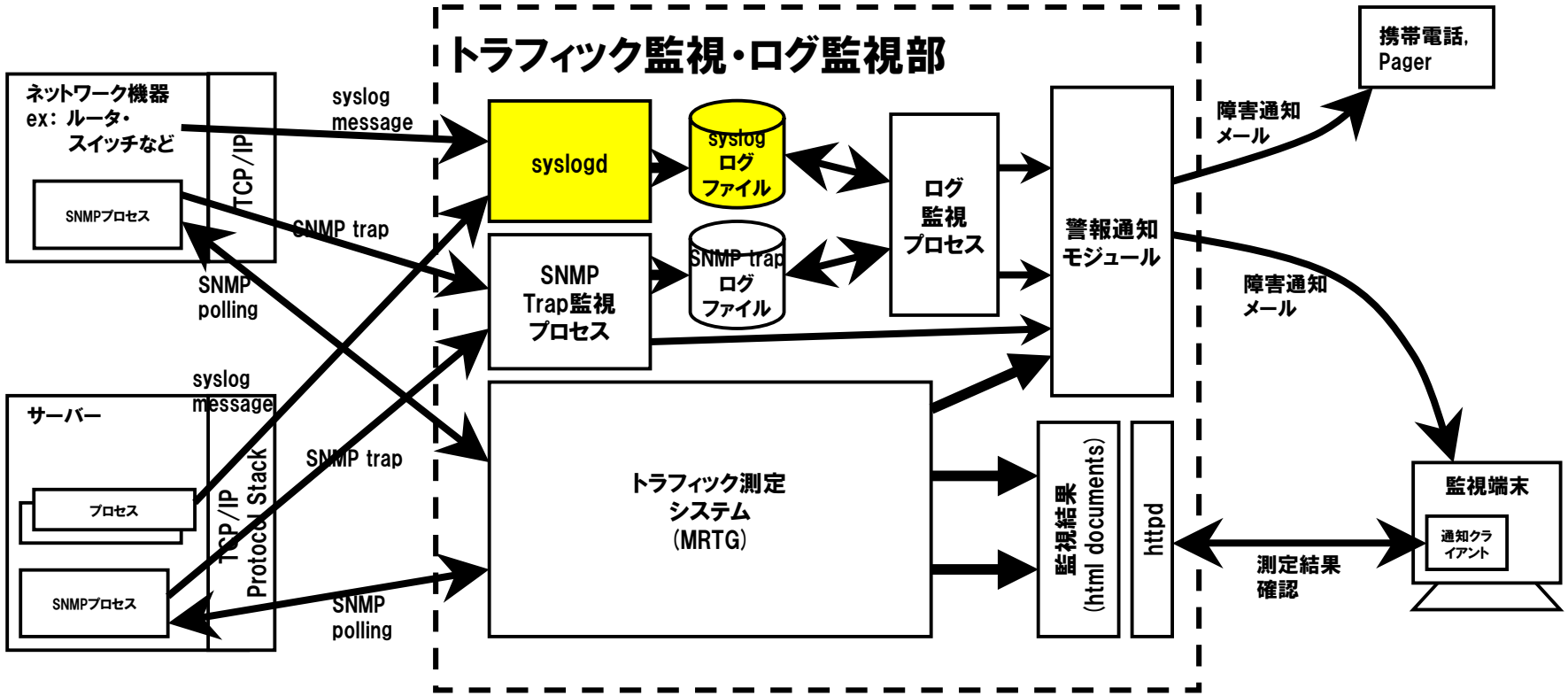
index

監視システムのモデル －トラフィック・ログ監視サーバ



自律メッセージ管理

- 自律メッセージとは?
  - ルータ・BRAS・スイッチから自律的に出力されるイベントメッセージ
  - 例: Interfaceのup/down、プロトコルの状態変化、パッケージ不良、ソフトリブート、・・・
- 各メッセージには重要度 (=プライオリティ) が設定されている
  - 重要度＝最大:　　システム再開
  - 重要度＝大:　　モジュール故障
  - 重要度＝小:　　インタフェースダウン
  - 重要度＝最小:　　ユーザログイン
- 出力先は入り口により選択される
  - コンソールへの自律メッセージ出力
  - 内部メモリ上のログバッファへの自律メッセージ出力
  - telnetターミナルへの自律メッセージ出力
  - logサーバへの自律メッセージ出力　→　syslog機構
  - 自律メッセージの管理はUNIXにて開発されたsyslogの仕様に基づき行われる。

syslogとは：1

- 機器から出力される自律メッセージ＝ログの集中管理を行う機構
- ハードディスクのような固定的な記憶媒体を持たないネットワーク機器はリブートしてしまうと、障害にいたるまでの経過が把握できない
- syslog機能により、ログサーバに対してログメッセージをネットワーク経由で記録する

- メッセージファシリティーとメッセージプライオリティ
  - ファシリティー：メッセージの送り先チャネルの指定
  - プライオリティー：メッセージの重要度の指定

syslogとは：2
メッセージファシリティとメッセージプライオリティ

- ファシリティー：メッセージの送り先チャネルの指定
  - 例：kern -> kernel message, mail -> mail system message,
    auth -> authorization system message,
    security -> security subsystem message
  - ユーザが独自に使用できるのは local0からlocal7までの8ファシリティとなる
- プライオリティー：メッセージの重要度の指定
  - 8種類のプライオリティ

0	EMERG	PANICメッセージ。全ユーザに通知される
1	ALERT	システムDBが壊れているような直ちに対処が必要な重要障害警告
2	CRIT	ハードウェアのデバイスエラーのような危急状態の警告
3	ERR	その他のエラーメッセージ
4	WARN	警告メッセージ
5	NOTICE	エラーではないが、注意が必要なメッセージ
6	INFO	参考情報メッセージ
7	DEBUG	デバッグメッセージ

- メッセージの指定例
  - kern.debug ： カーネルサブシステムのデバッグメッセージのみ
  - local2.crit ： ローカルファシリティ 2番のクリティカルメッセージのみ
  - mail.* ： メールサブシステムの全メッセージ

syslogの設定（サーバ側）：/etc/syslog.conf

- レコード形式：
  <facility>.<priority> { ; <facility>.<priority> } <TAB> <output>
  - <facility>：メッセージファシリティ。ワイルドカード * のみの使用可能
  - <priority>：メッセージプライオリティ。ワイルドカード * のみの使用可能
  - <TAB>：メッセージ指定と出力先を区切る記号。タブ文字のみ使用可能となっており、よくスペースと混同される。要注意
  - <output>：出力先。以下の指定例が可能
    - ファイル指定：/var/log/router.log
    - 特定ユーザ：root, yahagi
    - 全ログインユーザ：*
    - コマンドリダイレクト：| /usr/local/bin/filter.pl
    - 他ホストへの転送：@log0.branch.aa.jp

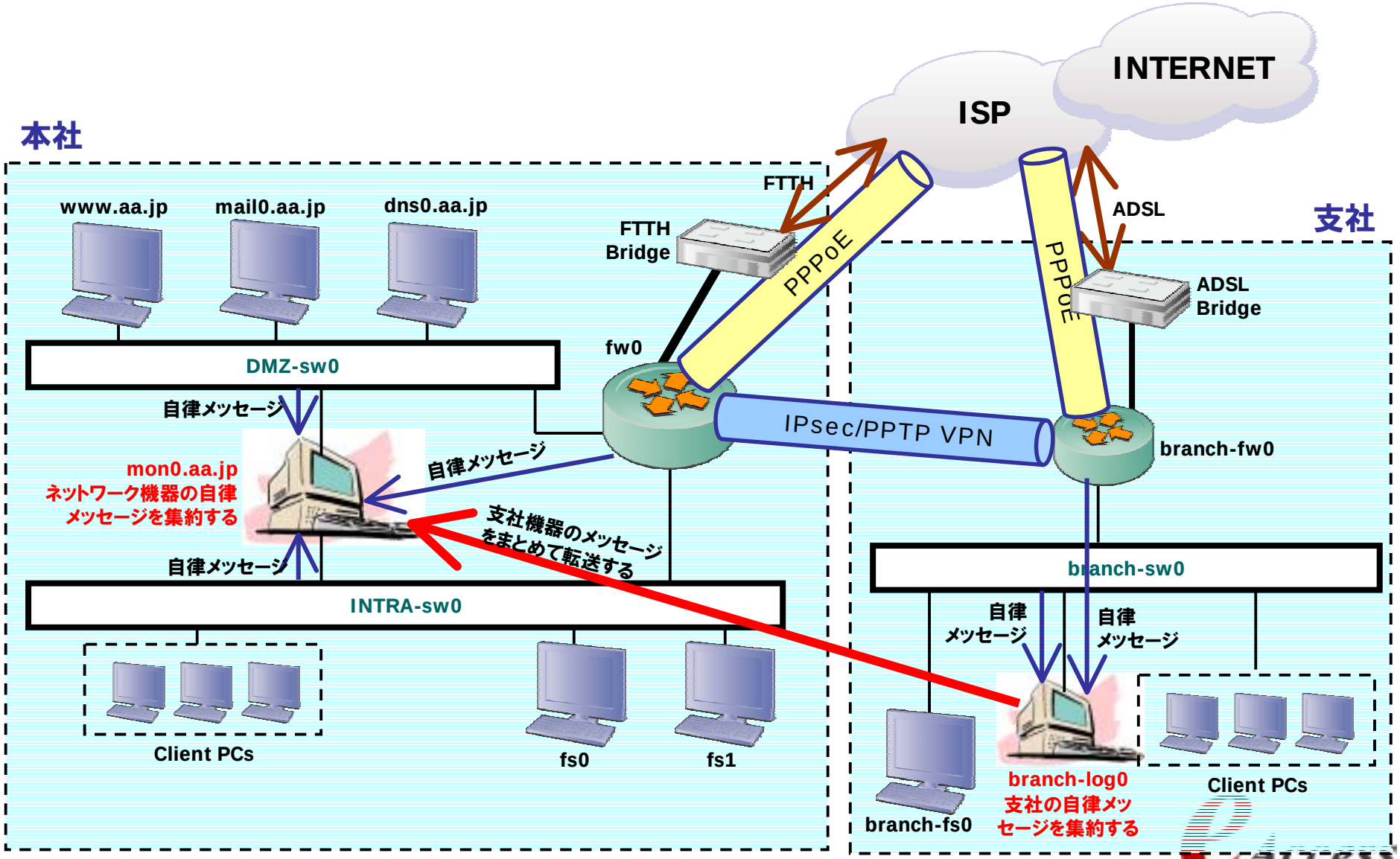
```
$ cat /etc/syslogd.conf.example
# syslog.conf.example
*.notice;kern.debug;mail.crit;news.err;*.crit     /var/log/messages
mail.info                                         /var/log/maillog
cron.*                                            /var/cron/log
*.emerg                                           *
local0.debug;local1.debug;local5.debug            @log0.branch.aa.jp
local0.*;local2.*;local5.*                        | /usr/local/bin/filter.pl
kern.panic                                        root,yahagi
$
```

実装検討2 – syslog5 ネットワーク機器側での設定

- サーバ側のファシリティ・プライオリティ設定は完全照合か”*”ワイルドカードでの指定
- ネットワーク装置側ではファシリティは個別指定であるが、プライオリティは指定プライオリティ以上のメッセージを転送となる

- CISCO IOSでの設定例
  - 172.16.0.4に ”local1” ファシリティで ”INFO” 以上のメッセージを送信
  - router (config) # logging trap info
    router (config) # logging facility local1
    router (config) # logging 172.16.0.4

実装検討2 – syslog3：メッセージの集約



実装検討2 – syslog4

- syslogにて情報を取得する対象を以下のように分類
  - ファイヤーウォール: local0
    - 対象：fw0, branch-fw0
  - DMZスイッチ: local1
    - 対象：dmz-sw0
  - イントラスイッチ：local2
    - 対象：intra-sw0, branch-sw0
- 各装置からはINFO以上のメッセージのみを送信する
- サーバは全てのログをうけとる設定
- 支社のログはlog0.branch.aa.jpでいったん受けて、mon0.aa.jpに転送

場所	機器	送付先	ファシリティ	プライオリティ	出力先
本社	fw0	mon0.aa.jp	local0	info	/var/log/fw.log
	dmz-sw0	mon0.aa.jp	local1	info	/var/log/dmzsw.log
	intra-sw0	mon0.aa.jp	local2	info	/var/log/intrasw.log
支社	branch-fw0	log0.branch.aa.jp	local0	info	/var/log/fw.log @mon0.aa.jp
	branch-sw0	log0.branch.aa.jp	local2	info	/var/log/intrasw.log @mon0.aa.jp

実装検討2 – syslog4 /etc/syslog.conf

- mon0.aa.jp, log0.branch.aa.jpの/etc/syslog.confに以下の設定を追加投入する
  - mon0.aa.jp:/etc/syslog.conf追加設定

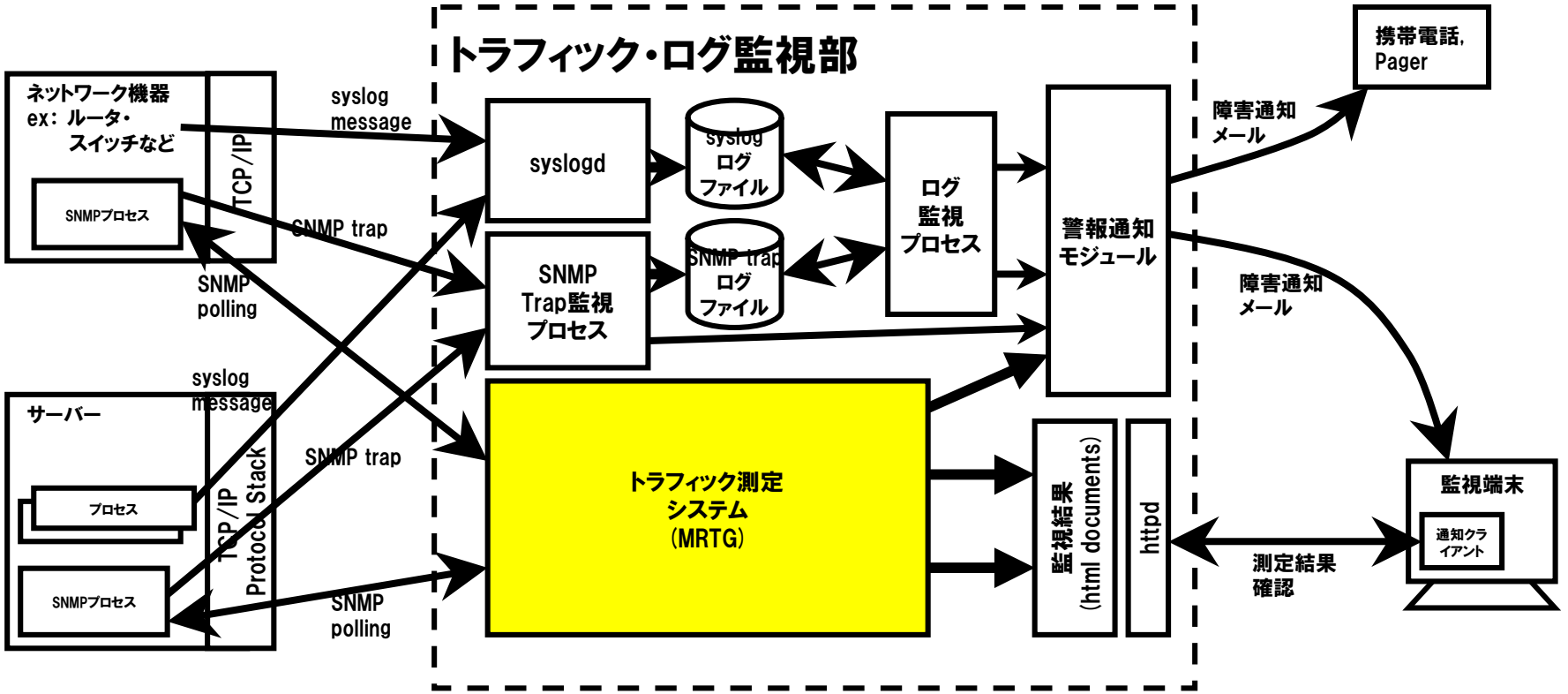
```
local0.*                        /var/log/fw.log
local1.*                        /var/log/dmzsw.log
local2.*                        /var/log/intrasw.log
```

  - log0.branch.aa.jp:/etc/syslog.conf追加設定

```
local0.*                        /var/log/fw.log
local2.*                        /var/log/intrasw.log
local0.*;local2.*               @mon0.aa.jp
```

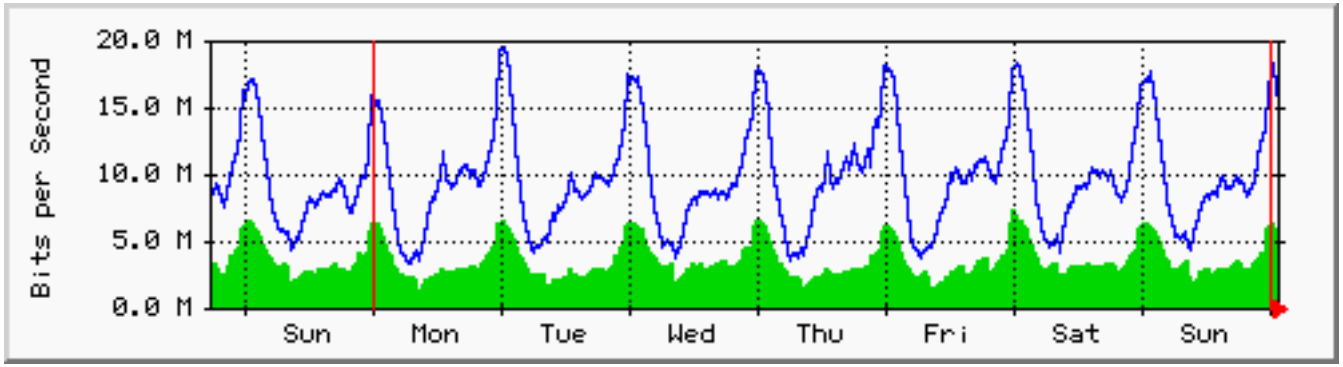
  - ＃注意 ファシリティと出力先の間はスペースではなくタブ（TAB）で区切ることを注意。
    ＃また、出力先のファイルは自動的に作成されないので事前に作成が必要

実装検討2 – トラフィック監視



実装検討2 – トラフィック監視
MRTGとは

- http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
- http://www.mrtg.jp/doc/
  （日本語翻訳サイト）
- MRTG : Multi Router Traffic Grapher
- ２系列のデータを基に集計を行い、短期・中期・長期トレンドグラフを生成するツール



実装検討2 – トラフィック監視 MRTGの特徴1

- ほとんどのUnixプラットフォームとWindowsNT/2k/XP上で稼動
- 独自にSNMPを実装。外部のSNMP Packageは不要
- 定期的にログをサマリーするデータ管理を行っており、ログファイルのサイズが大きくならない
- 半自動のコンフィグ作成ツールが付属
- 日・週・月・年ごとにデータを集計したWEBページを結果として生成する
- コンフィグからindexを簡単に生成するツールが付属
- デフォルトはcronによる定期起動だが、Daemon化することも可能
- Unixプラットフォームでは並列照会による高速化をサポート

実装検討2 – トラフィック監視 MRTGの特徴2

- 多様性に富んだ測定対象の指定方法
  - 以下のInterface属性をキーに、当該インタフェースを特定する
    - MAC address指定
    - Description指定
    - Interface Name指定
    - Interface Type指定
- RRDToolsとの統合： LogFormat: rrdtool
  - logの管理をRRDToolを使用することにより、劇的な高速化を実現する
  - データは本オプション指定により自動的にRRD形式にデータ移行される
  - グラフの作成は測定時しない。付属の14all.cgiによりon the flyで（要求のたびに）作成をする
  - 10倍以上高速になることも
- 最新版（2.10）でのトピック
  - IPv6対応
  - ConversionCode: Perlの外部サブルーチン関数を埋め込み可能

実装検討2 – トラフィック監視 MRTG – cfgmaker - 1

- mrtg付属の簡易設定ツール
  - `cfgmaker { <option> } <community>@<target>`
    `<community> : snmp community string`
    `<target>    : target address or hostname`
  - 例: `$ cfgmaker himitsu@ix-gw.aa.jp > ix-gw.cfg`

- communityとtargetを指定するだけで機器に存在するインタフェースをサーチし、ifInOctets/ ifOutOctetsを測定する設定の大部分を作成する
  - syscontact/locationなどの情報からコメントも自動作成
  - 保守停止しているインタフェースについてはコメントとして作成
  - 追加設定は WorkDir: だけでほぼ動く
  - pps/packet discardsなどの他の項目測定については、cfgmakerの結果を元に作成していくのが、普通のやり方
    - 各測定項目のスケルトンパターンを持つのが一番有効ではあるが...
- 以下のキー指定可能
  - `--ifref=nr    ... interface references by Interface Number(default)`
    `--ifref=ip    ... by Ip Address`
    `--ifref=eth   ... by Ethernet Number`
    `--ifref=descr ... by Interface Description`
    `--ifref=name  ... by Interface Name`
    `--ifref=type  ... by Interface Type`

実装検討2 – トラフィック監視
MRTG – cfgmaker の出力結果例

```
$cfgmaker  --ifref=name  himitsu@192.168.0.1

～初期設定処理表示:省略～

# Created by
# /usr/local/bin/cfgmaker --ifref=name himitsu@192.168.0.1

### Global Config Options

#  for UNIX
# WorkDir: /home/http/mrtg

#  or for NT
# WorkDir: c:¥mrtgdata

### Global Defaults

#  to get bits instead of bytes and graphs growing to the right
# Options[_]: growright, bits

######################################################################
# System: router1
# Description: Cisco Internetwork Operating System Software
# Contact:
# Location:
######################################################################

### Interface 1 >> Descr: 'ATM2/0' | Name: 'AT2/0' | Ip: '' | Eth: ''
###
### The following interface is commented out because:
### * it is operationally DOWN
#
# Target[192.168.0.1_AT2_0]: #AT2/0:himitsu@192.168.0.1:
# SetEnv[192.168.0.1_AT2_0]: MRTG_INT_IP="" MRTG_INT_DESCR="ATM2/0"
# MaxBytes[192.168.0.1_AT2_0]: 18720000
# Title[192.168.0.1_AT2_0]: Traffic Analysis for AT2/0 -- router1
# PageTop[192.168.0.1_AT2_0]: <H1>Traffic Analysis for AT2/0 --
router1</H1>
#  <TABLE>
#    <TR><TD>System:</TD>     <TD>router1 in </TD></TR>
#    <TR><TD>Maintainer:</TD> <TD></TD></TR>
#    <TR><TD>Description:</TD><TD>ATM2/0  </TD></TR>
#    <TR><TD>ifType:</TD>     <TD>sonet (39)</TD></TR>
#    <TR><TD>ifName:</TD>     <TD>AT2/0</TD></TR>
#    <TR><TD>Max Speed:</TD>  <TD>18.7 MBytes/s</TD></TR>
#  </TABLE>


### Interface 2 >> Descr: 'FastEthernet0/0' | Name: 'Fa0/0' | Ip:
'192.168.0.1' | Eth: '00-05-01-a0-7c-00' ###

Target[192.168.0.1_Fa0_0]: #Fa0/0:himitsu@192.168.0.1:
SetEnv[192.168.0.1_Fa0_0]: MRTG_INT_IP="192.168.0.1"
MRTG_INT_DESCR="FastEthernet0/0"
MaxBytes[192.168.0.1_Fa0_0]: 12500000
Title[192.168.0.1_Fa0_0]: Traffic Analysis for Fa0/0 -- router1
PageTop[192.168.0.1_Fa0_0]: <H1>Traffic Analysis for Fa0/0 --
router1</H1>
 <TABLE>
   <TR><TD>System:</TD>     <TD>router1 in </TD></TR>
   <TR><TD>Maintainer:</TD> <TD></TD></TR>
   <TR><TD>Description:</TD><TD>FastEthernet0/0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD>Fa0/0</TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>12.5 MBytes/s</TD></TR>
   <TR><TD>Ip:</TD>         <TD>192.168.0.1 ()</TD></TR>
 </TABLE>


### Interface 3 >> Descr: 'Ethernet1/0' | Name: 'Et1/0' | Ip: '' |
Eth: '00-05-01-a0-7c-1c' ###
### The following interface is commented out because:
### * it is operationally DOWN
#
# Target[192.168.0.1_Et1_0]: #Et1/0:himitsu@192.168.0.1:
# SetEnv[192.168.0.1_Et1_0]: MRTG_INT_IP=""
MRTG_INT_DESCR="Ethernet1/0"
# MaxBytes[192.168.0.1_Et1_0]: 1250000
# Title[192.168.0.1_Et1_0]: Traffic Analysis for Et1/0 -- router1
# PageTop[192.168.0.1_Et1_0]: <H1>Traffic Analysis for Et1/0 --
router1</H1>

後、省略
```

機能実装 – トラフィック監視 MRTGの使い方

- 独立コマンドとして作成されており、通常はcronにて定期的に起動する。（default：5分間隔）
  - `# crontab -l`
    `0-59/5 * * * /usr/local/sbin/mrtg /usr/local/etc/ix-foo.cfg`
    `#`
- RunAsDaemonしている際には以下のような設定をコンフィグに投入し、コマンドを投入
  - `RunAsDaemon:Yes`
    `Interval:5`
  - `$ mrtg --user=mrtg_user --group=mrtg_group mrtg.cfg`
- データ収集指定はconfigファイルのTargetレコードにて指定

実装検討2 – トラフィック監視 MRTG - Targetの指定法

- Keyword: Target - データ収集項目を指定
  - 例:
    - `Target[gw1-3]: 3:himitsu@gw1.aa.jp`
    - `Target[gw1-err-3]: ifInErrors.3&ifOutErrors.3:himitsu@gw1.aa.jp`
    - `Target[gw1-if-1]: -/10.0.0.101:himitsu@gw1.aa.jp`
    - `Target[gw1-pingloss]: `/usr/local/bin/check_loss.sh gw1``
- SNMPデータの収集
- 外部コマンド結果の埋め込み収集

実装検討2 – トラフィック監視
MRTG - Targetの指定法:SNMP 1

- SNMPデータの収集
  - `Target[<target name>]:`
    `<target kind>:<community>@<address>`

    - `<target name>` : 測定機器の名称
    - `<target kind>` : 測定項目
    - `<community>` : 測定機器に設定しているcommunity string
    - `<address>` : 測定機器のアドレス・ホスト名

実装検討2 – トラフィック監視 測定項目

- 各ネットワークノードのポートにおいて以下の項目を測定する
  - トラフィック
    - bps (incoming/outgoing)
    - pps (incomfing/outgoing)
  - エラー関係
    - packet discards (incoming/outgoing)
    - interface errors (incoming/outgoing)

実装検討2 – トラフィック監視
使用するSNMP OID/MIB Symbols

- [interfaces.ifTable.ifEntry] group
  - 1.3.6.1.2.1.2.2.1.1 : ifIndex
  - 1.3.6.1.2.1.2.2.1.2 : ifDescr
  - 1.3.6.1.2.1.2.2.1.3 : ifType
  - 1.3.6.1.2.1.2.2.1.7 : ifAdminStatus
  - 1.3.6.1.2.1.2.2.1.8 : ifOperStatus
  - 1.3.6.1.2.1.2.2.1.10 : ifInOctets
  - 1.3.6.1.2.1.2.2.1.16 : ifOutOctets
  - 1.3.6.1.2.1.2.2.1.11 : ifInUcastPkts
  - 1.3.6.1.2.1.2.2.1.17 : ifOutUcastPkts
  - 1.3.6.1.2.1.2.2.1.13 : ifInDiscards
  - 1.3.6.1.2.1.2.2.1.19 : ifOutDiscards
  - 1.3.6.1.2.1.2.2.1.14 : ifInErrors
  - 1.3.6.1.2.1.2.2.1.20 : IfOutErrors

実装検討2 – トラフィック監視 mrtg configの作り方

- 測定項目はひとつの対象に対して以下の4項目
  - bps, pps, packet discards, interface err
  - これらは独立したコンフィグとしてまとめるのがやりやすいが、indexmakerを使ってindex.htmlを作ることを考えると、正常トラフィック (bps, pps) とエラー系トラフィック (discards, error) にまとめるのが使いやすい。
  - 測定結果ディレクトリはマシンごとにまとめる
- Target指定のキー項目
  - ifIndex指定が一番素直だが、indexとインタフェースの関連を人がとらなければならない。リブートするとifIndexの対応表は変わってしまうことがある
  - IPアドレス指定はルータのように全インタフェースにアドレスがある場合には有効 ⇒ だが、アドレスのないスイッチのポートには適用できない
  - Interface Description指定もしくはInterface Name指定にて作成するのが簡単

実装検討2 – トラフィック監視 ディレクトリ構成

- /usr/local/mrtgのディレクトリ構成
  - /usr/local/mrtg
    /usr/local/mrtg/bin
    /usr/local/mrtg/lib
    /usr/local/mrtg/conf
    /usr/local/mrtg/data/fw0/
    /usr/local/mrtg/data/branch-fw0/
    /usr/local/mrtg/data/dmz-sw0/
    /usr/local/mrtg/data/intra-sw0/
    /usr/local/mrtg/data/branch-sw0/

- 測定コンフィグファイル構成

測定対象	データディレクトリ	測定分類	測定項目	コンフィグファイル名
fw0	/usr/local/mrtg/data/fw0/	トラフィック測定	bps, pps	fw0.cfg
		エラー測定	Discards, Errors	fw0-err.cfg
branch-fw0	/usr/local/mrtg/data/branch-fw0/	トラフィック測定	bps, pps	branch-fw0.cfg
		エラー測定	Discards, Errors	branch-fw0-err.cfg
dmz-sw0	/usr/local/mrtg/data/dmz-sw0/	トラフィック測定	bps, pps	dmz-sw0.cfg
		エラー測定	Discards, Errors	dmz-sw0-err.cfg
intra-sw0	/usr/local/mrtg/data/intra-sw0/	トラフィック測定	bps, pps	intra-sw0.cfg
		エラー測定	Discards, Errors	intra-sw0-err.cfg
branch-sw0	/usr/local/mrtg/data/branch-sw0/	トラフィック測定	bps, pps	branch-sw0.cfg
		エラー測定	Discards, Errors	branch-sw0-err.cfg

実装検討2 – トラフィック監視 mrtg configの作り方（続き）

- bps項目についてはGigabit Ethernetの測定にて注意が必要
  - ifInOctets/ifOutOctes　は32bit正数
  - 5分間隔の測定をした場合、114Mbps付近でカウンターがゼロリセットされてしまう。
  - 対処方法：
    - MRTG ver 2.9以上にてSNMPv2c 64bit counter MIBを使用する
      - `Target[192.168.0.1_gi_0_1]: 2:himitsu@router1:::::2`
    - 測定周期をDefault=5分以下の間隔にて測定を行う
      - `0-59/3 * * * /usr/local/sbin/mrtg ./ix-foo.cfg`
      - とはいってもこの設定では5分/3分=166%。いうことで増分66%（=190Mbps）を超えるとやはりカウンターがゼロリセットされる・・・
    - カウンターリセットしないEnterprise MIBを使用する
      - Cisco Enterprise MIB：`locIfInBitsSec　= .1.3.6.1.4.1.9.2.2.1.1.6`
      - Cisco Enterprise MIB：`locIfOutBitsSec = .1.3.6.1.4.1.9.2.2.1.1.8`

実装検討2 – トラフィック監視
config file: bps/pps: fw0.cfg

```
#####
# fw0 bps/pps config - fw0.cfg
###
WorkDir: /usr/local/mrtg/data/fw0/
IconDir: /mrtg-icons/
Forks: 4

Target[fw0-e1-bps]: ¥ethernet1:himitsu@172.16.0.1
MaxBytes[fw0-e1-bps]: 100000000
Title[fw0-e1-bps]: fw0: ethernet1 bps
PageTop[fw0-e1-bps]: <H1>fw0: ethernet1 bps</H1>
Options[fw0-e1-bps]: gauge,growright

Target[fw0-e1-pps]: ifInUcastPkts¥ethernet1&ifOutUcastPkts¥ethernet1:himitsu@172.16.0.1
MaxBytes[fw0-e1-pps]: 500000
Title[fw0-e1-pps]: fw0: ethernet1 pps
PageTop[fw0-e1-pps]: <H1>fw0: ethernet1 pps</H1>
Options[fw0-e1-pps]: growright

【中略】

Target[fw0-e8-bps]: ¥ethernet8:himitsu@172.16.0.1
MaxBytes[fw0-e8-bps]: 100000000
Title[fw0-e8-bps]: fw0: ethernet8 bps
PageTop[fw0-e8-bps]: <H1>fw0: ethernet8 bps</H1>
Options[fw0-e8-bps]: gauge,growright

Target[fw0-e8-pps]: ifInUcastPkts¥ethernet8&ifOutUcastPkts¥ethernet8:himitsu@172.16.0.1
MaxBytes[fw0-e8-pps]: 500000
Title[fw0-e8-pps]: fw0: ethernet8 pps
PageTop[fw0-e8-pps]: <H1>fw0: ethernet8 pps</H1>
Options[fw0-e8-pps]: growright
#####
# fw0 bps/pps config - fw0.cfg end
###
```

機能実装 – トラフィック監視

config file: discards/errors: fw0-err.cfg

```
#####
# fw0 discards/errors config - fw0-err.cfg
###
WorkDir: /usr/local/mrtg/data/fw0/
IconDir: /mrtg-icons/
Forks: 4

Target[fw0-e1-discards]: ifInDiscards¥ethernet1&ifOutDiscards¥ethernet1:himitsu@172.16.0.1
MaxBytes[fw0-e1-discards]: 500000
Title[fw0-e1-discards]: fw0: ethernet1 discards
PageTop[fw0-e1-discards]: <H1>fw0: ethernet1 discards</H1>
Options[fw0-e1-discards]: gauge,growright

Target[fw0-e1-errors]: ifInErrors¥FastEthernet0/1&ifOutErrors¥FastEthernet0/1:himitsu@172.16.0.1
MaxBytes[fw0-e1-errors]: 500000
Title[fw0-e1-errors]: fw0:  ethernet1 errors
PageTop[fw0-e1-errors]: <H1>fw0:  ethernet1 errors</H1>
Options[fw0-e1-errors]: growright

【中略】

Target[fw0-e8-discards]: ifInDiscards¥ethernet8&ifOutDiscards¥ethernet8:himitsu@172.16.0.1
MaxBytes[fw0-e8-discards]: 500000
Title[fw0-e8-discards]: fw0: ethernet8 discard
PageTop[fw0-e8-discards]: <H1>fw0: ethernet8 discards</H1>
Options[fw0-e8-discards]: gauge,growright

Target[fw0-e8-errors]: ifInErrors¥FastEthernet0/12&ifOutErrors¥FastEthernet0/12:himitsu@172.16.0.1
MaxBytes[fw0-e8-errors]: 500000
Title[fw0-e8-errors]: fw0: ethernet8 errors
PageTop[fw0-e8-errors]: <H1>fw0: ethernet8 errors</H1>
Options[fw0-e8-errors]: growrigh
#####
# fw0 discards/errors config - fw0-err.cfg end
###
```

実装検討2 – トラフィック監視

config file: bps/pps: dmz-sw0.cfg

```
#####
# dmz-sw0 bps/pps config – dmz-sw0.cfg
###
WorkDir: /usr/local/mrtg/data/dmz-sw0/
IconDir: /mrtg-icons/
Forks: 4

Target[dmzsw0-gi0-1-bps]: ¥GigabitEthernet0/1:himitsu@172.16.250.10:::::2
MaxBytes[dmzsw0-gi0-1-bps]: 1000000000
Title[dmzsw0-gi0-1-bps]: dmz-sw0: GigabitEthernet0/1 bps
PageTop[dmzsw0-gi0-1-bps]: <H1>dmz-sw0: GigabitEthernet0/1 bps</H1>
Options[dmzsw0-gi0-1-bps]: gauge,growright

Target[dmzsw0-gi0-1-pps]: ifInUcastPkts¥GigabitEthernet0/1&ifOutUcastPkts¥Gigabitthernet0/1:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-1-pps]: 5000000
Title[dmzsw0-gi0-1-pps]: dmz-sw0: GigabitEthernet0/1 pps
PageTop[dmzsw0-gi0-1-pps]: <H1>dmz-sw0: GigabitEthernet0/1 pps</H1>
Options[dmzsw0-gi0-1-pps]: growright

【中略】

Target[dmzsw0-gi0-12-bps]: ¥GigabitEthernet0/12:himitsu@172.16.250.10:::::2
MaxBytes[dmzsw0-gi0-12-bps]: 1000000000
Title[dmzsw0-gi0-12-bps]: dmz-sw0: GigabitEthernet0/12 bps
PageTop[dmzsw0-gi0-12-bps]: <H1>dmz-sw0: GigabitEthernet0/12 bps</H1>
Options[dmzsw0-gi0-12-bps]: gauge,growright

Target[dmzsw0-gi0-12-pps]: ifInUcastPkts¥GigabitEthernet0/12&ifOutUcastPkts¥GigabitEthernet0/12:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-12-pps]: 5000000
Title[dmzsw0-gi0-12-pps]: dmz-sw0: GigabitEthernet0/12 pps
PageTop[dmzsw0-gi0-12-pps]: <H1>dmz-sw0: GigabitEthernet0/12 pps</H1>
Options[dmzsw0-gi0-12-pps]: growright
#####
# dmz-sw0 bps/pps config – dmz-sw0-if.cfg end
###
```

機能実装 – トラフィック監視

config file: discards/errors: dmz-sw0-err.cfg

```
#####
# dmz-sw0 discards/errors config - dmz-sw0-err.cfg
###
WorkDir: /usr/local/mrtg/data/dmz-sw0/
IconDir: /mrtg-icons/
Forks: 4

Target[dmzsw0-gi0-1-discards]: ifInDiscards¥GigabitEthernet0/1&ifOutDiscards¥GigabitEthernet0/1:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-1-discards]: 500000
Title[dmzsw0-gi0-1-discards]: dmz-sw0: GigabitEthernet0/1 discards
PageTop[dmzsw0-gi0-1-discards]: <H1>dmz-sw0: GigabitEthernet0/1 discards</H1>
Options[dmzsw0-gi0-1-discards]: gauge,growright

Target[dmzsw0-gi0-1-errors]: ifInErrors¥GigabitEthernet0/1&ifOutErrors¥GigabitEthernet0/1:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-1-errors]: 500000
Title[dmzsw0-gi0-1-errors]: dmz-sw0: GigabitEthernet0/1 errors
PageTop[dmzsw0-gi0-1-errors]: <H1>dmz-sw0: GigabitEthernet0/1 errors</H1>
Options[dmzsw0-gi0-1-errors]: growright

【中略】

Target[dmzsw0-gi0-12-discards]: ifInDiscards¥GigabitEthernet0/12&ifOutDiscards¥GigabitEthernet0/12:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-12-discards]: 500000
Title[dmzsw0-gi0-12-discards]: dmz-sw0: GigabitEthernet0/12 discards
PageTop[dmzsw0-gi0-12-discards]: <H1>dmz-sw0: GigabitEthernet0/12 discards</H1>
Options[dmzsw0-gi0-12-discards]: gauge,growright

Target[dmzsw0-gi0-12-errors]: ifInErrors¥GigabitEthernet0/12&ifOutErrors¥GigabitEthernet0/12:himitsu@172.16.250.10
MaxBytes[dmzsw0-gi0-12-errors]: 500000
Title[dmzsw0-gi0-12-errors]: dmz-sw0: GigabitEthernet0/12 errors
PageTop[dmzsw0-gi0-12-errors]: <H1>dmz-sw0: GigabitEthernet0/12 errors</H1>
Options[dmzsw0-gi0-12-errors]: growright
#####
# dmz-sw0 discards/errors config - dmz-sw0-err.cfg end
###
```

実装検討2 – トラフィック監視 パフォーマンス調整のTIPS

- 測定項目数
  - 以下のようにインタフェース数を仮定した場合：全測定項目は 232項目
    - fw0 (8FE)
    - branch-fw0 (2FE)
    - dmz-sw0 (12GbE)
    - intra-sw0 (24GbE)
    - branch-sw0 (12GbE)

機器名称	インタフェース本数		各IF毎の測定項目数	測定項目数
	FE / E	GbE		
fw0	8	0	4	32
branch-fw0	2	0	4	8
dmz-sw0	0	12	4	48
intra-sw0	0	24	4	96
branch-sw0	0	12	4	48
合計				232

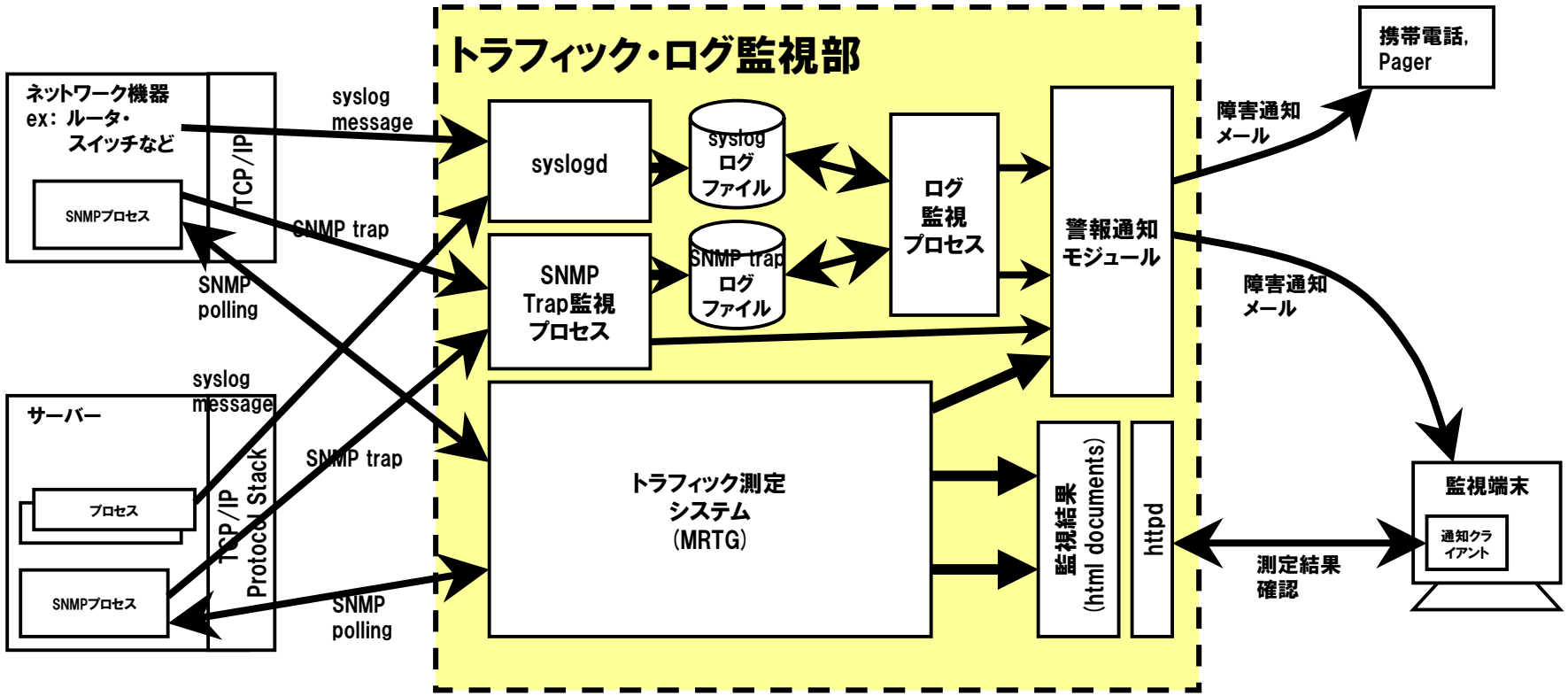
  - すべての計測を同時に実施した場合、5分ごとに過負荷となる可能性が高い

- パフォーマンス改善のための対処：
  - Forks: 指定で並列Query
    - 測定対象が無応答状態となったときには、無応答Queryだけ保留され、他の計測に影響しないため動作の保険になる。
    - Forks: 4
  - 起動順番を調整する。スタート基準は1分間隔
    - 0,5分スタート組、1,6分スタート組、2,7分スタート組、3,8分スタート組、4,9分スタート組

実装検討2 – トラフィック監視
crontab – mon0.aa.jp

```
####
# crontab mrtg@mon0.aa.jp
##
# fw01 mrtg
0-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/fw0.cfg > /dev/null 2>&1
2-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/fw0-err.cfg > /dev/null 2>&1

# fw01 mrtg
1-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/branch-fw0.cfg > /dev/null 2>&1
3-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/branch-fw0-err.cfg > /dev/null 2>&1

# dmz-sw0 mrtg
2-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/dmz-sw0.cfg > /dev/null 2>&1
4-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/dmz-sw0-err.cfg > /dev/null 2>&1

# intra-sw0 mrtg
0-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/intra-sw0.cfg > /dev/null 2>&1
2-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/intra-sw0-err.cfg > /dev/null 2>&1

# branch-sw0 mrtg
1-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/branch-sw0.cfg > /dev/null 2>&1
3-59/5 * * * * /usr/local/mrtg/bin/mrtg /usr/local/mrtg/conf/branch-sw0-err.cfg > /dev/null 2>&1
####
# crontab mrtg@mon0.aa.jp end
##
```

監視システムのモデル －トラフィック・ログ監視サーバ



index

TIPS - まずは

- ツールの挙動確認はまずオフラインで
  - 監視・測定ツールでネットワークに障害を与えることができる

TIPS – ping編

- ショートパケットが通ったからといって安心できない。開通確認はロングパケットで
  - トラフィックが多くなってくるとパケットが落ちるところも多い
  - ATM Megalink回線では必須。シェーピングレートの設定が失敗しているといきなり品質劣化して、通信障害となる
    - 私はpacket size=1400byte, count=1000以上、送出Interval=40msの設定で試験しています
  - スイッチのDuplexミスマッチもこれなら検知可能

- Internet経由の監視は タイムアウト＞1000msec
  - 22時～26時ぐらいの最繁時間帯は特に揺らぎが大きいため、マージンをとらないと誤検出が増える

TIPS - BB編1

- 監視対象拡大に伴う問題
  - 規模が大きくなると、NMSがポーリングして統計処理を行う時間も増加する
  - 監視対象機器を適正な数に抑えないと・・・
    - 次のポーリングタイミングまで計測が終らない
  - 適正範囲に分割が必要
    - 規模拡大時に見落としやすいので注意

TIPS - BB編2

- Longer than Sleeptime XXXがでたら環境限界の印
  - BBのシステムログは$BBHOME/BBOUT。これをチェック！
  - Longer than Sleeptimeメッセージは監視間隔以内に監視が終わらないというシステムメッセージ
    - `Thu Nov  1 06:12:07 JST 2001 bbrun: (/usr/local/bb/ext/fping.sh) Runtime 517 longer than Sleeptime 300`
    - `Thu Nov  1 06:13:21 JST 2001 bbrun: (/usr/local/bb/bin/bb-network.sh) Runtime 346 longer than Sleeptime 300`
  - マシンスペックのグレードアップ・監視サーバ分割を視野にいれた、システム環境・チューニングを含めた見直しが必要

TIPS - BB編3

- Big Brotherの高速化：fping + fping.sh
  - `http://www.fping.org/`
  - `http://www.deadcat.net/cgi-bin/download.pl?section=1&file=fping.sh`
  - fpingによりping試験を高速化
    - bbdef.sh内にて CONNTEST=FALSE としてBBのping試験を停止する必要あり
    - 一桁以上速度が向上！

- Big Brotherサーバのシステム監査ログには注意が必要
  - BBの基本はshell scriptとなっており必要な機能は外部コマンドで実現されている。よって一回の監視フェーズにおいて数十のプログラムが起動される
    - Accountingログが短時間に巨大になる
    - ログ領域の拡大。細かなメンテナンス
    - もしくはアカウンティングを停止
      - # accton

TIPS – MRTG編1

- データの方向性に注意
  - 対向している装置で同じポートを測定するとIn/Outが逆の結果がでる
  - 対外線を出口として、ここを起点にデータが流れるように設定すると考えやすい
- データの単位に注意
  - ifInOctets/ifOutOctetsはOctet単位系
  - 回線・物理接続速度はbps。つまりbit単位系
    - Options [hoge] bitsした上でMaxbytes [hoge] を8倍する
- IP address/MAC address/Comment指定Targetを効果的に使う

TIPS – MRTG編2

- Cronからのメッセージには注意
  - 必ずMRTGのエラーメッセージは取得できるようにする
    - /etc/aliases
    - ~/.forward
- 深刻なメッセージ
  - Config Error
  - No Response
  - Lockfile found

TIPS – MRTG編3

- 非常に深刻なメッセージ

```
 From: mrtg@mon0.aa.jp (Cron Daemon)
 To: alert@aa.jp.jp
 Date: Fri, 13 Oct 2003 02:03:16 +0900 (JST)
 Subject: Cron <mrtg@mrtg1> /usr/local/mrtg/mrtg /usr/local/mrtg/conf/mrtg.cfg
--

ERROR: I guess another mrtg is running.
A lockfile (/usr/local/mrtg/conf/mrtg.cfg_l) aged 303 seconds is hanging around.
If you are sure that no other mrtg is running you can remove the lockfile
```

TIPS – MRTG編4

- では、逆手にとって、エラーメッセージによるネットワーク監視
  - 5分に毎に起動されるSNMP health checkという観点もある
    - MRTGのエラーメッセージを/dev/nullにするのはちょっともったいない
  - 経験的予兆
    - 同じインタフェースのno responseエラーが続いて上がってきたら、該当インタフェース回線のダウンか故障の可能性がある
    - どっと、まとめてエラーが帰ってきたら、ルータやスイッチなどのネットワーク障害が発生している可能性が高い

ご清聴ありがとうございました。

追加資料1：SNMP

監視する手段 – SNMP ポーリング

- 標準プロトコルベースでの管理方法
- サーバー・クライアント型プロトコル
  - サーバー：SNMPマネージャー
  - クライアント：ネットワーク機器（エージェント）
- ベンダーに依存せず、様々な機器において各種トラフィック・運用状況の監視が可能
- ルーターやインテリジェントスイッチから詳細情報を得るにはもっとも一般的
- アプリケーションサーバーでは個別にSNMP daemonを追加しなければいけない場合が多い
  - 商用製品が多い。例：HP OpenView

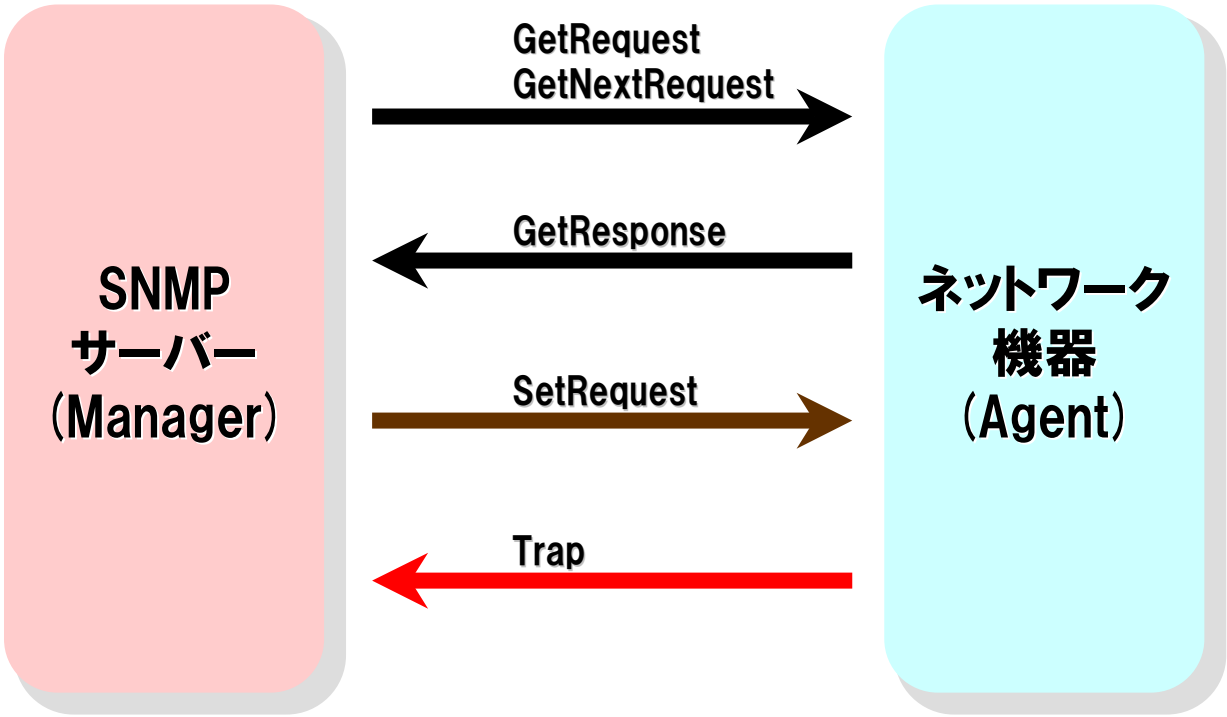
SNMP：Simple Network Management Protocol

- SNMP: Simple Network Management Protocol
  - UDP : polling port 161, trap port 162
- マルチベンダーを実現するための2つのフレームワーク
  - 情報取得のための簡潔なプロトコル
  - 取得情報を標準化するMIB（Message Information Base）
- 情報伝達の2つのモード
  - ポーリング
    - マネージャからエージェントに情報を要求する
  - トラップ
    - エージェントからマネージャーに対してイベントを転送する
- 3つのバージョンが規定される
  - version.1：基本プロトコルと情報管理体系を規定
    - community（パスワード）により権限を規制
    - read-only community/read-write community
  - version.2：IPブロックアクセス制限とView規制機能を追加
    - communityにアクセス可能IPブロックを規定可能
    - View定義によりアクセスできる情報を規制可能
  - version.3：ユーザ認証と暗号化機能を追加

SNMP Messages

- GetRequest : manager → agent
  - マネージャが更新情報を要求する
- GetNextRequest : manager → agent
  - マネージャがテーブルの次のエントリーを要求する
- GetResponse : manager ← agent
  - エージェントがマネージャからの要求に応答する
- SetRequest : manager → agent
  - マネージャが管理対象機器装置のデータを修正する
- Trap : manager ← agent
  - エージェントがマネージャにイベントを通知する

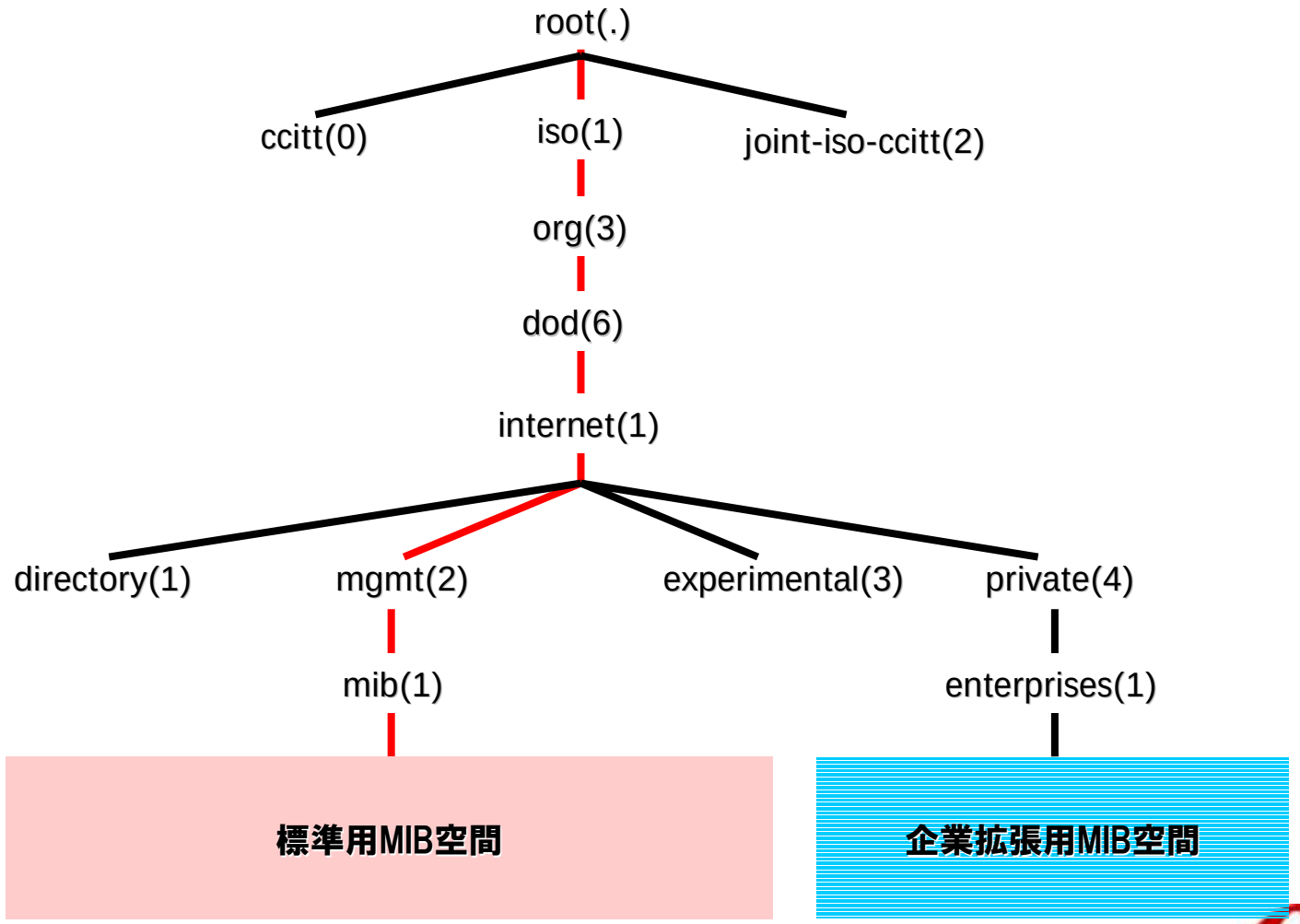
SNMP Message Hundling



MIB: Message Information Base

- RFC-1213 インターネット標準 MIBv2
- 階層的な命名体系で管理オブジェクトを定義
  - 木構造により情報を定義・管理
- オブジェクト識別子 (OID: Object ID) とMIB Symbol
  - 各木構造の枝に番号(OID)と識別子(MIB Symbol)を規定し、枝番号をたどることにより情報を階層管理する
- 標準勧告部分(MIBv2)と企業特有部分 (Enterprise MIB) に分かれる

MIB Tree



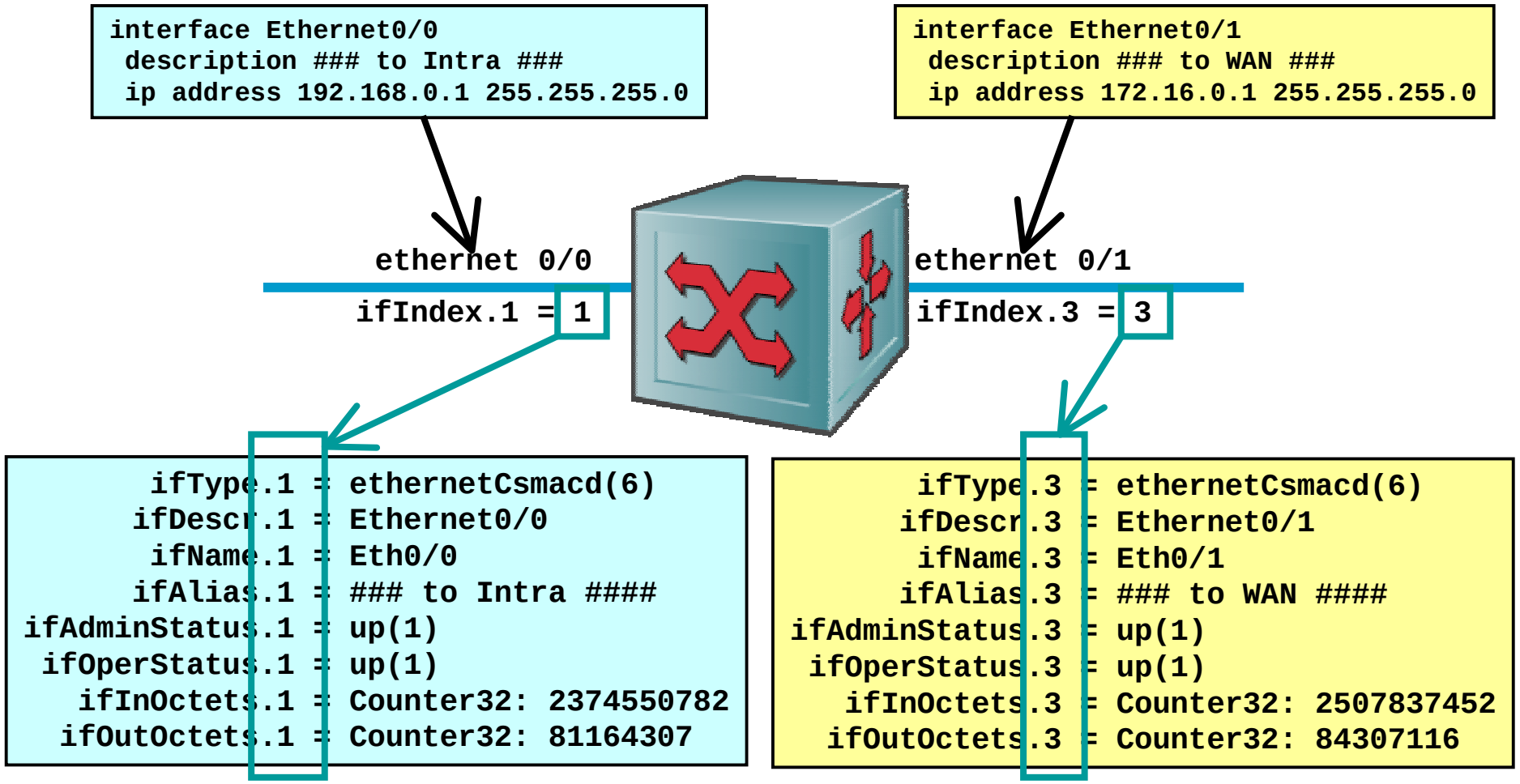
MIB OID / MIB Symbol

- 表記法:
  - (root) .iso (1) .org (3) .dod (6) .internet (1) .mgmnt (2) .mib (1) .
    1: system　　システムグループ
    2: interfaces　　インタフェースグループ
    3: at　　アドレス変換グループ
    4: ip　　IPグループ
    5: icmp　　ICMPグループ
    6: tcp　　TCPグループ
    7: udp　　UDPグループ
    11: snmp　　SNMPグループ
  - “.iso.org.dod.internet.mgmnt.mib.interfaces”が正式名称であるが、MIB Symbolは必ずユニークなシンボルを割り当てるルールとなっているためにmib (1) より前の部分は省略可能。interfaces MIB とよぶことが多い
    - 例 (Interface MIB) :
      .iso.org.dod.internet.mgmnt.mib.interfaces = .1.3.6.1.2.1.2
  - OIDは木構造をたどることで管理対象を決定する。
    - mib (1) より前の ".1.3.6.1.2.1" の部分については省略可能
    - 完全指定の場合は "."で始まり、数字で始まる場合には ".1.3.6.1.2.1" が省略されたものとして解釈する

基本SNMP OID/MIB Symbols

- [interfaces.ifTable.ifEntry] Group
  - .1.3.6.1.2.1.2.2.1.1 : ifIndex
    - インタフェースを識別するための管理番号。SNMP Interfaces MIBでは各インタフェース毎にユニークな番号を付与し、この番号をキーに各インタフェース毎の管理情報を管理する
  - .1.3.6.1.2.1.2.2.1.2 : ifDesc
    - インタフェースに付与された詳細コメント
  - .1.3.6.1.2.1.2.2.1.3 : ifType
    - インタフェースの媒体種別
- [ifMIB.ifMIBObjects.ifXTable.ifXEntry] Group
  - .1.3.6.1.2.1.31.1.1.1.1 : ifName
    - インタフェースに設定されている名称
  - .1.3.6.1.2.1.31.1.1.1.18 : ifAlias
    - インタフェースの別名。Descriptionコメントなどが適用される

ifIndexとインタフェースの関係



基本SNMP OID/MIB Symbols

- [interfaces.ifTable.ifEntry] Group
  - .1.3.6.1.2.1.2.2.1.7 ：ifAdminStatus
    - インタフェースの管理状態。保守停止(administratively down)されて稼動中(in operation)なのか状態を管理する
  - .1.3.6.1.2.1.2.2.1.8 ：ifOperStatus
    - インタフェースの稼動状態。インタフェースの稼動状態を管理する
  - .1.3.6.1.2.1.2.2.1.10：ifInOctets
  - .1.3.6.1.2.1.2.2.1.16：ifOutOctets
    - インタフェースの32bitオクテットカウンタ
  - .1.3.6.1.2.1.2.2.1.11：ifInUcastPkts
  - .1.3.6.1.2.1.2.2.1.17：ifOutUcastPkts
    - インタフェースのユニキャストパケット数カウンタ
  - .1.3.6.1.2.1.2.2.1.13：ifInDiscards
  - .1.3.6.1.2.1.2.2.1.19：ifOutDiscards
    - インタフェースの廃棄パケット数カウンタ
  - .1.3.6.1.2.1.2.2.1.14：ifInErrors
  - .1.3.6.1.2.1.2.2.1.20：IfOutErrors
    - インタフェースのエラー数カウンタ

SNMP OID/MIB Symbols - CISCO Enterprise MIB

- CISCO Enterprise MIB
  - locIfInBitsSec = .1.3.6.1.4.1.9.2.2.1.1.6
  - locIfInPktsSec = .1.3.6.1.4.1.9.2.2.1.1.7
  - locIfOutBitsSec = .1.3.6.1.4.1.9.2.2.1.1.8
  - locIfOutPktsSec = .1.3.6.1.4.1.9.2.2.1.1.9
  - locIfInRunts = .1.3.6.1.4.1.9.2.2.1.1.10
  - locIfInGiants = .1.3.6.1.4.1.9.2.2.1.1.11
  - locIfInCRC = .1.3.6.1.4.1.9.2.2.1.1.12
  - locIfInFrame = .1.3.6.1.4.1.9.2.2.1.1.13
  - locIfInOverrun = .1.3.6.1.4.1.9.2.2.1.1.14
  - locIfInIgnored = .1.3.6.1.4.1.9.2.2.1.1.15
  - locIfInAbort = .1.3.6.1.4.1.9.2.2.1.1.16
  - locIfResets = .1.3.6.1.4.1.9.2.2.1.1.17
  - locIfRestarts = .1.3.6.1.4.1.9.2.2.1.1.18
  - locIfLoad = .1.3.6.1.4.1.9.2.2.1.1.24
    - OID/MIBを使用する際には、Interfaceに対して"bandwidth"定義が必要
  - locIfCollisions = .1.3.6.1.4.1.9.2.2.1.1.25
  - locIfInputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.26
  - locIfOutputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.27
  - avgBusy1 = .1.3.6.1.4.1.9.2.1.57 (CPU usage)
  - avgBusy5 = .1.3.6.1.4.1.9.2.1.58 (CPU usage)

CISCOルータでのSNMP設定

- Cisco SNMP Query設定(ver.1, アクセス規制なし)：
  - (config)# snmp-server location tokyo-noc
    (config)# snmp-server contact admin@aa.jp
    (config)# snmp-server community READ-ONLY-COM RO
    (config)# snmp-server community READ-WRITE-COM RW
- Cisco SNMP Query設定(ver.1, アクセス規制あり)：
  - (config)# access-list 99 permit 172.16.0.0 0.0.0.255
    (config)# access-list 99 permit 172.16.4.0 0.0.0.255
    (config)# access-list 99 deny   any
    (config)# snmp-server location tokyo-noc
    (config)# snmp-server contact admin@aa.jp
    (config)# snmp-server community READ-ONLY-COM RO 99
    (config)# snmp-server community READ-WRITE-COM RW 99
- Cisco SNMP Trap設定(ver.1)：
  - (config)# snmp-server host 172.16.0.100 traps TRAP-COM
    (config)# snmp-server enable traps snmp
    (config)# snmp-server enable traps envmon
    (config)# snmp-server enable traps config
    (config)# snmp-server enable traps tty

CISCOルータでのSNMP設定2

- Cisco SNMP Query設定 (ver.2, アクセス/View規制あり)：
  - ```
    (config)# access-list 99 permit 172.16.0.0 0.0.0.255
    (config)# access-list 99 permit 172.16.4.0 0.0.0.255
    (config)# access-list 99 deny   any
    (config)# snmp-server location tokyo-noc
    (config)# snmp-server contact admin@aa.jp
    (config)# snmp-server view Not-Routing-Table mib-2 included
    (config)# snmp-server view Not-Routing-Table ip excluded
    (config)# snmp-server view Not-Routing-Table cisco excluded
    (config)# snmp-server community READ-RESTRICTED view Not-Routing-Table RO 99
    ```
    ```

Net-SNMP Package

- http://net-snmp.sourceforge.net/
- さまざまなUnixプラットフォームで稼動するSNMP Package

- 以下のコマンドを提供
  - snmpd, snmptrapd, snmpbulkwalk, snmpget, snmpset, snmptest, snmpusm, snmpcheck, snmpgetnext, snmpstatus, snmptranslate, snmpwalk, snmpdelta, snmpnetstat, snmptable, snmptrap

Net-SNMP – snmptrapd2

- SNMP trap eventを監視するdaemon
- trap eventごとに処理を規定することが可能
- Trap受信後、以下の処理を行う
  - 外部コマンドがアクションとして規定されている際には、アクションである外部コマンドの標準入力に受信したTrap eventを渡し、コマンドを起動する
- Trap受信によりアラートなどの通知を行うことが可能
- Snmptrapd.confの記述
  - `traphandle <OID> <action> <parameters･･･.>`
  - `traphandle default <action> <parameters･･･.>`

snmptrapd.conf

```
# SNMP Trap : Cold Start
traphandle .1.3.6.1.6.3.1.1.5.1 /usr/bin/mail -s "coldStart Trap" alert@aa.jp
# SNMP Trap : Warm Start
traphandle .1.3.6.1.6.3.1.1.5.2 /usr/bin/mail -s "warmStart Trap" alert@aa.jp
# SNMP Trap : Link Down
traphandle .1.3.6.1.6.3.1.1.5.3 /usr/bin/mail -s "linkDown Trap" alert@aa.jp
# SNMP Trap : Link Up
traphandle .1.3.6.1.6.3.1.1.5.4 /usr/bin/mail -s "linkUp Trap" alert@aa.jp
# SNMP Trap : Authentication Failure
traphandle .1.3.6.1.6.3.1.1.5.5 /usr/bin/mail -s "authFail Trap" alert@aa.jp
# SNMP Trap : Other
traphandle   default  /usr/bin/mail -s "Other Traps" alert@aa.jp
```

NetSNMP snmptrapd - 通知結果

```
From: log-admin <root@mon0.aa.jp>
To: alert@aa.jp
Date: Thu, 1 Nov 2001 22:01:49 +0900 (JST)
Subject: linkDown Trap

gw0.aa.jp
192.168.244.21
system.sysUpTime 24:10:03:09.12
.iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapOID
 .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTraps.linkDown
interfaces.ifTable.ifEntry.ifIndex.1 1
interfaces.ifTable.ifEntry.ifDescr.1 "GigabitEthernet1/0"
interfaces.ifTable.ifEntry.ifType.1 Fddi
enterprises.9.2.2.1.1.20.6 "administratively down"
.iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapEnt
erprise enterprises.9.1.48
```

TIPS - SNMP編1: アクセス規制

- SNMPに関する規制
  - SNMPは便利。しかし便利なものには必ず穴がある
    - セキュリティーホールになりやすい
    - SNMPでネットワークを落とすことも可能！
- Default communityはつかわない
  - Read only community != "public"
  - Write community != "private"
- 不要なrw、rwaはできるだけ使えないように設定する

TIPS - SNMP編2:アクセス範囲の限定

- SNMPクライアントにはアクセス規制が必須
  - 意外に狙われているルーター・スイッチ・www server
- SNMP package
  - libwrapをlink。hosts.allow/hosts.denyでアクセス規制する
    - ./configure --with-libwrap=・・・
- Cisco
  - SNMP ver2, 3で設定。とはいっても結構これが難しいので、
  - SNMP ver.1+アクセス規制用access-listの設定
- そんな機能のない装置は・・・
  - Private address blockにいれてしまう
  - ガードの低い装置をルーティング的にInternetから隔離する
    （例:Switching Hub, ...）

TIPS - SNMP編4: ifIndex問題

- パッケージタイプのルーター・スイッチは以下の事象においてifIndexとinterfaceの割付が変わる可能性がある
  - パッケージ障害交換
  - パッケージの増減設
  - 仮想インタフェースの増減設
  - その他...
- インタフェースの増減設が伴う際には監視ツールの設定を合わせて見直す

TIPS - SNMP編5: 使えるNet-SNMPコマンド例 (ver 5.0.9)

- NET-SNMP v4とv5はかなり挙動が違うので注意が必要

- $ snmpwalk -c himitsu 10.0.0.1 1
- $ snmpwalk -c himitsu 10.0.0.1 interface
- $ snmpwalk -c himitsu 10.0.0.1 ifDescr
- $ snmpwalk -c himitsu 10.0.0.1 ifType

- $ snmptranslate –Td -IR ifInDiscards
  - OIDの他にMIB Tree及び詳細説明を表示
- $ snmptranslate –Tp -IR Interfaces
  - Interface(2) MIB配下のMIB Treeを表示
- $ snmptranslate -On -Td .1.3.6.1.2.1.2.2.1.1
  - OIDをMIB Symbolに変換して表示

追加資料2：
MRTGのTargetの指定方法

MRTG - Targetの指定法

- Keyword: Target - データ収集項目を指定
  - 例:
    - `Target[gw1-3]: 3:himitsu@gw1.aa.jp`
    - `Target[gw1-err-3]: ifInErrors.3&ifOutErrors.3:himitsu@gw1.aa.jp`
    - `Target[gw1-if-1]: -/10.0.0.101:himitsu@gw1.aa.jp`
    - `Target[gw1-pingloss]: `/usr/local/bin/check_loss.sh gw1``
- SNMPデータの収集
- 外部コマンド結果の埋め込み収集

MRTG - Targetの指定法:SNMP 1

- SNMPデータの収集
  - `Target[<target name>]:`
    `<target kind>:<community>@<address>`

    - `<target name>` : 測定機器の名称
    - `<target kind>` : 測定項目
    - `<community>` : 測定機器に設定している community string
    - `<address>` : 測定機器のアドレス・ホスト名

MRTG - Targetの指定法:SNMP 2

- SNMPデータ収集指定方法
  - Port指定 (ifIndex指定)
  - SNMP OID指定 / SNMP MIB symbol指定
  - Interface Address指定
  - 組み合わせ指定
  - 新規追加の指定方法
    - MAC address指定
    - Description指定
    - Interface Name指定

MRTG - Targetの指定法:SNMP 3

- Port指定 (ifIndex指定)
  - SNMP Client側で管理しているPort番号 (ifIndex) を使ってデータ照会する。
  - ifInOctetsとifOutOctetsを測定
- 例1:Target[gw1-3]: 3:himitsu@gw1.aa.jp
  - gw1.aa.jpに収容されているifIndex=3のInterfaceに関して ifInOctets/ifOutOctetsを測定
- 例2:Target[gw1-3]: -3:himitsu@gw1.aa.jp
  - 例1のIn/Outを逆にしてデータ収集する

MRTG - Targetの指定法:SNMP 4

- SNMP OID指定 / SNMP MIB symbol指定
  - SNMP OID (Object ID) またはMIB symbolを指定し、データ照会する。
  - 変数1、変数2は"&"で連結指定する
- 例3: `Target[gw1-err-3]: ifInErrors.3&ifOutErrors.3:himitsu@gw1.aa.jp`
  - gw1.aa.jpに収容されているifIndex=3のInterfaceに関して `ifInErrors/ifOutErrors`を測定
- 例4: `Target[gw1-err-3]: 1.3.6.1.2.1.2.2.1.14.3& 1.3.6.1.2.1.2.2.1.20.3:himitsu@gw1.aa.jp`
  - 上の例のOID指定

MRTG – ifIndex指定の設定例

```
#------------------------------------------------------------------#
Target[ETHERNET11.0-BPS]: 8:himitsu@10.1.0.7
MaxBytes[ETHERNET11.0-BPS]: 155000000
Title[ETHERNET11.0-BPS]: router7: ETHERNET11.0 BPS
PageTop[ETHERNET11.0-BPS]: <H1>router7: ETHERNET11.0</H1>
Options[ETHERNET11.0-BPS]: bits,growright

#------------------------------------------------------------------#
Target[ETHERNET11.0-PPS]: ifInUcastPkts.8&ifOutUcastPkts.8:himitsu@10.1.0.7
MaxBytes[ETHERNET11.0-PPS]: 155000000
Title[ETHERNET11.0-PPS]: router7: ETHERNET11.0 PPS
PageTop[ETHERNET11.0-PPS]: <H1>router7: ETHERNET11.0 PPS</H1>
Options[ETHERNET11.0-PPS]: growright

#------------------------------------------------------------------#
```

MRTG - Targetの指定法:SNMP 5

- Interface Address指定1
  - パッケージタイプのルーター・スイッチはインタフェースの増減設によりPort番号(ifIndex)が変化する
  - loopbackやtunnel Interfaceのような仮想インタフェースもSNMP上では一つのポート番号をもつ
    - → ifIndexの割付が変化する可能性がある
  - 機器の構成変更の度に設定変更をさけるためにインタフェースに割り振られたアドレスをキーにしてデータ照会を行う
    - numberedで使われていることが前提！
  - デフォルトでは`ifInOctets`と`ifOutOctets`を測定

MRTG - Targetの指定法:SNMP 6

- Interface Address指定2
- 例5:Target[gw1-if-1]:
  /10.0.0.101:himitsu@gw1.aa.jp
  - gw1.aa.jpに収容されている10.0.0.101のInterfaceに関して
    ifInOctets/ifOutOctetsを測定

- 例6:Target[gw1-if-1]:
  -/10.0.0.101:himitsu@gw1.aa.jp
  - 例5のIn/Outを逆にしてデータ収集する

MRTG - Targetの指定法:SNMP 7

- 組み合わせ指定
  - Interface address指定とOID/MIB symbol指定を組み合わせる

- 例7:Target[gw1-if-1-disc]: ifInDiscards/10.0.0.101& ifOutDiscards/10.0.0.101:himitsu@gw1.aa.jp
  - gw1.aa.jpに収容されている10.0.0.101のInterfaceに関して ifInDiscards/ifOutDiscardsを測定

- 例8:Target[gw1-if-1-disc]: 1.3.6.1.2.1.2.2.1.13/10.0.0.101& 1.3.6.1.2.1.2.2.1.19/10.0.1.101:himitsu@gw1.aa.jp
  - 例7のOIDパターン

MRTG – Interface Address指定の設定例

```
#------------------------------------------------------------------#
Target[ETHERNET11.0-BPS]: /192.168.0.1:himitsu@10.1.0.2
MaxBytes[ETHERNET11.0-BPS]: 155000000
Title[ETHERNET11.0-BPS]: router2: ETHERNET11.0 BPS
PageTop[ETHERNET11.0-BPS]: <H1>router2: ETHERNET11.0</H1>
Options[ETHERNET11.0-BPS]: bits,growright

#------------------------------------------------------------------#
Target[ETHERNET11.0-PPS]:
ifInUcastPkts/192.168.0.1&ifOutUcastPkts/192.168.0.1:himitsu@10.1.0.2
MaxBytes[ETHERNET11.0-PPS]: 155000000
Title[ETHERNET11.0-PPS]: router2: ETHERNET11.0 PPS
PageTop[ETHERNET11.0-PPS]: <H1>router2: ETHERNET11.0 PPS</H1>
Options[ETHERNET11.0-PPS]: growright

#------------------------------------------------------------------#
```

MRTG - Targetの指定法:SNMP 8

- Interface Name指定
  - Interface Address指定はIP Addressをキーにしているために、switching hubのようにポートごとにアドレスをもたないものには適用できない。
  - これの状況に適応するためにInterfaceに割り振られたInterface名前をキーにしてデータ照会を行う
  - デフォルトではifInOctetsとifOutOctetsを測定
- 例9:
  - ```
    Target[sw1-2-11]: #2/11:himitsu@sw1.aa.jp
    Target[sw-2-11]: -#2/11:himitsu@sw1.aa.jp
    Target[sw-3-7]:
    1.3.6.1.2.1.2.2.1.14#3/7&1.3.6.1.2.1.2.2.1.20#3/7:himitsu@sw1.aa.jp
    Target[sw-3-7]: ifInErrors#3/7&ifOutErrors#3/7:himitsu@sw1.aa.jp
    ```
    ```

MRTG – Interface Name指定の設定例

```
#--------------------------------------------------------------------#
Target[gi1.1-bps]: #1/1:himitsu@10.1.0.2
MaxBytes[gi1.1-bps]: 1937500000
Title[gi1.1-bps]: switch1:  1/1 bps
PageTop[gi1.1-bps]: switch1:  1/1 bps
Options[gi1.1-bps]: bits,growright

#--------------------------------------------------------------------#
Target[gi1.1-pps]: ifInUcastPkts#1/1&ifOutUcastPkts#1/1:himitsu@10.1.0.2
MaxBytes[gi1.1-pps]: 500000
Title[gi1.1-pps]: switch1:  1/1 pps
PageTop[gi1.1-pps]: switch1:  1/1 pps
Options[gi1.1-pps]: growright

#--------------------------------------------------------------------#
```

MRTG - Targetの指定法:SNMP 9

- Interface Description指定
  - Interface Address指定では、故障時にポートの入れ替えなどが発生した際に、MRTG側の設定を修正しなければならない
  - サーバー側で対応するよりも収容変更先の装置の設定情報を元に変更できたほうが適応範囲が広いことから、これらのキーとしてInterfaceに割り振られるDescriptionをキーにデータ照会を行う
  - デフォルトではifInOctetsとifOutOctetsを測定
- 例9:
  - ```
    Target[sw1-2-11]: ¥to_web1:himitsu@sw1.aa.jp
    Target[sw-2-11]: -¥to_web1:himitsu@sw1.aa.jp
    Target[sw-3-7]:
    1.3.6.1.2.1.2.2.1.14¥to_web1&1.3.6.1.2.1.2.2.1.20¥to_web1:himitsu@sw1.aa
    .jp
    Target[sw-3-7]: ifInErrors¥to_web1&ifOutErrors¥to_web1:himitsu@sw1.aa.jp
    ```
    ```

MRTG – Interface Description指定の設定例

```
#------------------------------------------------------------------#
Target[gi1.1-bps]: ¥GigabitEthernet1/1:himitsu@10.1.0.1
MaxBytes[gi1.1-bps]: 1000000000
Title[gi1.1-bps]: router16:  GigabitEthernet1/1 bps
PageTop[gi1.1-bps]: <h1>router16:  GigabitEthernet1/1 bps</h1>
Options[gi1.1-bps]: bits,growright

#------------------------------------------------------------------#
Target[gi1.1-pps]:
ifInUcastPkts¥GigabitEthernet1/1&ifOutUcastPkts¥GigabitEthernet1/1:himitsu@10.1.0.1
MaxBytes[gi1.1-pps]: 500000
Title[gi1.1-pps]: router16:  GigabitEthernet1/1 pps
PageTop[gi1.1-pps]: <h1>router16:  GigabitEthernet1/1 pps</h1>
Options[gi1.1-pps]: growright

#------------------------------------------------------------------#
```

MRTG - Targetの指定法:コマンド埋め込み

- コマンド埋め込み指定
  - Target[<target name>]: `<command>`
    - <target name> : 測定機器の名称
    - <command> : 測定コマンド
      - “`“:バックシングルコーテーションでくくるのがミソ
  - コマンドの結果として4行の値が必要
    - 1行目:第1変数、通常 incoming bytes数
    - 2行目:第2変数、通常 outgoing bytes数
    - 3行目:文字列、targetのuptime
    - 4行目:文字列、targetの名称

MRTGによる品質計測

- 埋め込みコマンドによりSNMPでは計測が難しい品質測定なども可能となる
- 例：特定の2点間のpacket lossの定常監視
  - 一定間隔でpingによる定期監視を実施
    - ```
      # ping –i 0.02 -c 100 ftp.aa.jp
      PING ftp.aa.jp (192.168.101.238): 56 data bytes
      .
      --- ftp.aa.jp ping statistics ---
      100 packets transmitted, 95 packets received, 5% packet loss
      round-trip min/avg/max/stddev = 0.161/0.164/0.221/0.006 ms
      #
      ```
    - `-i 0.02 : supervisor only option.`
      FeeBSDのpingにおける指定。送出間隔を20ms。
      ネットワークに高負荷を強いることから取り扱い注意
      ```

MRTGによる品質計測 - check_loss.sh

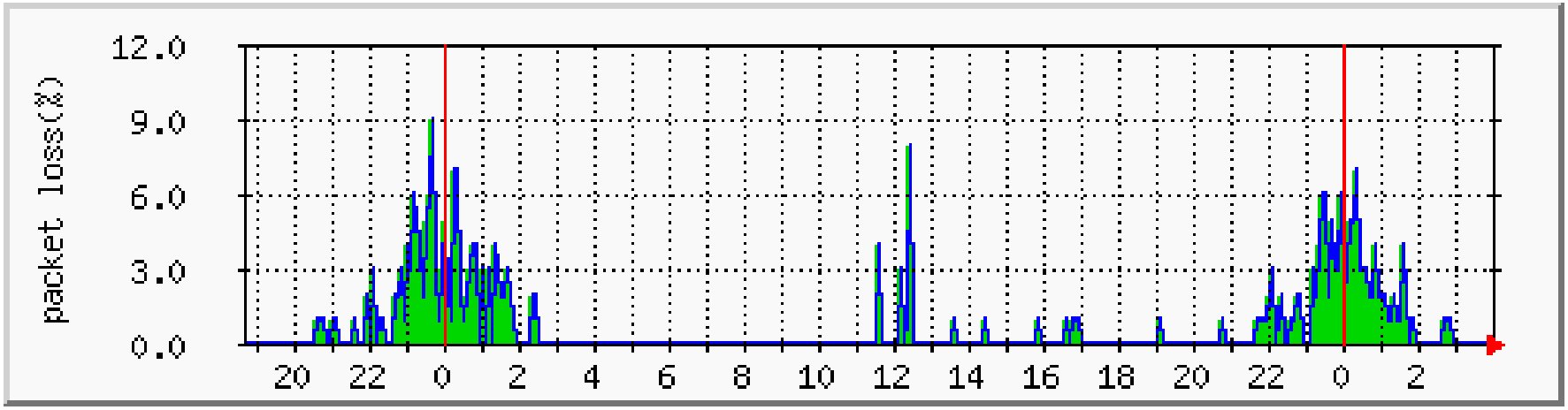
- pingの出力結果からpacket lossのデータを抽出
  - 100 packets transmitted, 95 packets received, 5% packet loss

```
# cat /usr/local/bin/check_loss.sh
#!/bin/sh
/sbin/ping -f -c 100 $1 | /usr/bin/sed 's/%//g' | /usr/bin/awk ’
    /packet loss/ { printf("%d¥n%d¥n", $7, $7)
  }’
echo 0 ; echo $*
# /usr/local/bin/check_loss2.sh ftp.aa.jp
5
5
0
/usr/local/bin/check_loss.sh ftp.aa.jp
#
```

MRTGによる品質計測 - ping-loss.cfg

```
# cat ping-loss.cfg
WorkDir: /usr/local/etc/www/mrtg/ping-loss

Target[pingloss-ftp]: `/usr/local/bin/check_loss.sh ftp.aa.jp`
Title[pingloss-ftp]: ftp.aa.jp - pingloss
MaxBytes[pingloss-ftp]: 100
PageTop[pingloss-ftp]: <H1> ftp.aa.jp - pingloss </H1>
YLegend[pingloss-ftp]: packet loss(%)
ShortLegend[pingloss-ftp]: %
LegendI[pingloss-ftp]:  loss:
LegendO[pingloss-ftp]:  loss:
Legend1[pingloss-ftp]: packet loss
Legend2[pingloss-ftp]: packet loss
Legend3[pingloss-ftp]: Maximal 5 Minute packet loss
Legend4[pingloss-ftp]: Maximal 5 Minute packet loss
Options[pingloss-ftp]: noinfo, growright, gauge, nopercent
#
```

MRTGによる品質計測 - 結果



参考資料:文献/URL

参考:Open Source/Free software link

- OSDN (Open Source Development Network)
  - http://www.osdn.com/
- OSDN.jp
  - http://osdn.jp/
- SOURCE FORGE
  - http://sourceforge.net/
- SOURCE FORGE JAPAN
  - http://sourceforge.jp/
- Fresh Meat – Free Software Index
  - http://www.freshmeat.net/
- Solaris Freeware Project
  - http://sunsite.sut.ac.jp/sun/solbin/

参考: 文献

- “Big Brotherで快適ネットワークシステム管理”
  - Software Design 2003/9,10,11,12, 連載中
  - 矢萩茂樹@イーアクセス / 越川康則@プラムシステムズ
    - 2003/9　: BigBrother概説
    - 2003/10 : BigBrotherのインストール
    - 2003/11 : BigBrotherサーバの詳細設定
    - 2003/12 : bbclientと拡張スクリプト
    - もう少し続きます

- “ネットワークにおける経路の安定性について”
  - 永見健一　インテックネットコア
  - http://www.janog.gr.jp/meeting/janog12/pdf/janog12_keiro_nagami.pdf
  - JANOG12 in 札幌 2003/7/24

ツールURL集1

- AWARE
  - http://www.elegant-software.com/software/aware/
- Big Brother
  - http://bb4.com/
  - Extensions Archive: http://www.deadcat.net/
- Big Sister
  - http://bigsister.sourceforge.net/
- Expect
  - http://expect.nist.gov/
- fping
  - http://www.fping.com/
- Ganglia
  - http://ganglia.sourceforge.net/
- hping
  - http://www.hping.org/
- IPTraf
  - http://cebu.mozcom.com/riker/iptraf/index.html

ツールURL集2

- Lire
  - http://www.logreport.org/
- LogSentry
  - http://www.gnu.org/directory/security/misc/LogSentry.html
- MRTG
  - http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/
- mon
  - http://www.kernel.org/software/mon
- monit
  - http://www.tildeslash.com/monit/
- moodss
  - http://jfontain.free.fr/moodss/
- Nagios (NetSaint)
  - http://www.nagios.org/
  - http://www.netsaint.org/
- NeTraMet
  - http://www.auckland.ac.nz/net/Accounting/ntm.Release.note.html

ツールURL集3

- MTR
  - http://www.bitwizard.nl/mtr/
- NISCA
  - http://nisca.sourceforge.net/
- Net-SNMP (UCD-SNMP)
  - http://www.net-snmp.org/
- ngrep – Network grep
  - http://ngrep.sourceforge.net/
- nocol/multiping
  - http://www.netplex-tech.com/software/nocol
- nPULSE
  - http://www.horsburgh.com/h_npulse.html
- ntop
  - http://www.ntop.org/

ツールURL集4

- RRDTool
  - http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
  - Frontend - Cacti
    - http://www.raxnet.net/products/cacti/
  - Frontend - CRICKET
    - http://cricket.sourceforge.net/
  - Frontend - NRG
    - http://nrg.hep.wisc.edu/
  - Frontend - ORCA
    - http://www.orcaware.com/orca/
  - Frontend – RRDBrowse
    - http://www.rrdbrowse.org/
  - Frontend – SmokePing
    - http://people.ee.ethz.ch/~oetiker/webtools/smokeping/

ツールURL集5

- Scotty
  - http://wwwhome.cs.utwente.nl/~schoenw/scotty/
- seafelt
  - http://seafelt.unicity.com.au/
- shepherd
  - http://atrey.karlin.mff.cuni.cz/~clock/twibright/shepherd/
- sing
  - http://sourceforge.net/projects/sing
- snort
  - http://www.snort.org/
- SPONG
  - http://spong.sourceforge.net/
- ssh
  - http://www.ssh.com/
- statscout
  - http://www.statscout.com
- SWATCH
  - http://www.oit.ucsb.edu/~eta/swatch/

ツールURL集6

- syslog-ng
  - http://www.balabit.hu/products/syslog-ng/
  - php-syslog-ng
    - http://www.vermeer.org/syslog/
- SysOrb
  - http://www.sysorb.com
- Treno
  - http://www.psc.edu/~pscnoc/treno.html
  - Experimental TCP Implementations
    http://www.psc.edu/networking/tcp.html
- visualroute
  - http://www.visualroute.com
- Zabbix
  - http://zabbix.sourceforge.net/

参考：URL集1

- みっきーのネットワーク研究所
  - http://www.hawkeye.ac/micky/
- EXP.（旧名：「働け!! linux!!」）
  - http://www.tujige.info/manage/index.html
- いちばん近道なLinuxマスター術
  - http://www.zdnet.co.jp/help/howto/linux/0007master/
  - 「第4回： システムログの読み方を理解しよう」
    - http://www.zdnet.co.jp/help/howto/linux/0007master/04/
  - 「第6回： SNMPによるネットワークモニタリング」
    - http://www.zdnet.co.jp/help/howto/linux/0007master/06/

参考：URL集2

- SNMP FAQ
  http://www.cis.ohio-state.edu/hypertext/faq/usenet/snmp-faq/part1/faq.html
- SIMPLE WEB
  - http://www.simpleweb.org/software/
- Cisco SNMP TIPs&FAQs
  - http://www.cisco.com/japanese/warp/public/3/jp/service/tac/l3_technology-tcpip_a.html#a5
- Cisco device SNMP configuration tips
  - http://www.cisco.com/pcgi-bin/Support/browse/psp_view.pl?p=Internetworking:SNMP&s=Implementation_and_Configuration#Samples_and_Tips

参考：組織

- IETF
  - http://www.ietf.org/
- NANOG
  - http://www.nanog.org/
- JANOG
  - http://www.janog.gr.jp/
- CAIDA
  - http://www.caida.org/tools/
    - cflowd ,RRD ･･･etc
- LBNL's Network Research Group
  - http://ee.lbl.gov/
    - tcpdump, libpcap , arpwatch, traceroute, pathchar