TCP のデータ量が全体の約 86%を占めている事が分かる。

図 3.13、図 3.14はそれぞれ国外から国内向け、国内から国外向けのプロトコル別の毎月のトラフィック量を、各月の 1 日平均のデータ量の推移により表したものである。

グラフより、全体の傾向としてどちらも増加の傾向にあることが分かる。1993 年 3 月になると、国外から国内向けのトラフィックは、1ヵ月の平均で 52.6%、国内から国外向けのトラフィクは 43.4%となり、どちらも最高の利用率となっている。1992 年 6 月から 1993 年 3 月にかけて、一日の平均トラフィック量は、国外から国外向けが 2.5 倍、国内から国外向けが 2.6 倍と増加している。グラフの示す傾向から今後更に増加していくことが予想される。また 1992 年 11 月と 1993 年 3 月はほかの月と比較して UDP のトラフィックの占める割合が多くなっているが、これは各々の月の行われた IETF の Audiocast の実験によるものと考えられる。

以下の節では各 TCP、UDP、ICMP の各プロトコルに関して解析を行った結果に付いて述べる。

3.4.2 TCP アプリケーション別トラフィック

表 3.2は、TCP アプリケーション別のトラフィック量を各月の一日平均のキロバイト数で表したものである。

図 3.15は、表 3.2をもとに、1992 年 6 月から 1993 年 3 月までの全 TCP トラフィックのうち、主要なアプリケーションに関するデータ量の占める割合を表したグラフである。どちらの方向に関しても FTP が TCP の全データ量の 70%以上を占めている。

図 3.16と図 3.17は、それぞれ国外から国内向け、国内から国外向けの TCP アプリケーション別の毎月のトラフィック量を、各月の 1 日平均のデータ量の推移により表したものである。また、図 3.18と図 3.19は、1992 年 6 月の 1 日あたりのデータ量を 1 とした、アプリケーション別のデータ量推移の傾向を表すグラフである。

全体の傾向として国外から国内向けの TCP のトラフィックは約 3 倍、国内から国外向けの TCP のトラフィックは約 4 倍になっている。各アプリケーションに関しては、FTP、SMTP、NNTP などのデータ転送型のアプリケーションのトラフィックは Telnet や login のようなインタラクティブなアプリケーションに比べて増加の傾向が強い。また、どちらの方向のトラフィックに関しても、Other のカテゴリーに入るプロトコルのトラフィックが増えている。これはインターネット上で提供されるサービスの種類の広がりを示唆しているといえよう。

3.4.3 UDP アプリケーション別トラフィック

表 3.3は、UDP アプリケーション別のトラフィック量を各月の一日平均のキロバイト数で表したものである。

図 3.20は、表 3.3をもとに、1992 年 6 月から 1993 年 3 月までの全 TCP トラフィックのうち、主要なアプリケーションに関するデータ量の占める割合を表したグラフである。主要なアプリケーションの内 Domain のトラフィックは、国外から国内向きでは、UDP の全データ量の約 30%、国内から国外向きでは約 50%を占めている。

図 3.21と図 3.22は、それぞれ国外から国内向け、国内から国外向けの UDP アプリケーション別の毎月のトラフィック量を、各月の 1 日平均のデータ量の推移により表したものである。また、図 3.23と図 3.24は、1992 年 6 月の 1 日あたりのデータ量を 1 とした、アプリケーション別のデータ量推移の傾向を表すグラフである。ただし、アプリケーションによる変動が極端に異なっていたので、図 3.23では Talk/Phone および Other に対して、また図 3.24では Archie に対して、それぞれ他のものとは異なる Y 軸 (グラフの右端に表示されている) でプロットしている。

全体の傾向として、国外から国内向けの UDP トラフィックが約 5 倍に増加しているのに対して、国内から国外向けの UDP トラフィックの増加率は 2 倍にも達していない。しかし、図 3.23から分かるように、国外から国内向けの UDP トラフィックの増加傾向は、Other に分類されたアプリケーションのトラフィックの増加傾向に影響されている。とくにこの増加傾向は 1992 年 11 月および 1993 年 3 月に顕著に現れており、このトラフィックの増加はそれぞれの時期に行われた IETF の Audio Cast の実験によるものと考えられる。また、国内から国外向けのトラフィックの内、Archie によるものが著しく増加しているが、これに呼応するように、国内から国外向けの FTP のトラフィックが増加していることが図 3.22よりわかる。

3.4.4 ICMP の種類別トラフィック

表 3.4は、ICMP メッセージのトラフィック量を各月の一日平均のパケット数で表したものである。

図 3.25は、表 3.4をもとに、1992 年 6 月から 1993 年 3 月までに観測された全 ICMP パケットのうち、主な種類のパケット数の全体に占める割合を表したグラフである。

国外から国内に向かう ICMP のうち約 45%が Network Unreachable を示す ICMP メッセージであった。これは国内で何らかのルーティングエラーが起こり、特定の IP アドレスに対するルーティング情報が、経路上のルーターのルーティングテーブルから消えてしまった場合に default route にひきずられて海外に流れて行き、FIX-West のルーターで Network Unreachable となりそれに対して ICMP が送られることに起因している。また同様に、間違った IP アドレスに対してパケットを送った時にも同様のことが起こる。

国内から国外に向かう ICMP のうち約 52%が Port Unreachable を示す ICMP メッセージであった。また、約 19%が Time Exceeded in Transit を示す ICMP メッセージで、他の種類のメッセージの数と比較して多くなっている。とくに後者の発生する原因の一つとして、ルーティングループの存在が考えられるため、このメッセージが大量に発生しているような場合には注意が必要である。ただし、traceroute コマンドによっても用いられているため、Time Exceeded が発生したからといって、ルーティングループが起こっているとは一概には言えない。他の SNMP などでえられる情報と併せて判断していく必要がある。

図 3.26と図 3.27は、それぞれ国外から国内向け、国内から国外向けの ICMP の種類ごとの毎月のトラフィックを、1 日平均のパケット数で表したものである。当り前のことではあるが、図 3.26の Echo Request メッセージの数と、図 3.27の Echo Reply メッセージの数の変動 (またその逆の場合) が呼応している。また、図 3.26および図 3.27の両方とも、Net Unreachable メッセージの数が 6 月にとくに多かったことが分かる。このことから、昨年 6 月には国内のルーティングが何らかの理由で不安定であったことが予想される。また、図 3.27の、6 月の Time Exceeded メッセージの数もルーティングの不安定さを裏付けるかのように、大量に発生している。しかし、7 月、8 月、9 月にも同じ程度の数の Time Exceeded メッセージが発生している。これらが全てルーティングループに起因しているとは考えにくい。この点に関してはさらに詳しい解析が必要であろう。

図 3.27から、11 月以降に、国内から国外に向かう Port Unreachable メッセージが急激に増加していることがわかる。これは何らかの障害に起因していることが予想される。仮に、どのポートに対する Unreachable メッセージなのかが分かればある程度その原因を究明することも可能であるが、現在の NNStat によるデータ収集方法では不可能である。なぜなら、2で解説したように、NNStat では IP や ICMP のヘッダの情報をもとに統計データを記録しており、したがって ICMP のボディーとして返される、その ICMP をひきおこした IP データグラムのヘッダの情報などを用いることができない。このような解析を行うためには、tcpdump のようなツールを用いることが必要になるだろう。

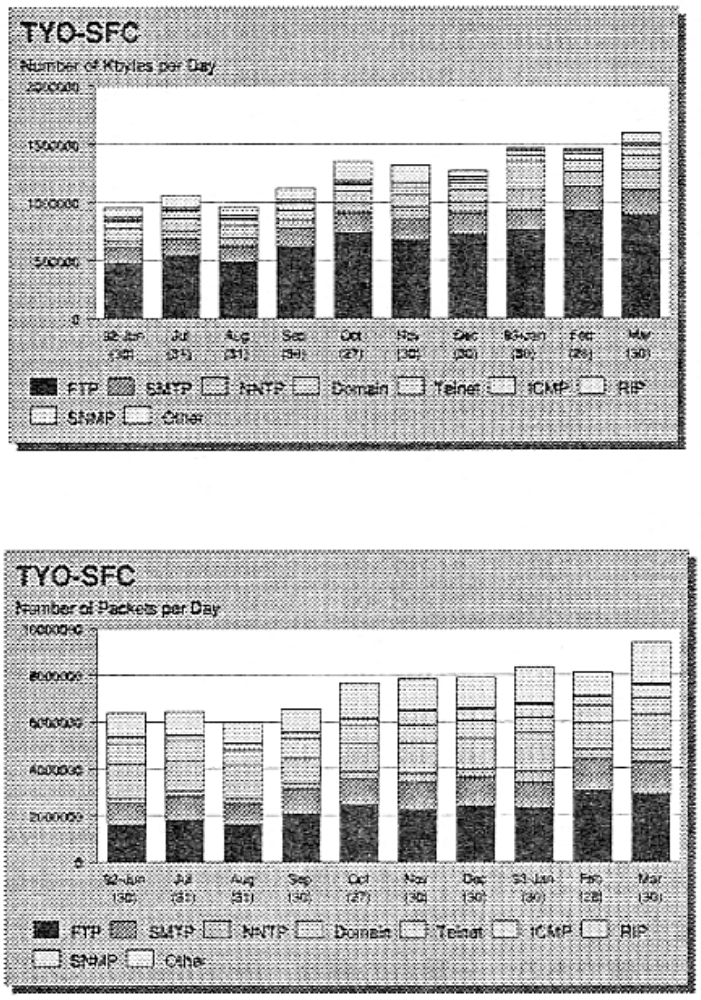

図 3.11: WNOC-TYO と WNOC-SFC 間のトラフィック

表 3.1: プロトコル別トラフィック (単位：キロバイト)

		ICMP	IGMP	TCP	UDP	合計	一日平均	回線利用率
6 月	IN	330,219	0	9,016,057	840,499	10,186,775	442,903	21.4%
(23 日)	OUT	259,574	0	6,130,023	1,440,745	7,830,348	340,450	16.4%
7 月	IN	300,219	545	14,070,767	2,043,562	16,415,092	529,519	25.5%
(31 日)	OUT	303,861	768	9,685,353	1,214,786	11,204,768	361,444	17.4%
8 月	IN	124,714	0	14,943,681	2,200,149	17,268,545	557,050	26.9%
(31 日)	OUT	171,125	0	10,952,087	1,092,845	12,216,058	394,066	19.0%
9 月	IN	80,638	0	13,201,710	1,169,489	14,451,838	498,339	24.0%
(29 日)	OUT	131,441	0	9,880,677	611,585	10,623,703	366,335	17.7%
10 月	IN	64,643	0	13,286,104	1,324,699	14,675,447	611,477	29.5%
(24 日)	OUT	188,894	0	10,178,627	1,788,799	12,156,325	506,514	24.4%
11 月	IN	212,677	19,118	18,651,562	4,926,715	23,810,071	793,669	38.3%
(30 日)	OUT	345,160	23,488	13,679,215	3,838,067	17,885,932	596,198	28.8%
12 月	IN	185,176	29,220	20,368,457	3,015,233	23,598,087	786,603	37.9%
(30 日)	OUT	438,892	34,429	12,996,192	1,246,867	14,716,381	490,546	23.7%
1 月	IN	176,433	22,012	16,985,993	1,860,617	19,045,054	680,181	32.8%
(28 日)	OUT	347,340	21,420	8,930,094	772,622	10,071,478	359,696	17.4%
2 月	IN	281,358	7,051	24,787,572	2,246,190	27,322,172	975,792	47.1%
(28 日)	OUT	569,901	9,696	16,093,806	1,519,153	18,192,562	649,734	31.3%
3 月	IN	394,293	40,073	27,957,875	4,123,046	32,700,113	1,090,004	52.6%
(30 日)	OUT	890,211	55,970	23,946,019	2,072,386	26,964,664	898,822	43.4%
合計	IN	2,150,369	118,018	173,269,777	23,750,199	199,473,194	702,370	33.9%
(284 日)	OUT	3,646,398	145,771	122,472,094	15,597,855	141,862,220	499,515	24.1%

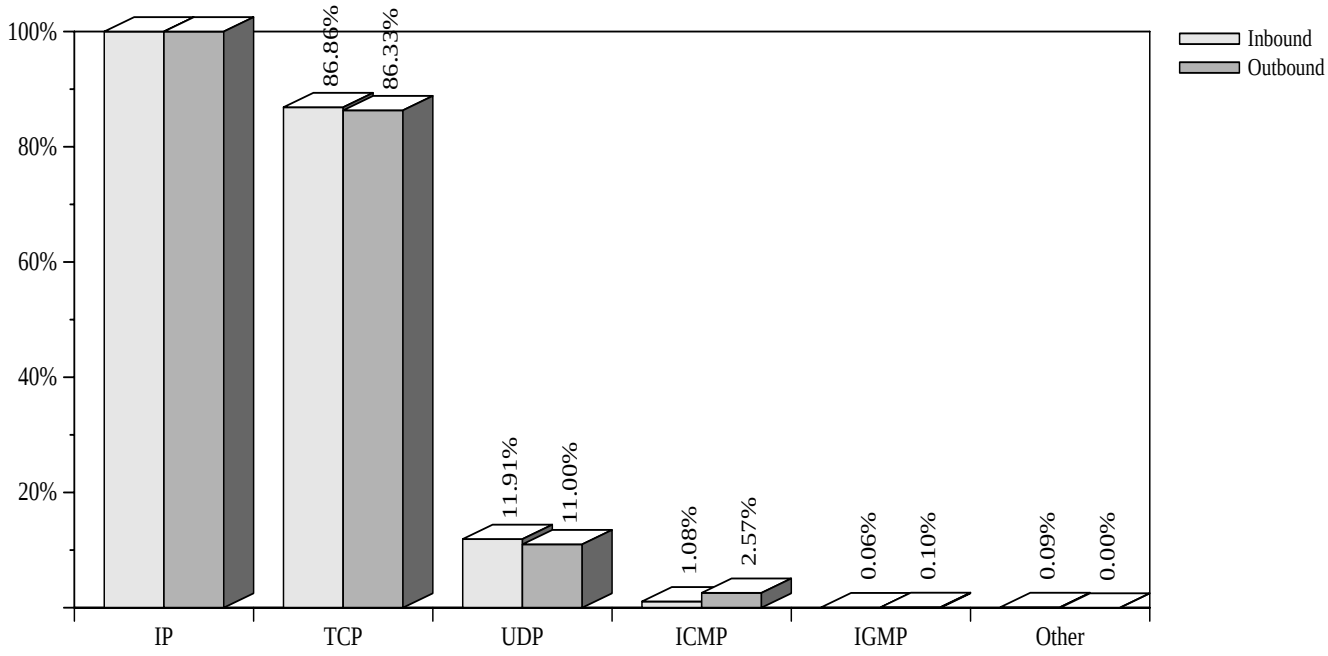

図 3.12: IP プロトコル別トラフィック量分布

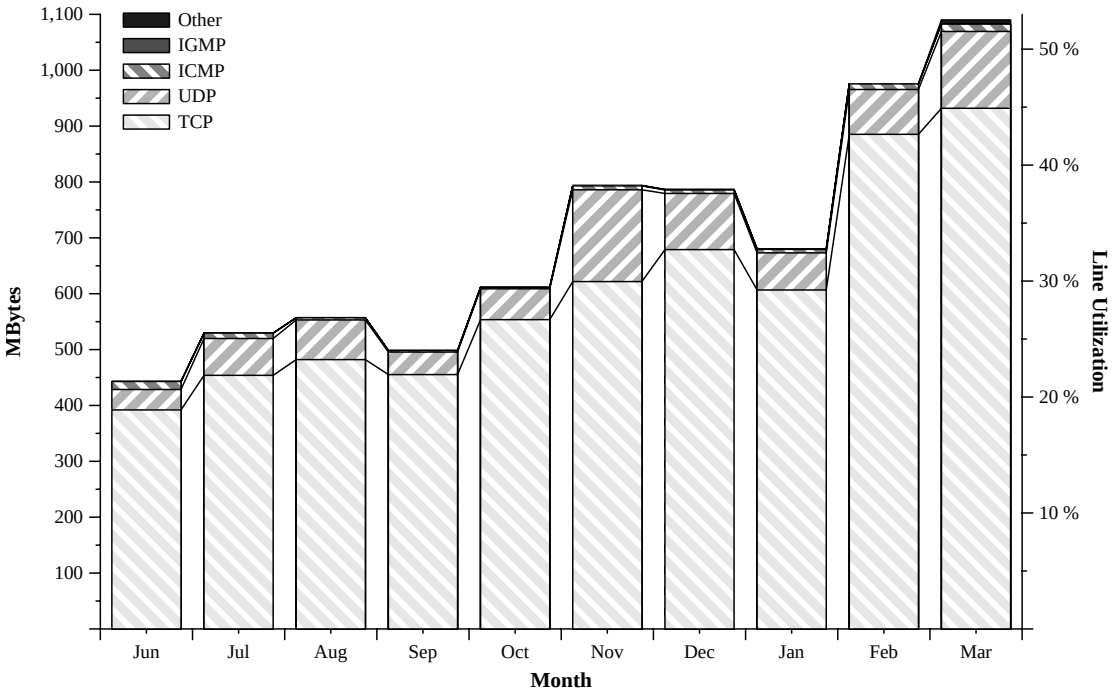

図 3.13: 国外から国内向けの、プロトコル別トラフィック量推移 (1 日平均)

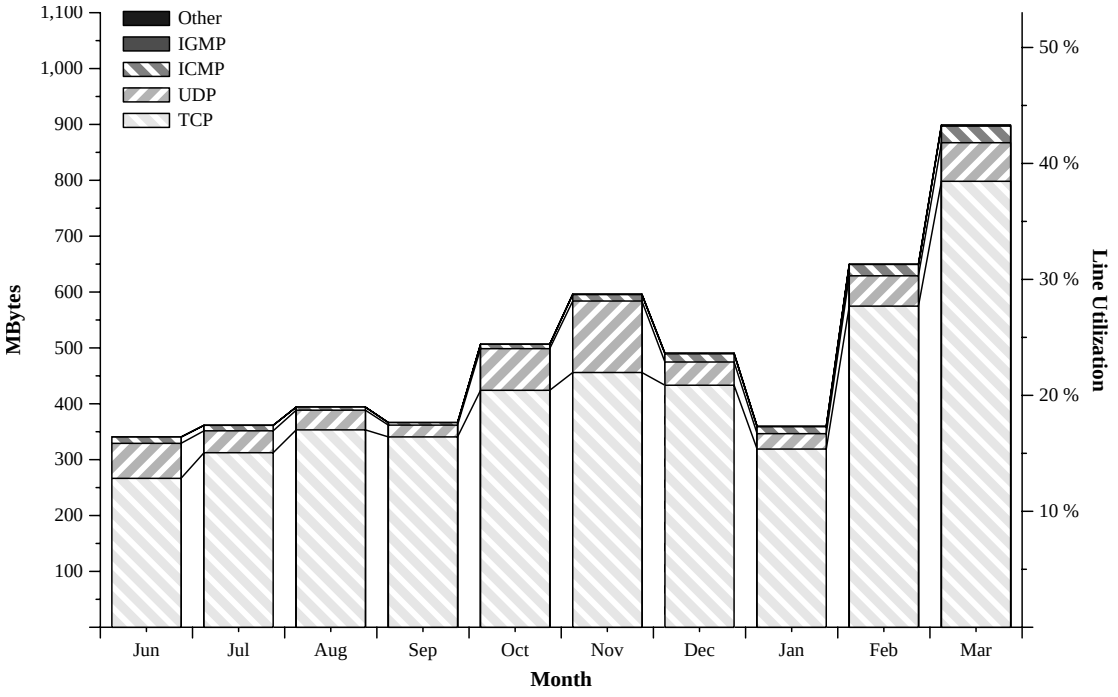

図 3.14: 国内から国外向けの、プロトコル別トラフィック推移 (1 日平均)

表 3.2: TCP アプリケーション別トラフィック (単位：キロバイト／日)

		FTP	SMTP	NNTP	Telnet	Login	Domain	Finger	Z39.50	Whois	その他
6 月	IN	303,392	35,785	13,959	19,411	3,701	4,009	285	291	79	11,090
	OUT	193,648	23,698	19,493	18,077	4,737	759	271	29	52	5,760
7 月	IN	346,111	44,073	17,206	22,824	4,116	4,240	492	185	61	14,589
	OUT	221,895	32,423	17,702	20,646	5,743	552	489	15	28	12,937
8 月	IN	360,439	57,686	20,269	21,898	3,128	3,744	323	193	36	14,337
	OUT	261,992	27,762	18,465	27,841	4,908	481	253	20	16	11,556
9 月	IN	353,977	43,078	25,165	13,119	2,874	3,586	383	459	47	12,543
	OUT	270,943	24,779	12,392	18,968	3,530	506	331	35	15	9,215
10 月	IN	446,506	46,985	20,715	15,690	3,015	807	1,077	144	187	18,462
	OUT	346,426	33,272	13,141	15,368	4,166	1,045	501	19	38	10,133
11 月	IN	478,120	60,434	36,915	17,184	4,310	769	556	678	93	22,658
	OUT	358,778	35,801	16,892	18,357	6,563	963	488	69	34	18,029
12 月	IN	519,335	54,611	44,257	15,368	5,293	884	722	417	138	37,924
	OUT	321,276	43,897	18,724	19,720	6,132	1,336	572	48	75	21,426
1 月	IN	442,304	46,683	77,383	13,376	3,535	775	511	173	152	21,750
	OUT	233,339	30,229	13,724	14,854	3,608	1,100	354	18	54	21,652
2 月	IN	648,657	72,951	89,194	20,671	4,116	2,234	669	595	114	46,068
	OUT	401,272	75,043	29,252	21,833	6,221	1,775	688	68	143	38,483
3 月	IN	674,162	83,405	94,385	23,138	3,829	2,429	595	419	116	49,451
	OUT	569,538	84,377	39,877	28,163	7,965	2,760	841	97	142	64,440
平均	IN	459,711	55,133	44,509	18,359	3,812	2,358	556	360	100	25,206
	OUT	320,010	41,542	20,114	20,606	5,410	1,132	483	42	60	21,841

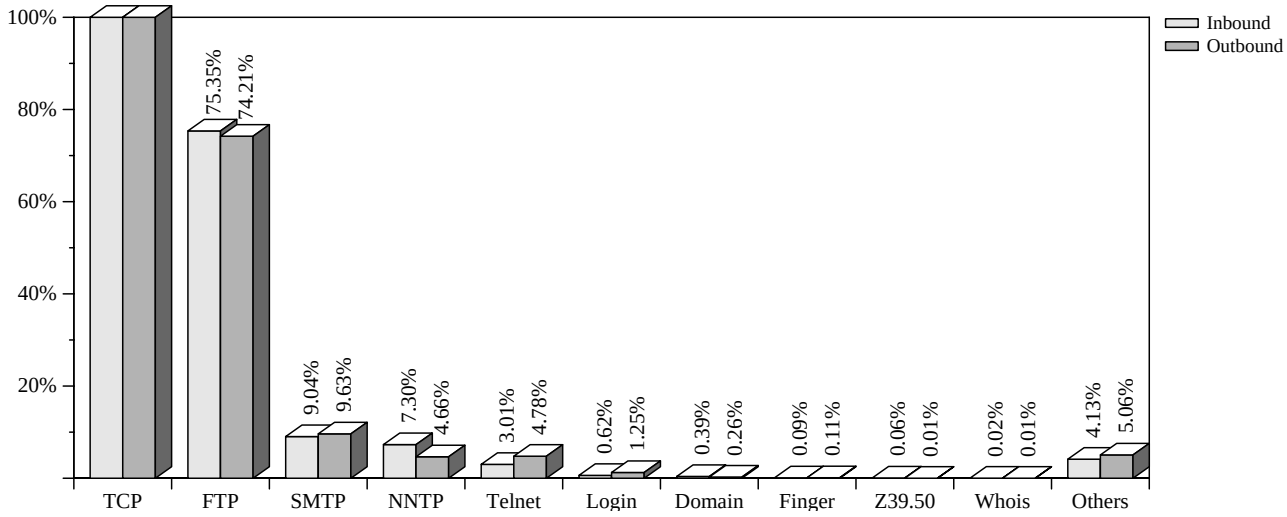

図 3.15: TCP アプリケーション別トラフィック量分布

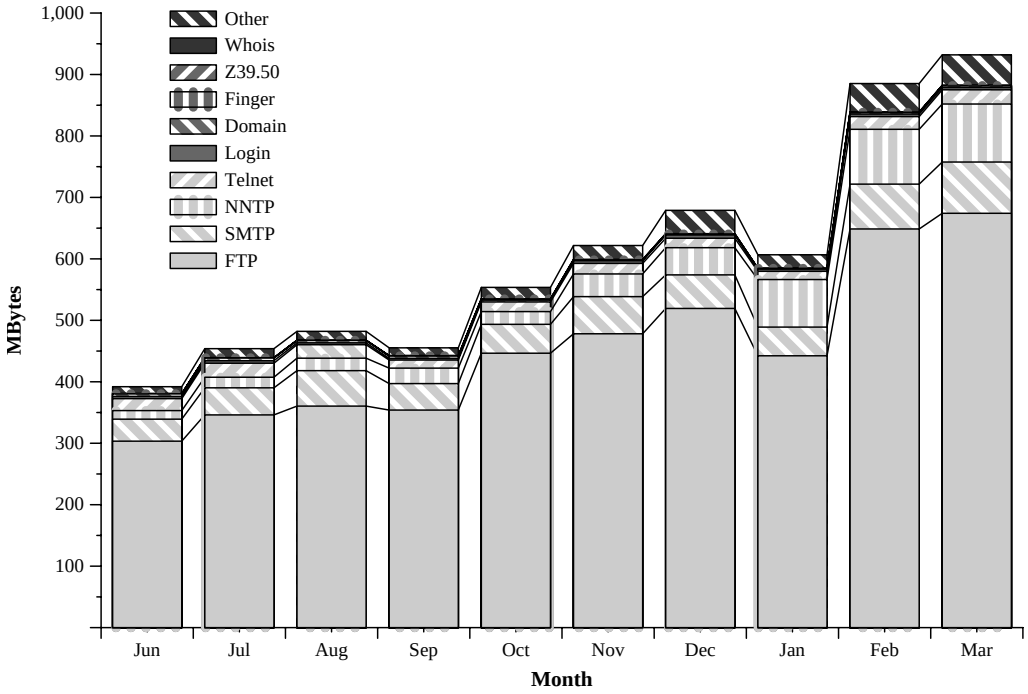

図 3.16: 国外から国内向けの TCP アプリケーション別トラフィック量推移 (1 日平均)

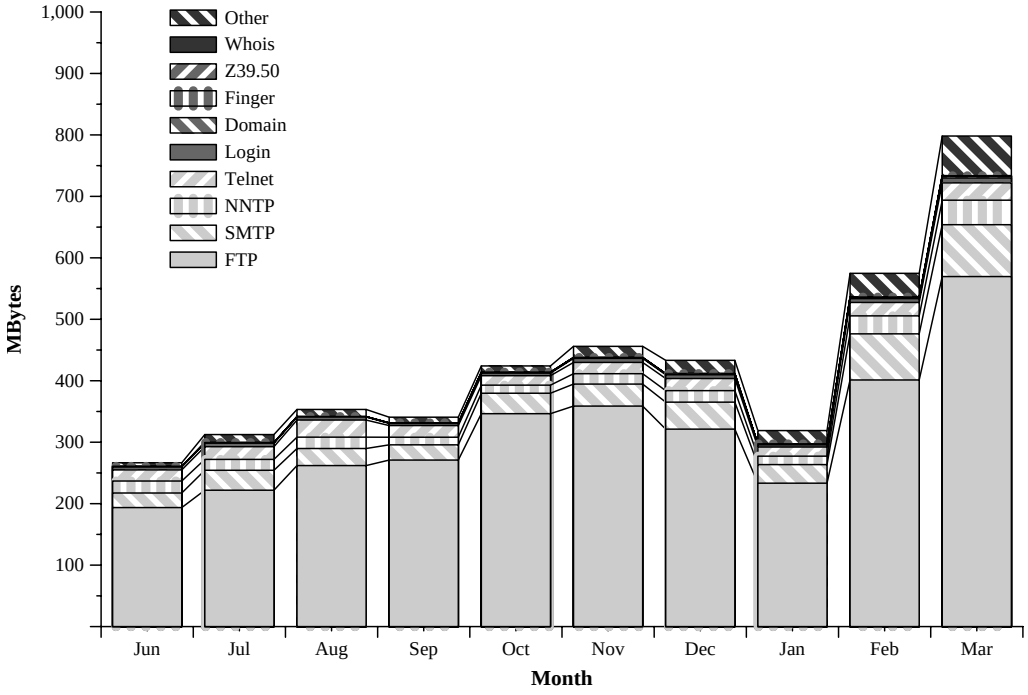

図 3.17: 国内から国外向けの TCP アプリケーション別トラフィック量推移 (1 日平均)

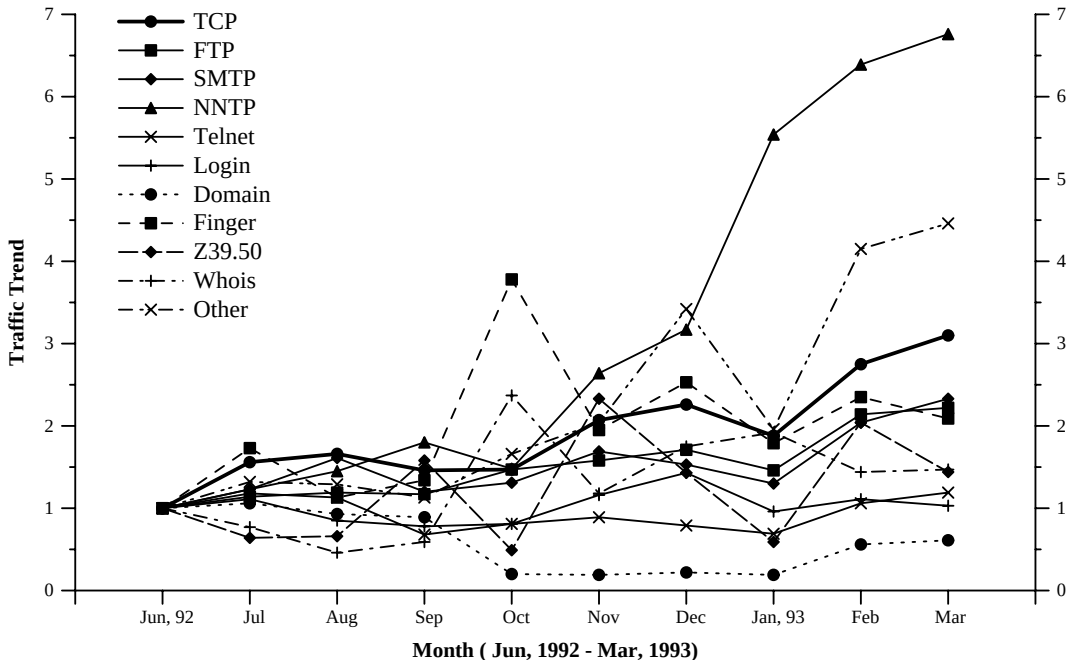

図 3.18: 国外から国内向きの TCP アプリケーション毎のトラフィックの傾向

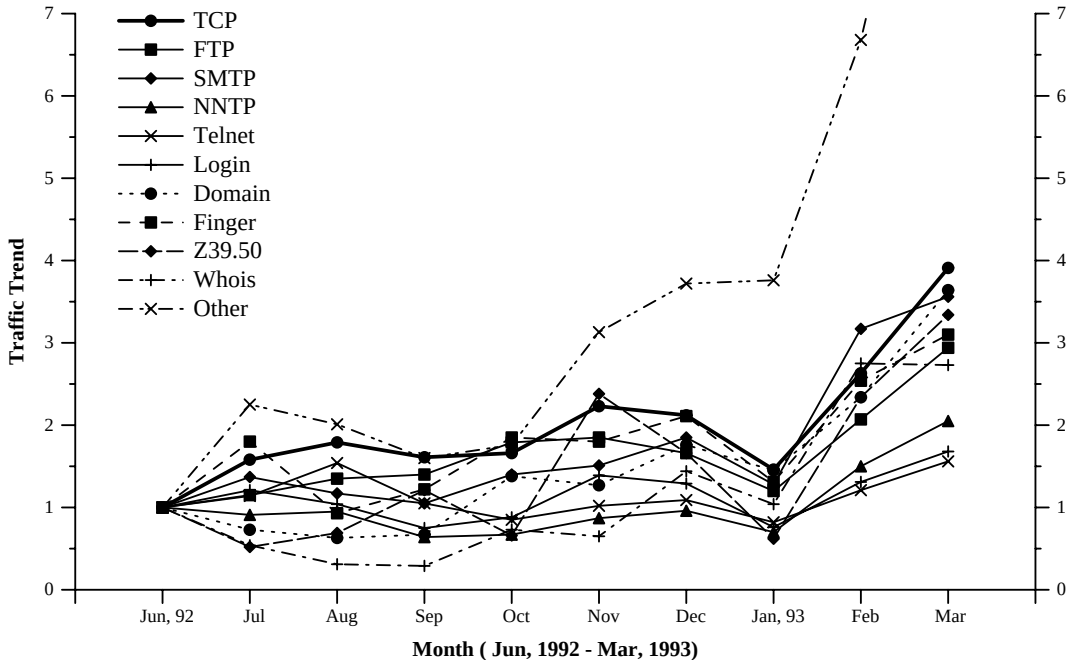

図 3.19: 国内から国外向きの TCP アプリケーション毎のトラフィックの傾向

表 3.3: UDP アプリケーション別トラフィック (単位：キロバイト／日)

		Domain	NTP	Archie	Talk/Phone	SNMP	その他
6 月	IN	26,013	1,026	1,291	294	556	7,364
	OUT	53,272	1,563	29	3,157	266	4,354
7 月	IN	30,286	1,269	1,467	192	353	32,354
	OUT	29,873	1,594	40	215	26	7,438
8 月	IN	26,585	1,418	1,067	162	261	41,480
	OUT	21,893	1,590	26	131	112	11,501
9 月	IN	9,091	1,232	1,208	278	210	28,308
	OUT	13,447	1,518	24	234	25	5,842
10 月	IN	26,734	1,154	1,590	621	186	24,912
	OUT	32,362	1,285	350	411	14	40,111
11 月	IN	19,059	1,486	821	588	166	142,103
	OUT	20,097	1,210	610	513	1	105,505
12 月	IN	28,945	1,516	1,132	1,820	79	67,015
	OUT	24,860	1,213	1,336	301	6	13,846
1 月	IN	19,840	1,465	1,556	155	329	43,107
	OUT	17,522	1,214	984	495	310	7,069
2 月	IN	26,207	1,605	1,148	5,926	310	45,024
	OUT	26,791	1,239	4,621	761	385	20,459
3 月	IN	34,226	1,363	1,444	949	238	99,215
	OUT	38,397	1,243	5,006	891	534	23,010
平均	IN	24,728	1,364	1,264	1,097	263	54,911
	OUT	27,247	1,367	1,329	656	166	24,158

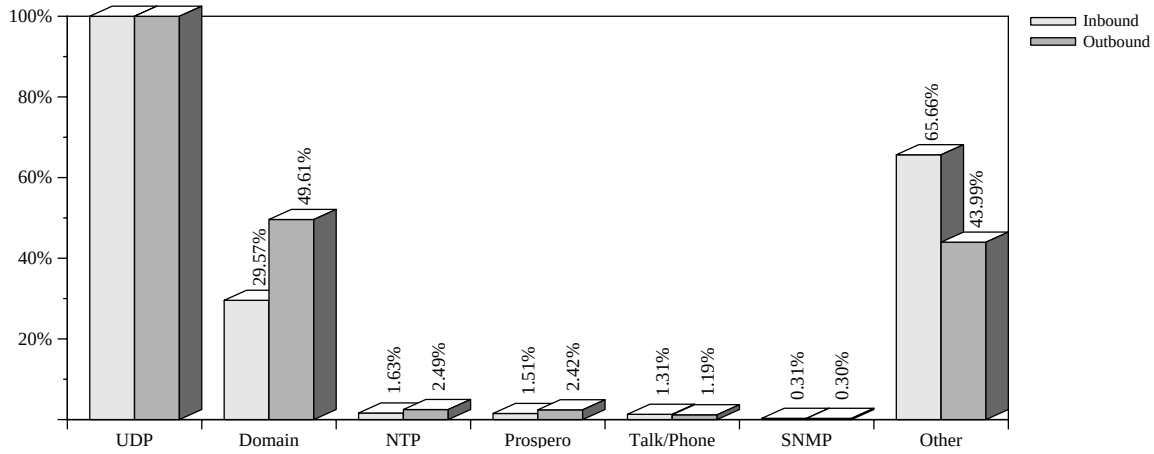

図 3.20: UDP アプリケーション別トラフィック量分布

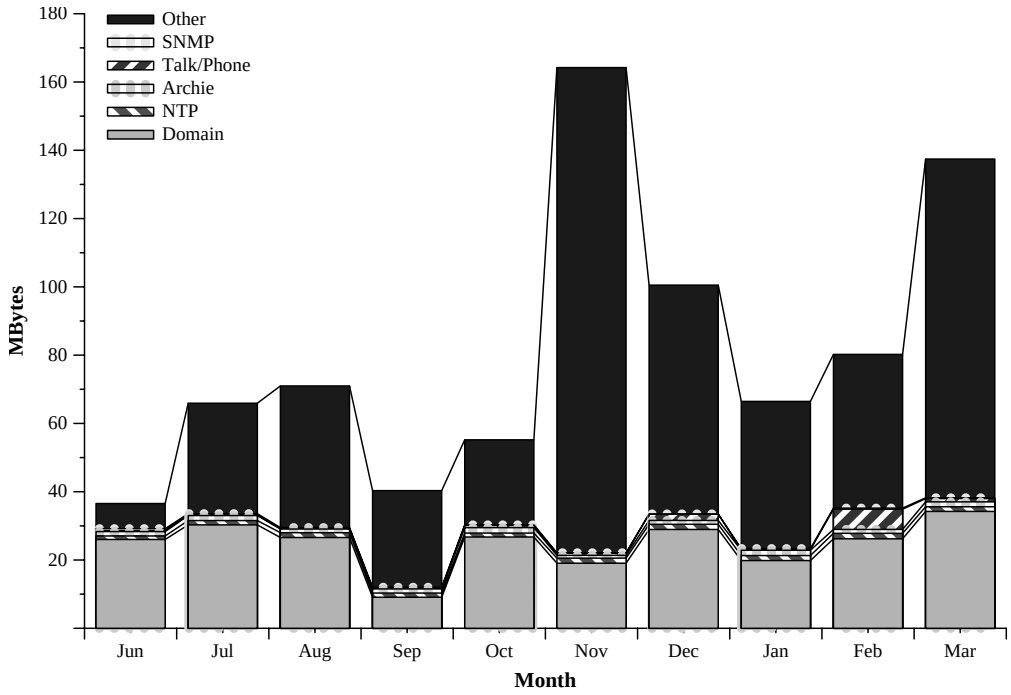

図 3.21: 国外から国内向けの UDP アプリケーション別トラフィック量推移

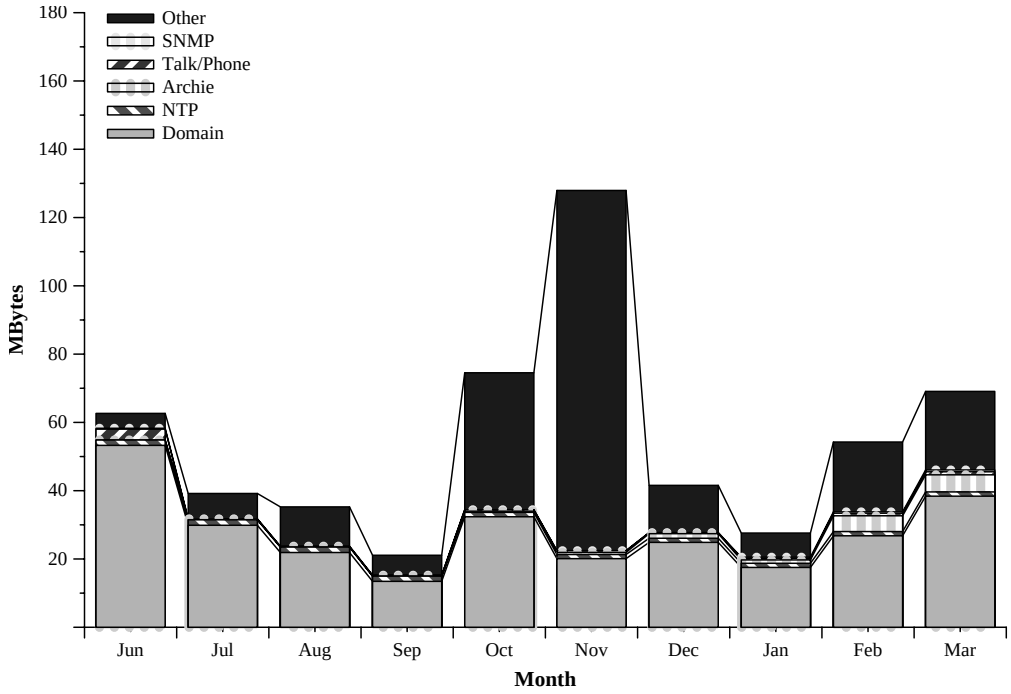

図 3.22: 国内から国外向けの UDP アプリケーション別トラフィック量推移

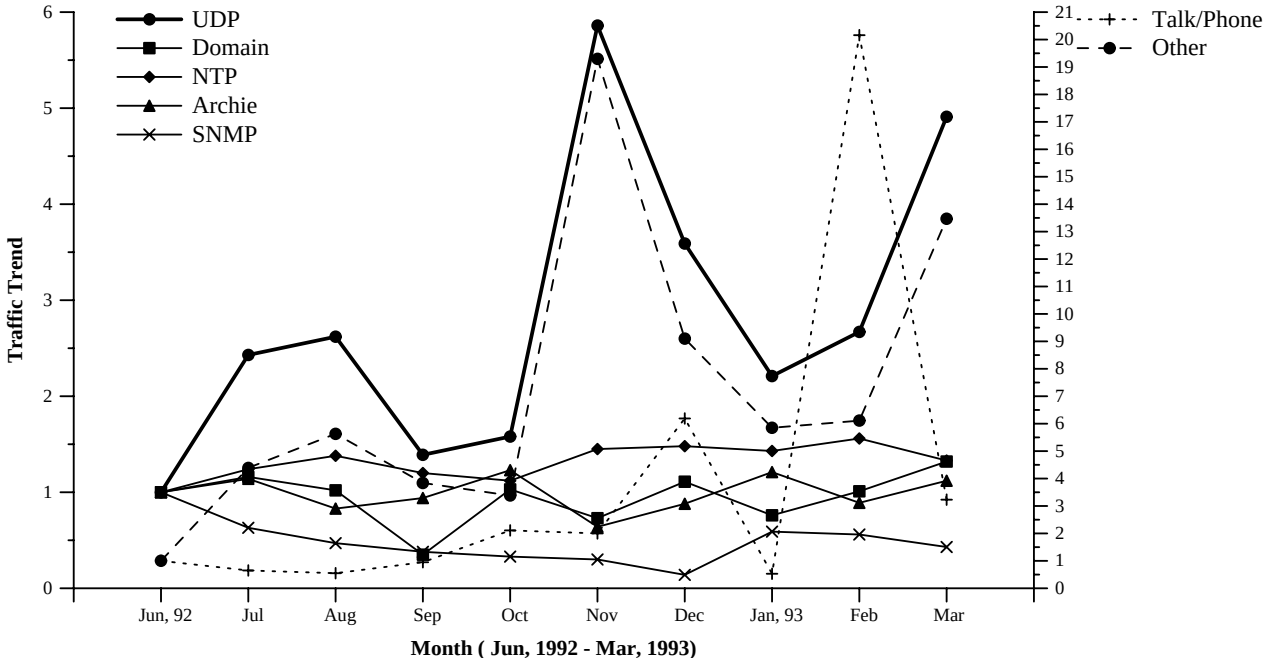

図 3.23: 国外から国内向きの UDP アプリケーション毎のトラフィックの傾向

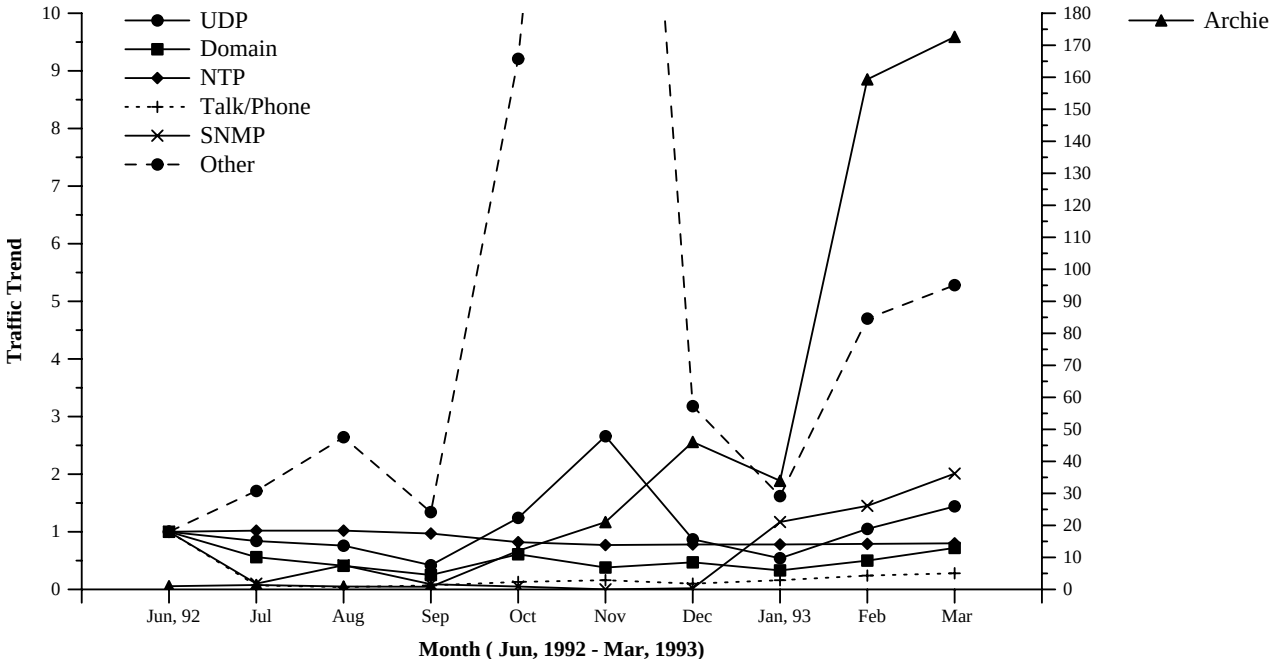

図 3.24: 国内から国外向きの UDP アプリケーション毎のトラフィックの傾向

表 3.4: ICMP の種類別トラフィック (単位：パケット数／日)

		Echo Reply	Net Unreach	Host Unreach	Port Unreach	Source Quench	Redirect Host	Echo Request	Time Exceeded in Transit	Other
6 月	IN	6,630	196,372	11,087	23,971	1,744	33	10,059	3,700	15
	OUT	7,715	89,892	1,779	6,199	247	0	8,762	82,942	58
7 月	IN	21,712	62,939	5,681	14,440	3,241	33	36,315	3,663	2
	OUT	37,273	2,562	397	1,434	746	1	27,440	76,949	77
8 月	IN	4,505	51,745	3,103	4,637	1,292	72	1,173	4,144	5
	OUT	799	3,970	1,001	1,761	199	86	16,135	68,684	45
9 月	IN	4,246	21,984	2,811	6,446	931	290	1,966	4,822	125
	OUT	1,128	4,734	2,228	5,781	82	1	6,419	56,575	199
10 月	IN	5,873	15,014	3,423	9,928	1,441	340	3,008	4,900	187
	OUT	702	1,637	105,479	9,860	214	4	7,608	13,341	27
11 月	IN	24,168	40,845	4,581	25,670	3,900	1,364	5,102	6,909	18
	OUT	668	8,810	53,353	83,278	100	2	38,432	4,106	89
12 月	IN	8,285	41,669	16,424	22,201	1,737	70	6,113	7,668	5
	OUT	846	2,257	8,719	232,878	105	1	7,197	11,432	353
1 月	IN	4,373	53,446	11,140	18,169	2,124	11,774	2,769	10,742	289
	OUT	326	7,748	25,544	146,475	76	2	6,930	31,809	20
2 月	IN	3,295	47,760	14,753	40,675	3,573	20,879	22,285	18,036	44
	OUT	972	12,011	942	285,323	172	1	27,846	20,487	35
3 月	IN	5,901	61,436	12,842	35,876	2,916	29,032	48,024	16,358	37
	OUT	59,948	26,488	5,128	323,610	202	392	37,653	40,296	40
平均	IN	9,136	57,119	8,561	20,169	2,318	6,510	14,089	8,163	70
	OUT	11,576	14,530	19,148	112,427	217	52	18,991	40,572	97

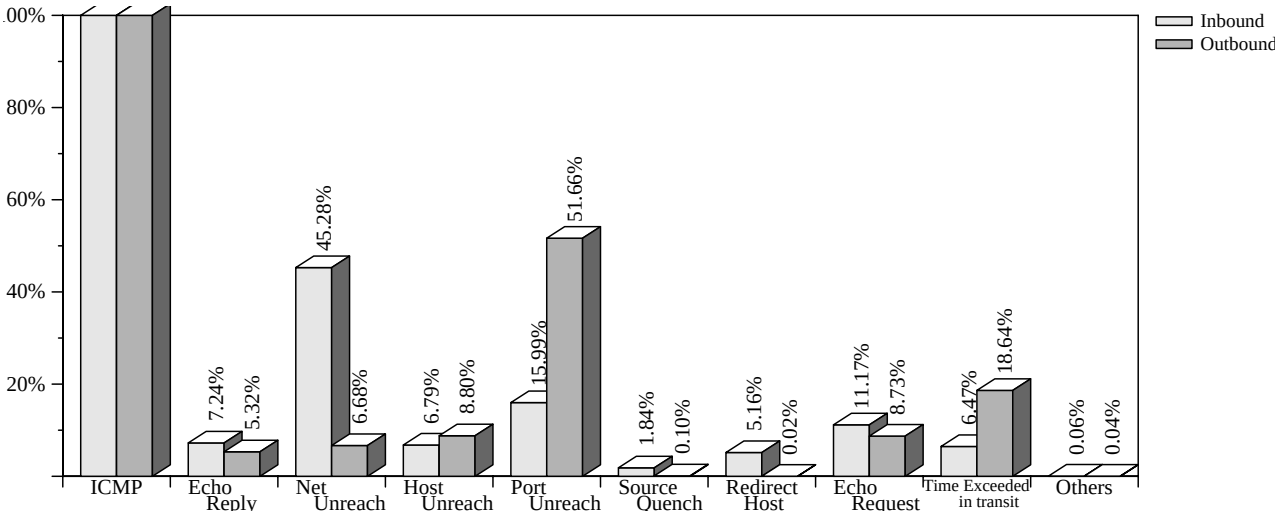

図 3.25: ICMP 種類別メッセージ量分布

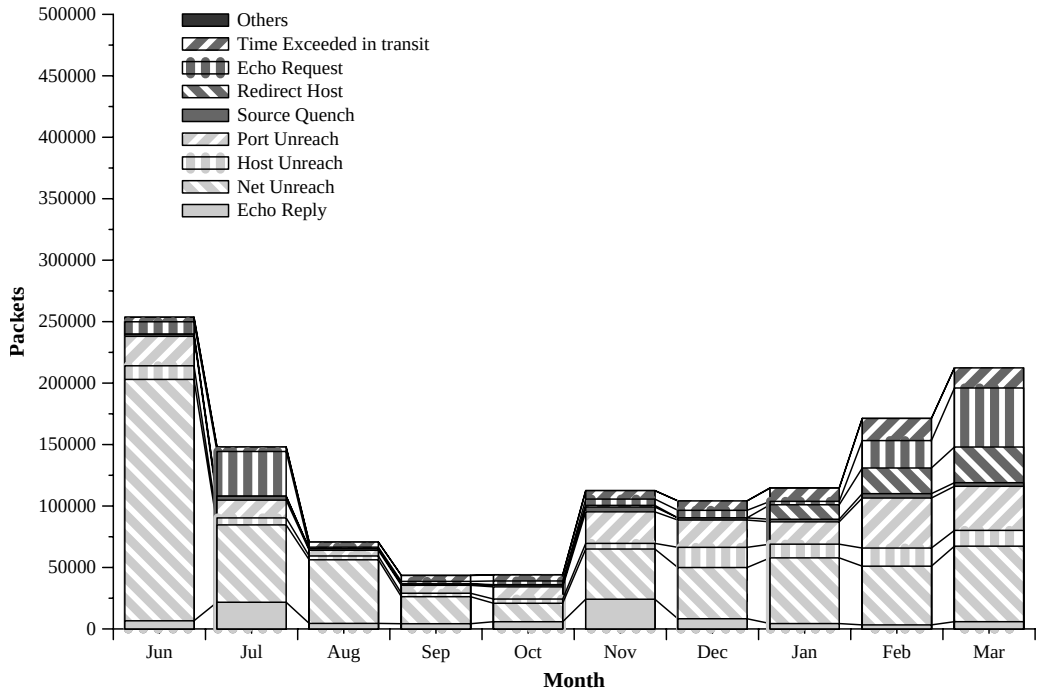

図 3.26: 国外から国内向けの ICMP の種類別トラフィック量推移 (パケット数／日)

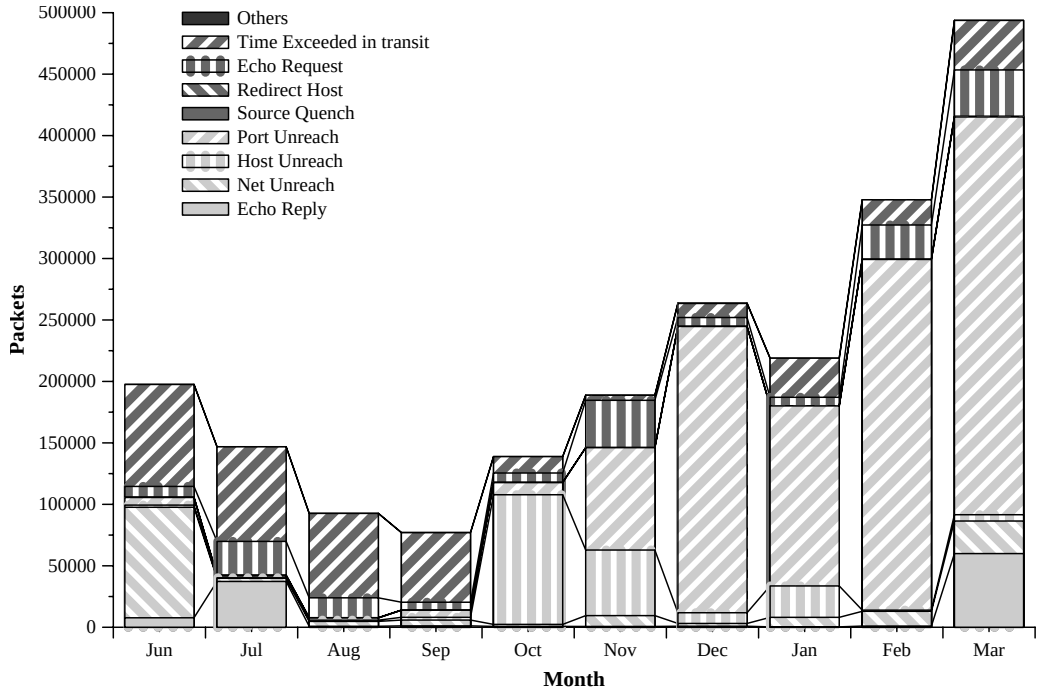

図 3.27: 国内から国外向けの ICMP の種類別トラフィック量推移 (パケット数／日)

第 4 章

RFC1404:A Model for Common Operational Statistics

本章では、RFC1404[97] で提案されている、Internet に於ける operational statistics の為のモデルについてまとめる。この RFC では、尺度、測定方法、ポーリング期間、格納時のフォーマット、そして表示時のフォーマットについて推薦されているものが示されている。

4.1 序章

現在、多くのネットワークの管理において、ネットワークの利用や成長を表すネットワーク管理の尺度 (metric) を収集、格納することは珍しいことではない。その主な目的は、組織内の短期間の問題の分離や長期間のネットワークの計画を容易にすることである。ネットワーク管理者の間での、共同な問題の分離やネットワークの計画といったより大きな目的も存在する。

ネットワーク管理の尺度の収集や表示を行なうためのネットワーク管理ツールはいろいろあるが、測定や表示の方法が異なっているので、ネットワーク同士のデータを比較することが難しくなっている。それに加えて、どの尺度を常に集めるか、それらをどのように表すかについての共通の合意がない。

以下のような条件に一致したモデルが必要となる。

1) 上記に述べた目的を満足させるための共通なネットワーク管理尺度の最小セット

2) これらの尺度を収集するためのツール

3) 共通な表示ツールによってこれらのデータを簡単に使用できるようにするための共通の保存フォーマット

4) 共通の表示方法のフォーマット

この Operational Statistics モデルの基では、データ収集ツールは与えられたフォーマットにしたがってデータを収集・保存し、あらかじめ定義された方法でそのデータを表すための表示ツールによって後でデータを引き出す（図 4.1 を参照)。

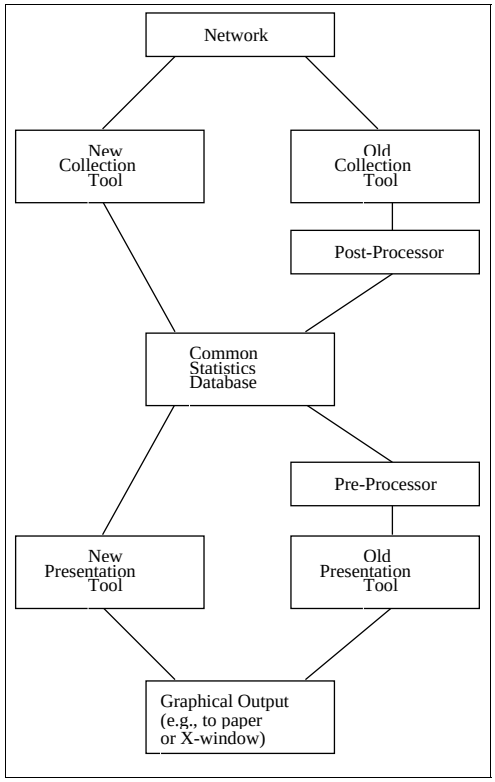

図 4.1: The Operational Statistics Model

このメモでは共通の operational statistics のためのモデルの概観 (ネットワーク operational statistics の収集や保存、表示法) を示す。

このモデルは尺度の最小セットや、これらの尺度がどのように収集・保存されるべきかを定義する。最終的には、NOC 間でネットワーク統計値を簡単に比較できるようにするための統計レポートの内容や構成について、その推薦されるものを示す。

このモデルの第一の目的は、複数の NOC 間で operational statistics を最も効果的に分担する方法を定義することである。

4.2　モデル

モデルの定義は 3 つに分けられる。

1. 収集される尺度の最小セットの定義

2. 収集された統計データを保存するためのフォーマットの定義

3. レポートを作成するための方法とフォーマットの定義

このモデルは、現在使用されている古い仕様のツールが、データをコンバートするフィルタを提供することによって、新しい仕様に適合できることを示している（図 4.1 を参照)。

このモデルの一つの基本的な考えとして、統計データを一箇所に保存して、複数の他の場所で取り出したり表示したりできるようにすることがある。

4.2.1 尺度とポーリング期間

ここでは、標準の SNMP[79] をベースとしたネットワーク管理ツールを用いることで簡単に収集することができる尺度の最小セットを定義する。したがってこれらの尺度は Internet 標準の MIB[98] の変数として利用できる。

Internet 標準の MIB が変更された場合、この尺度の最小セットの内容も考え直される。収集することを強く望むような尺度の中で、SNMP を用いて Internet 標準の MIB から現行では取り出す方法がない物もあるが、もしその様な尺度について良く検討されているなら、その様な尺度を収集するためのツールや方法が明白に定義されるべきである。

4.2.2 収集したデータを保存するためのフォーマット

余分な情報を最小にするような、データを保存するためのフォーマットを定義する。そのフォーマットは保存されている統計データのセットに関係する情報をすべて一つのヘッダーの構造に入れ、そしてそのヘッダー部に対応する統計データを収集した時間と場所についての情報を含める。

4.2.3 レポート

ある期間におけるオクテットとパケットの総量に関するレポートは、ネットワークのトラフィックの流れの全体を見渡すために重要である。どんなトラフィックのタイプが多いか考えるために、アプリケーションとプロトコルを区別する必要がある。最終的には、資源の利用についてのレポートが推薦されるだろう。

レポートではその目的によって測定期間が異なってもよい。

4.2.4 セキュリティに関する問題点

データの共有には、法律的、倫理的、政治的な関連がある。見ているデータのことを心配する人がいる。なぜなら、そのデータがそのネットワークの一つを悪く見せるかも知れないからである。

このため、共有されるデータが、完全な状態で、正確に一致していて、信頼できる物であることを保証する必要がある。有用なものにするためには、同種のデータが関係するすべてのサイトから収集されなければならず、そしてそれは同一の間隔で収集されなくてはならない。

4.3　尺度の分類

4.3.1　概観

この章では尺度の分類、および将来的なことまで含めて考えられた尺度に関するヒントについて述べる。

4.3.2　測定方法の区分を基準とした尺度の分類

ネットワークのトラフィックを評価するために用いられる尺度は、少なくとも 4 つの主要なカテゴリーに分類される。

- 利用に関する尺度
- 性能に関する尺度
- 可用性に関する尺度
- 安定性に関する尺度

- 利用に関する尺度

  ネットワークを通して転送されたトラフィックの総量について異なる側面を述べる物。

  – 入出力されたパケットとオクテットの総量
  – さまざまな尺度のピーク時の値
  – プロトコル毎、アプリケーション毎の尺度

- 性能に関する尺度

  ネットワークの遅延や混雑の状態のようなサービスの質について述べる物。

  – 異なるプロトコルレイヤーに関する RTT metric
  – バス (bus) ネットワーク上での衝突 (collision) の数
  – ICMP Source Quench メッセージの数
  – drop したパケットの数
  – その他

- 可用性に関する尺度

  異なるプロトコルレイヤーの、長期間の可用性を表す尺度とみなすことができるもの。

  – 利用可能であった割合で表された、回線の可用性。
  – 経路に関する可用性

- アプリケーションに関する可用性

- 安定性に関する尺度

  ネットワークのサービスのレベルを下げる様な、ネットワークの短期間の変動について述べる物。またこれらの尺度を用いることでトラフィックのパターンの変化についても知ることができる。

  - 急激な回線状態の変化に関する数
  - 急激な経路の変更に関する数 (経路の flapping としても知られている)
  - テーブル内のインターフェース毎の経路に関する数
  - 次の hop count の安定性
  - 短い間隔での ICMP のふるまい

4.3.3 尺度の可用性を基準とした分類

尺度を取り出すことを可能にするためには、対応する変数が、統計値が収集されている管理ドメインの一部となっているすべてのネットワークからアクセス可能でなくてはならない。

Internet 標準の MIB で変数として定義されているので容易に取り出すことができる尺度もあるし、あるベンダーのプライベートな enterprise MIB の subtree の一部として取り出すことができる尺度もある。また、ある尺度は SNMP の概念に含めることができないために取り出すことが不可能であると考えられる。あるいは、これらの尺度の実際の測定方法は広範囲に渡るポーリングを必要とするので取り出すことが不可能であると考えられる。

以降で示すような分類された尺度が、ネットワークの振舞いを評価する上で重要な尺度であると判断することができた。

- Internet 標準の MIB の中に既に含まれている、インターフェースに関する変数

  Internet 標準の MIB に既に含まれている変数で、簡単に取り出せるもの

  - ifInUcastPkts (入ってきた単一アドレス宛のパケット数)
  - ifOutUcastPkts (出て行った単一アドレス宛のパケット数)
  - ifInNUcastPkts (入ってきた、非単一アドレス宛 (ブロードキャスト等) のパケット数)
  - ifOutNUcastPkts (出て行った、非単一アドレス宛 (ブロードキャスト等) のパケット数)
  - ifInOctets (入ってきたオクテット数)
  - ifOutOctets (出て行ったオクテット数)

  - ifOperStatus (回線の状態)

- プライベートな Enterprise MIB の中に含まれている、インターフェースに関する変数

  プライベートな enterprise MIB に入れられている変数で、時には取り出し可能なもの

  - 入ってきた時点で捨てられてしまったパケットの数
  - 送り出そうとした時点で捨てられてしまったパケットの数
  - 入ってきた輻輳イベントの数
  - 出て行った輻輳イベントの数
  - 集計エラーの数
  - インターフェースのリセットの数

- 高頻度のポーリングを必要としている、インターフェースに関する変数

  高い頻度でのポーリングを必要とする変数で、ネットワークに負荷を与えるために難しいもの

  - インターフェースの待ち行列の長さ
  - 統計情報を収集できなかった期間
  - インターフェースが利用不可能であった時間
  - 経路の変更の頻度
  - そのインターフェースにおける次のホップ数

- どんな MIB の中にも含まれていない、インターフェースに関する変数

  MIB の中に含まれていない変数で、MIB の中に含むことは可能であるが、SNMP を用いて取り出すことが不可能なもの

  - リンクレイヤーに入ってきたパケット数
  - リンクレイヤーから出て行ったパケット数
  - リンクレイヤーに入ってきたオクテット数
  - リンクレイヤーから出て行ったオクテット数
  - パケットの到着時間間隔
  - パケット長の分布

- ノード毎の変数

  これまでに分類されていないもの

- プロトコル別の、入ってきたパケット数
- プロトコル別の、出て行ったパケット数
- プロトコル別の、入ってきたオクテット数
- プロトコル別の、出て行ったオクテット数
- 入ってきた時点で捨てられたパケット数
- 送り出そうとした時点で捨てられたパケット数
- パケット長分布
- システムの uptime
- ポーリングの間隔
- リブートした回数

- SNMP を用いて取り出すことができない尺度

  SNMP を用いて取り出すことができない尺度

  - 異なるプロトコルレイヤーでの遅延 (RTT)
  - アプリケーションレイヤーの可用性 (availability)
  - ピーク時の尺度の振舞い

4.3.4 推薦される尺度

このモデルが一般的な総意に至ることを容易にするために、尺度の最小セットを定義する必要がある。それは、必要不可欠な物、今日のネットワークの大部分で取り出すことが可能な物の両方である。一般的に取り出すことが可能であることを示すものとして、Internet 標準の MIB の中にあるものが必須であると考えられる。

- 選ばれた尺度

  以下に続く尺度は Internet 標準の MIB の中で合理的で望ましいものとして選ばれたものである。

  各インターフェース毎に

  - ifInOctets
  - ifOutOctets
  - ifInUcastPkts
  - ifOutUcastPkts
  - ifInNUcastPkts

- ifOutNUcastPkts
- ifInDiscards
- ifOutDiscards
- ifOperStatus

各ノード毎に

- ipForwDatagrams
- ipInDiscards
- sysUpTime

上にあげたすべての尺度は Internet 標準の MIB の中で利用できる。しかし RTT metric の様に、多分どんな MIB の中にも決して含まれていないような他の尺度でも、推薦できるものがある。このような尺度のためには、SNMP とは違ったデータ収集用のツールについてはっきりと定義する必要がある。

4.4　ポーリング頻度

ポーリングは、ネットワークの動向とキャパシティについての計画を行なうための基準として役に立つ統計量を集め保存するために行なう。すべての値について調査し蓄えることが推薦されるが必須ではない。

4.4.1　高頻度のポーリングを必要としている変数

ピーク時の振舞いを見つけ出すために推薦できることは、1 分を最大とした期間でトラフィックデータの収集を行なうことである。この頻度で集められる尺度は、
各インターフェース毎に、

- ifInOctets
- ifOutOctets
- ifInUcastPkts
- ifOutUcastPkts

もしこの高いポーリングの頻度でデータの収集を行なうことが可能でないなら、60 秒の偶数倍の時間を用いることが推薦される。

4.4.2 高頻度のポーリングを必要としていない変数

各インターフェース毎、

- ifInNUcastPkts
- ifOutNUcastPkts
- ifInDiscards
- ifOutDiscards
- ifOperStatus

そして、各ノード毎、

- ipForwDatagrams (IP forwards)
- ipInDiscards (IP in discards)
- sysUpTime (system uptime)

これらの変数はより低いポーリングの頻度で集められても良い。特別なポーリングの頻度については言及しないが、60 秒の偶数倍を選ぶことが勧められる。

4.5 生の統計データに対する前処理

4.5.1 データの最適化と圧縮

冗長なデータの保存を避けるために、(保存する前に) 生のデータに処理を加える必要がある。例えば、リンクが落ちている時にデータを保存する必要はないし、ある期間全くトラフィックのないリンクやノードからデータを集めて保存し続けることもない。

処理を行なうことの他の側面は、調査が行なわれている未加工のインターフェースからデータを取り出し、それを役に立つデータに変換することである。

4.5.2 データの集計

ここでは集計をエントリーが記録され一つのポーリング期間が終ることを意味するものとする。新しく集められたエントリーは、いくつかの集計期間を終えて前もって記録されているエントリーの、その最初の物と最後の物を用いて作り出され、新しいエントリーが計算される。

集められたデータは与えられたファイルに前処理を施されて入れられる。

集計期間に関する提案:

- 24 時間の期間では、総計 15 分、
- 1 ヶ月の期間では、総計 1 時間、
- 1 年の期間では、総計 1 日

集計は、その前の集計期間のデータを基にした集計期間の値の、新しい平均値と最大値の計算結果である。それぞれの集計期間について、その最大値と平均値が計算されて保存されている。

4.6　統計データの保存

この章では統計データを保存するためのフォーマットについて述べる。最終的な目的は、統計データの収集や保存、解析のためのツールを簡単にすることである。フォーマットは、冗長な情報を最小にするという目的で定義されている。

フォーマットは、異なる 3 つの部分からなり、ラベルセクション、デバイスセクション、データセクションとなる。ラベルセクションは続くデータセクションのデータの始まりと終りの時間を示し、データセクションは実際にデータが格納されたファイルと同じ物である。デバイスセクションは対応しているデータセクションに何が登録されているか特定する。

ラベルセクションとデバイスセクション、データセクションは一度以上現れても良い。それぞれのセクションの形式は一つの BEGIN-END のペアによって範囲が決められる。ラベルとデバイスセクションはともにデータファイルの中に直接格納されるか、ラベルセクションの data-file のエントリによって指定される別のファイルに格納される。

一つのデータセクションが一つのラベルセクションと一つのデバイスセクションに正確に対応していなくてはならない。

4.6.1　格納時のフォーマット

```
stat-data ::=
<label-section><FS><device-section><FS><data-section><FS>
[<device-section><FS><data-section><FS>]

FS ::= "," | <LF> | <LF> # any text here <LF>
```

ファイルはラベルセクションから始まり、続いてデバイスセクション、データセクションと続く。もし何かの理由で記録されたデータの保存が中断された場合には、再びデータの保存が始められた時に新しいラベルが挿入される。記録されているデバイスが変えられた場合、新しいラベルと新しいデバイスを挿入することでこのことが表される。

実際のデータの物理的な保存はローカルに決められることで、いろいろ変えることができる。一つのインターフェース毎、あるいは同じデータファイルの中に記録された複

数のインターフェース毎に一つのデータファイルがあってもよい。ラベルセクションとデバイスセクションは、データファイルの中に入れても別ファイルに入れて保存しても良い。

- ラベルセクション

```
label-section ::=  "BEGIN_LABEL"  <FS>
                   <start_time>   <FS>
                   <stop_time>    <FS>
                   <data_file>    <FS>
                   "END_LABEL"



start-time  ::= <time-string>
end-time    ::= <time-string>
file-name   ::= <ascii-string>
time-string ::= <year><month><day><hour><minute><second>
year        ::= <digit><digit><digit><digit>
month       ::= 01 | ... | 12
hour        ::= 00 | ... | 23
minute      ::= 00 | ... | 59
second      ::= 00 | ... | 59
digit       ::=  0 | ... | 9

ascii-string ::=  MIB II で定義された <ascii-string> と同じ
```

時間は記録されているデータのセットに関連した始まりと終りの時間である。時間は UTC 形式。

- デバイスセクション

```
device-section ::= "BEGIN_DEVICE" <FS>
                   <device-field> <FS>
                   "END_DEVICE"

device-field   ::= <networkname><FS><routername><FS><linkname><FS>
                   <bw-value><FS><bw-sort><FS><proto-type><FS>
                   <proto-addr><FS><time-zone><FS><tag-table>
                   [<tag-table>]
```

```
networkname    ::= <ascii-string>
routername     ::= <ドメイン名に完全に適合したもの>
linkname       ::= <ascii-string>
bw-value       ::= <実際の bandwidth の値>
bw-sort        ::= "bps" | "Kbps" | "Mbps" | "Gbps" | "Tbps"
proto-type     ::= "IP" | "DECNET" | "X.25" | "CLNS"
proto-addr     ::= <proto-type に依存したネットワークアドレス>
timezone       ::= <"+" | "-"><00 | ... | 12><00 | 30>
tag-table      ::= <tag><FS><tag-class><FS><variable-field>
                   [<FS><variable-field>]
tag-class      ::= "total" | "peak"
variable-field ::= <variable-name> <FS> <initial-polling-period><FS>
                   <aggregation-period>
tag            ::= <ascii-string>
variable-name  ::= <ascii-string>

initial-polling-period ::= <digit>[<digit>]
aggregation-period     ::= <digit>[<digit>]
```

networkname は人間が読みやすい文字列の形式で、記録されたデータがどのネットワークに属しているかを表す。

routername はルータがどこに設置されているかというネットワークアーキテクチャにとって適切な、完全に適合した名前。

linkname は人間が読みやすい文字列の形式で、記録されたデータがどのリンクから集められたかを表す。

bandwidth は数値で、使用されている種類が続く。有効な種類は bps, Kbps, Mbps, Tbps 。

prototype フィールドは記録されているインターフェースがどんなネットワークアーキテクチャに繋がれているかを示す。有効なものは IP, DECNET, X.25, CLNP 。

network address は記録されているインターフェースの固有の数値形式のアドレスである。このアドレスの実際の形式は proto-type フィールドに示されたプロトコルに依存する。

time-zone は、データセクションの中で記録されたインターフェースの、ローカルな時間で表されているタイムスタンプに加えられるべき、時間差を示している。

tag-table はすべての集められた変数のリストである。変数名は Internet MIB の名前に完全に適合するものである。テーブルは複数の tag を含んで良い。それぞれの

tag は一回のポーリング及び集計期間にのみ関係づけられなくてはならない。もし、変数が異なる期間に集められたなら、テーブルの中で一つの別のタグがそれぞれの期間毎に用いられなくてはならない。

変数は、記録されたデータの同じセット内で異なるポーリング期間で調査されたものを含んでも構わないので、それぞれの記録されたデータについてはっきりとポーリング期間を関連づける必要がある。処理が終った後で、実際の期間は initial polling period と比較して変化していても良く、これは aggregation period のフィールドに記される。initial polling period と aggregation period は、秒で示される。

集計はその前に調査されたデータの最大値を計算することでもあるので、集計の過程ではこれらの最大値を含めるために tag table を拡張する必要がある。収集された変数のフィールドは、前の期間からピーク時の値も含めることで拡張される。他の可能性はピーク時の値用に新しい tag を作ることである。未加工のデータと集められた物の総量、集められた物のピーク時の値が識別できるように、そのようなエントリーが一意に決まるような名前付けがインプリメントされる必要がある。

- データセクション

```
data-section    ::= "BEGIN_DATA"<FS>
                    <data-field><LF>
                    "END_DATA"

data-field      ::= <timestamp><FS><tag><FS>
                    <poll-delta><FS><delta-val>
                    [<FS><delta-val>]

poll-delta  ::= <digit> [<digit>]
tag         ::= <ascii-string>
delta-value ::= <digit> [<digit>]
timestamp   ::= <year><month><day><hour><minute><second>
year        ::= <digit><digit><digit><digit>
month       ::= 01 | ... | 12
hour        ::= 00 | ... | 23
minute      ::= 00 | ... | 59
second      ::= 00 | ... | 59
digit       ::=  0 | ... | 9
```

データフィールドは tag フィールドに対応するデータが含まれている。それぞれのデータフィールドはタイムスタンプから始まり、集められたデータの tag の定義、その前のポーリングからの差を秒数で示している値が続く。変数の値は、カウンター

用には一つ前との差の値として格納され、OperStatus の様なカウンターでないものには絶対値が格納される。タイムスタンプは UTC で表され、デバイスセクションにある time-zone フィールドは記録されているデバイス用にローカルな時間を計算するために使用される。

4.6.2　保存に必要な容量の推定

ヘッダーセクションについてはこの例の中では含めていない。ポーリングが最大、推薦される 12 の変数のすべてについて用いられると仮定し、また、それぞれの変数を ascii 文字で表した大きさが 8 バイトであると仮定すると、以下のような計算が一年を基準とした統計データの収集と保存について行なえる。

データは以下の分類表にしたがって保存されると仮定する。

```
1 minute non-aggregated          saved 1 day.
15 minute aggregation period     saved 1 week.
1 hour aggregation period        saved 1 month.
1 day aggregation period         saved 1 year.
```

ここから次のようになるであろう：
それぞれの集計期間毎の一つのエントリーの大きさ：

	Aggregation periods			
	1 min	15 min	1 hour	1 day
Timestamp	14	14	14	14
Tag	5	5	5	5
Poll-Delta	2	3	4	5
Total values	96	96	96	96
Peak values	0	96	192	288
Field separators	14	28	42	56
Total entry size	131	242	353	464

毎日 60*24 = 1440 エントリありトータルサイズで 1440*131 = 187 Kbytes.
毎週 4*24*7 = 672 エントリが保存され、トータルサイズ 672*242 = 163 Kbytes.
毎月 24*30 = 720 エントリが保存され、トータルサイズ 720*353 = 254 Kbytes.
毎年 365 エントリが保存され、トータルサイズ 365*464 = 169 Kbytes.
全体の総数として、一年を通して 773 Kbytes 保存されると推定した。

4.7 レポートのフォーマット

この章ではいくつかの レポートフォーマットを提案し、これらレポートの中で使われている尺度を定義する。

4.7.1 レポートの形式と内容

ネットワークにおける長期間にわたる傾向、動向を示すには何カ月、何年もの間レポートを集める必要がある。ネットワークの振舞いにおいてやや長い期間の変化の動向を与える短期間の週間的なレポートがある。それはやや長い期間の技術的アプローチの入力として役立つ。結局、ネットワークの日々のオペレーションについての断片的なあらましを示す、日々のレポートが必要となる。

これらのレポートは次のような情報を必要とする。

- 外部インターフェースの総トラフィックによりもたらされた負荷
- 利用者によって分類された負荷
- プロトコル、アプリケーションによって分類された負荷
- 回線やルータの資源利用

4.7.2 レポートの内容

- リンクあたりの負荷

  尺度の区分 :

  - 外部インタフェース当たりの入力オクテット
  - 同、出力オクテット
  - 外部インタフェース当たりの入力パケット
  - 同、出力パケット

  目的は外部インターフェースで各々つながったネットワークのトラフィックの全体的な傾向を視覚化することにある。これは上記 4 つの尺度区分の各合計を棒グラフにして表すことが出来る。1 時間、1 日、1 カ月、1 年など時間的区分をいろいろ選択して作れば良い。

- 利用者あたりの負荷

  尺度の区分:

  - 利用者当たりの入力オクテット

  - 同、出力オクテット
  - 利用者当たりの入力パケット
  - 同、出力パケット

  ここでは利用者によって影響が出た負荷を（降順に）選別しておく。関数 F(n) を考える。ここで F(n) は、上位 n 人の利用者により与えられた (offerd) トラフィックの合計のパーセンテージである。また関数 f(n) は n 番目にランクされた利用者によって与えられたトラフィックのパーセンテージである。

  どんな動きをすれば、それが利用者によって行なわれた事を意味するのかという定義は、統計が集められているサイトでローカルに行なわなければならない。

  累積すればトラフィックが各ユーザー間においてどのように分割されたかを見渡すのに有益である。そうしておけば、「何番目のユーザーにより発生したトラフィックがどの位の割合になるか」などを素早く抽出することが可能となる。

  棒グラフで各値の平均やピーク時の変化を表す方法は、ある期間内における平均値とその同じ期間内でのピークの値を計算すればよい。平均とピークの値は同じ棒の中に表す。

- 資源の利用に関する報告

  4.7.2.0.1　最大ピーク時の振舞いとして表された利用率　回線の利用状態はネットワーク負荷の情報を集めるのに用いられる。ポーリングの間隔は人間の活動の変化を十分に表せるよう注意するために、出来る限り小さくしなければならない。なぜならこれがネットワークの変化における負荷の動きの動向を表すからだ。一方、特に大きな影響を与えそうな過度の遅れに関しては間隔を小さくする必要はない。なぜなら過度の遅れは注目すべき影響を引き起こさないからだ。その理由は、人が一つの活動に従事し続けられる時間はだいたい 30 分が限度といえるからだ。そしてかなり長い遅れは彼らの作業効率に悪影響を与える。30 分ごとの変化を記録するのは時間当たり二回すなわち 15 分おきにサンプルが必要ということになる。10 分間隔で集計することを勧めるのは、利用状態の変化を捉えるのにはそれで十分であるからだ。

  ピーク時の様子を見る利用状態レポートにおいて可能なフォーマットは平均値とピーク値の合計を同じ図に表すようにする。例えばオーディオ機器と比べて見れば良い。もしグラフがある時点での一日の合計値を示しているとすれば、そのピークはそれらの日の中で一番混んでいる時を表している。また時間単位での合計値ならば、そのピークは各時間における一番混んでいる 10 分間ということになる。

  ある程度の時間単位でその平均や最大値を集計することで 回線の利用状態や一時的に負荷が非常に高くなることによるボトルネックになる部分の解析が可能になるだろう。

4.7.2.0.2 ピークの頻度分布により表された利用率 回線の利用状態を視覚化するその他の方法には、10 分間のサンプルを作り、サンプルと実際の負荷における頻度との比較を表すヒストグラムを作ることだ。

4.8 将来の発展についての考察

このメモは統計値収集作業の共通基盤を正式化するための初めての試みである。この作業の主要な指針の一つは、ベンダーと NOC が各自のオペレーションツールでこのモデルを簡単に統合できるように、モデルを単純化しておくことにある。

また、このモデルの活用範囲や有用性をより大きく発展させたいくつかの案もある。

4.8.1 クライアント／サーバーモデルを基にした統計情報交換システム

この方法がより発達すれば、 クライアント/サーバシステムにおける、統計収集作業をするためのインタネットアクセスを供給するための定義が可能になる。そのような構造を心に描くと 、各 NOC はローカルに集めた様々な形の情報を提供するサーバをクライアントのためにインストールしなければならない。

質問的な語調を使えば、クライアントは供給されたネットワークオブジェクトやインタフェース, 尺度や時間の区切りを定義出来なければならないのかということになる。

TCP をベースとしたプロトコルを使うとすれば、サーバはリクエストデータを送らなければならないだろう。一度クライアントがこれらのデータを受けとると数多くのツールを使うことで処理したり表示したりすることが出来るようになる。

統計収集サーバから X Window ツールのディレクトリに与えられたデータから、あらかじめ定義されたグラフを表示する、それらのタイプのグラフを補っている X Window ベースのツールに一つの可能性がある。他の捕捉方法としては、グラフをプリント出来るように、Post Script の出力が出来るようにすることもある。どのケースでも、以前の処理でローカルに収集されたデータを、保存できる可能性がなければならない。

4.8.2 Internet 標準の MIB に含まれていない変数の追加について

尺度の分類に注目してみると、もし Internet 標準の MIB の中で利用できれば確実に勧められる尺度がある。勧められる尺度のセットの一部を成す尺度を簡単にするため、統計収集作業を実行する範囲において必要と判断された変数を含んだ サブツリーを MIB の中で明示する必要があるだろう。

4.8.3 詳細な資源利用に関する統計

尺度や表示方法に関する今までの紹介では興味ある分野のほんの一部でしかないが、それらはトラフィックの流れを細かい視点から見た統計値を表示するためのものである。そ

れらの視点には アプリケーションやプロトコルを基にした割合の統計値も含まれる。今日それらの尺度は MIB の標準ではない。NSF や NNStat のようなツールはこれらの情報を集めるのに使われている。

第 5 章

まとめ

本報告では、92 年度に行ったデータ収集および解析結果にもとづき、WIDE インターネット上のトラフィックについての考察を行った。とくに今回は、国際線上を流れている ICMP の解析も行い、これらの解析から得られる情報をネットワーク管理にも利用可能である事を示した。最後に、今年度の活動の基盤の一つでもある、現在 RFC により提案されている Operational Statistics Model についての紹介を行った。

本年度は、昨年度までの研究結果を引き継いで、WIDE バックボーン全域にデータ収集範囲を広げ、広域でのデータ収集系の運用を開始する。それにともない、収集したデータの効率の良い保存手法ならびに解析手法に関する研究に焦点をあてる。そして、今回紹介した IETF の Operational Statistics Model にもとづいて、これらの統計情報の効率の良い保存方法や解析結果の表現方法についての考察を進め、我々独自の案を IETF に提案する等をとおして、データ保存と解析に関する国際的な協調のもとでの研究を進めていく予定である。また、現在行っている比較的長い期間でのデータ収集・解析に加えて、ICMP 等に着目したリアルタイムのトラフィックデータ解析の、ネットワーク管理への応用について、SNMP による管理手法と比較しながら研究を進めていく。