C10:ネットワーク管理と監視フリーソフトの利用法

佐藤 友治　株式会社インターネット総合研究所
矢萩 茂樹　インテリジェント・テレコム株式会社

チュートリアルの目的

- インターネット、コンピュータネットワークの設計、運用に関する見方、考え方をレビューする
- フリーソフトによるネットワーク監視
- ネットワークの（再）設計の手掛かり
- セキュリティ、トラブルシューティング、経路制御は他のチュートリアルを参考に。

フリーソフトについて

- ここでは、 GPL,Opensource等フリーソフトについて厳密な定義はしない
- ネットワーク等で手軽に入手でき、商品より緩やかな使用許諾条件のソフト全般を対象としている。
- 使用許諾条件、ライセンス等は各自で確認してください。
- 最近アップデートがないソフトもあります
  - ヒット数を尺度にするのもよいかもしれません
- セキュリティに注意
  - 例：TCP-Wapper 「トロイの木馬」

チュートリアルの時間割

- 第1部：ネットワーク管理の基礎知識
  IRI 佐藤　　１時間
- 第2部：フリーソフトによるネットワーク監視
  ITNet 矢萩　１時間30分
- 第3部：TIPS-質問
  佐藤・矢萩　１時間

第１部：ネットワーク管理の基礎知識

佐藤 友治
株式会社インターネット総合研究所
1999/12/15

ネットワーク管理の変遷

- 10年前迄
  - **メインフレーム、ミニコンなど階層構造のネットワークを管理**
- 1980年代末
  - **NFSによるディスクやNISによるアカウントの共有**
- 1990年代
  - **ネットワーク管理者の仕事が急増**
    - **UNIXシステム以外に、ルータなどのネットワーク機器も管理**
    - **経路制御が重要な業務になった**

ネットワーク管理の現在

- 多用なシステムを管理対象とする
  - **Unixマシンからパーソナル・コンピュータ ネットワーク機器（ルータ等）まで**
- 急増する管理対象機器を把握
- ネットワーク環境を前提とした業務の増加
- ネットワーク・ユーザが技術的な知識を有することを前提とした管理はできない

新たに考えなければならない管理作業

- 共有資源の管理
- ネットワーク・サービスの管理
- 複雑化するセキュリティ管理
- ユーザ環境の整備
- ユーザ教育と啓発

コンピュータネットワークの規模

- ISP
- エンタープライズ
- キャンパス
- Small Office,Home Office(SOHO)

ネットワーク管理を行う理由

- 機器の稼動状況を把握する
  - **アラーム処理**
    - **スレッシュホールド条件が起こったことを視覚的・聴覚的に通知**
- 個人の生産性とコストの制御
- ネットワークの規模と複雑さの管理
- ネットワーク計画

SOHO

- 数台のマシン
- 単一セグメント
- 外部接続
  - ダイアルアップ
  - 64～128Kbps常時接続
- DHCP、PPP

キャンパス・ネットワーク

- 比較的フラットなネットワーク
- バックボーン部とリーフ部
- 外部コネクティビティの管理
- バックボーン技術
  - 100BASE-TX/FX,ATM,GigabitEther,FDDI
- 経路制御
  - OSPFの導入

エンタープライズ・ネットワーク

- セキュリティ
  - **フィルター（スクリーニング）、ファイアウォール**
  - **アカウンティング**
- ポリシー
- NAT
- VLAN
  - **物理トポロジーと論理トポロジーが違う**

サービスプロバイダ

- 接続サービス
  - **認証、アカウンティング**
- バックボーン設計
  - **高速ネットワーク技術**
  - **通信技術**
  - **経路制御**
- 負荷分散
- 相互接続
  - BGP

ネットワーク分析・管理設計の考察

- トータルコストの低減
  - **人件費が一番高い**
  - **運用は集約したほうが効率的**
- 単純なネットワーク構成
  - **スタート時から管理・監視の設計を行う。**
  - **In Band か Out Band のどちらで管理するか**
  - **イニシャルコスト**
- 回線負荷を減らすための機能分散の可否
  - **サーバ**
  - **外部との接続**
- 信頼できるデータ転送を構築する
- 重要なリンクを監視する
  - **プライベートアドレスで監視ネットワークをつくる**

ネットワーク管理・監視心得

- 手に余る管理は目指さない
  - **上司、組織内での理解**
  - **権限の委譲**
- 正常な状況の把握と日常監視
  - **ネットワーク設計書の把握**
  - **自動化**
  - **履歴管理**
  - **インベントリー管理**
- 健康管理と同じ
  - **リモートメンテナンス用機器等、たまに使う機器もチェックする**
- 人間と違って平均値はない
  - **平均値を把握することの重要度が高い**

管理ポリシーの策定

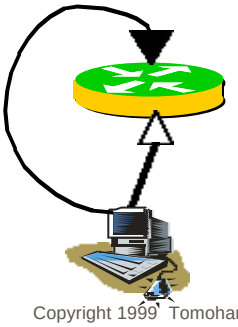
- 何を管理対象とするか
- どのような管理作業をおこなうか
- 管理作業の担当者、責任者は誰か
- トラブルが発生した際にどのような手順で対応するか
- どこまでがユーザの責任範囲か
- 管理作業に付随する作業（作業報告など）にはどのようなものがあるか

ネットワークの監視

- (1)ネットワークの動きを理解する
  - **リアルタイム監視とオフライン・レポート**
  - **トラフィック・フロー**
  - **利用度**
  - **ネットワーク利用の傾向を特定、分析**
  - **アクティビティの監視**
- (2)ネットワーク負荷のバランスをとる
- (3)今後の計画

監視ポイントの設定

- 監視基点とモニタリングする方向
- Inside と Outgoingの区別がつくように



ルータ、スイッチでは何を監視するか？

- インターフェース毎のトラフィック
- パケット　ロス
- コリジョン

経路制御

- 実トラフィックの経路の分析
  - **大量のPCの利用**
  - **ブロードキャスト、マルチキャストアプリケーション**
- ルータ、スイッチの必要性と適用
- 経路の冗長性の高いネットワークの存在
- ダイナミックルーティングの必要性

サーバでは何を管理・監視するか

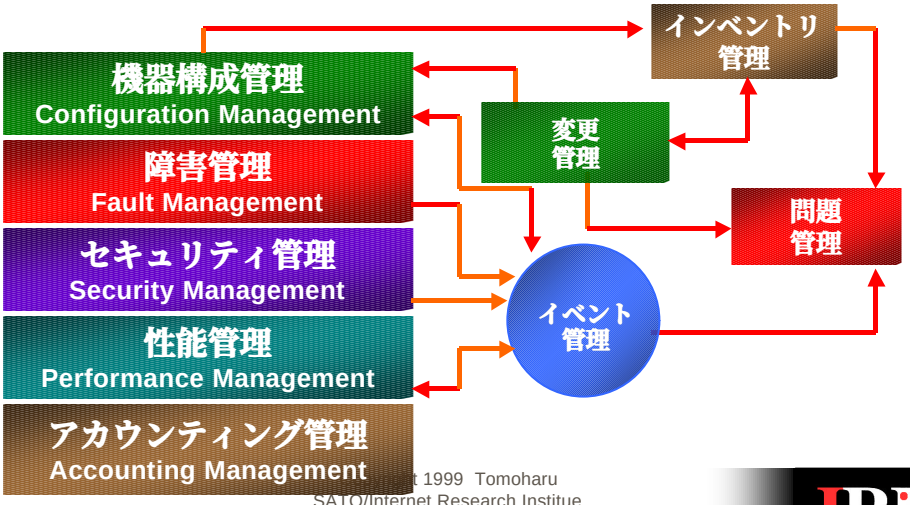
- サーバ全体
  - **インターフェース**
  - **ディスク**
  - CPU
- 用途の分析
  - **ユーザ数**
  - **トランザクション数**
  - **アプリケーション**
- プロセス毎(サービス毎)
  - **トランザクション**
  - **エラーステート**
- セキュリティ
  - **ファイアウォール自身の管理を忘れがち**
  - **考えすぎて、ファイアウォールの監視の口を塞いでしまわないように**
- シミュレーション

運用開始後の監視

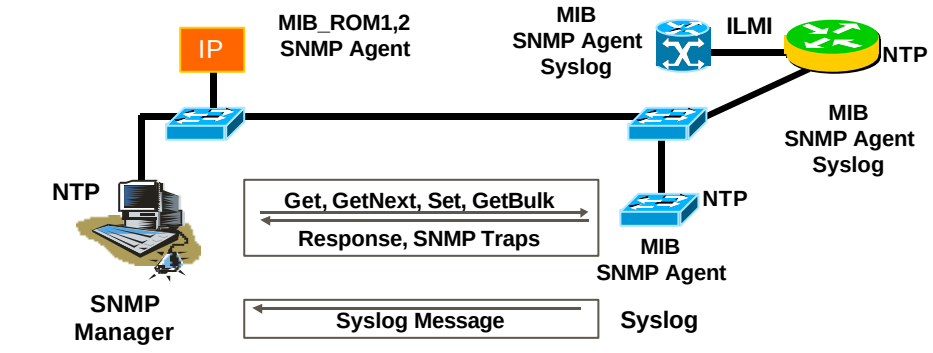
- 正常に動作しているか
- ボトルネックはないか？
- vmstat, iostat, sar, netstat, ps などでチェック
  - **あるマシンの正常運用状態**
  - **破綻状態**

ネットワーク
マネージメントシステム

ネットワーク管理構造



ネットワーク管理技術のベース



IP コネクティビティ	SNMP Traps/RMON	Syslog	Network Time Protocol	Neighbor Discovery ILMI

ルータのネットワーク管理設定準備

- 管理IPアドレスのアサイン
  - ルータ：Loopback0
    - SNMPの情報をやり取りするIPアドレスを決める
- SNMP community
  - public ReadOnly
  - pr1v8 ReadWrite (“private”は使わない)
- Logging (Syslog Message用)
- NTP (Network Time Protocol)

ルータ追加コンフィギュレーション (例：Cisco)

- Enable Telnet access
  - (“line vty 0 4” and enable passwords)
- Hostname, SNMP contact, location, chassis-id
- User login authorization local か TACACS(+)による管理者の認証と階層的管理の導入
- SNMP access lists

最近の動向: セキュアな監視機能への対応

- SSH
  - 最近、ハイエンド・ネットワーク機器のファームウェアに搭載れている
  - Cisco：IOS12.*S,Juniper：JUNOSなど
  - Expectで処理を「** over SSH」化してみるか
- IPsec
- SNMP3
  - PDUの暗号化
- 暗号鍵やCAの扱いはどうなるか？

監視の種類

- イベント監視
  - 今を見る
    - SNMPトラップ
    - Syslog
    - コマンドでチェック
- トラフィックモニタ
  - これまでの傾向を見る
    - SNMPでデータ取得
    - コマンドでデータ取得
- パケットモニタリング
  - ネットワークを流れるパケット自体をモニタする
  - スイッチの利用によって困難に
    - モニターポート付きスイッチ
  - ネットワークの高速化で大量のデータが発生
  - tcpdumpの利用スキルはつけておいたほうがよい

監視通知の方法

- グラフィカル
- メール
- ログの解析
- レポートの自動化
- ページャ
- 携帯電話 (i-modeで遠隔監視も可能な時代)
- 音

ツールとシステム

- 診断ツール
- モニターツール
- コンピュータ利用型管理システム
- オペレーション（よりの）ツール

何故フリーソフトを使用するか

- 小規模なLANでも、手軽にネットワーク監視
- 実際に管理している人のノウハウが反映されている
- ソースに手を加えることが可能
  - **商用NMSですぐに対応できないところに適応する**
- ソースコードと教育的価値
- PCの低価格化とPC-UNIXの普及
- Webによるビジュアライズ化が充実

フリーのネットワーク管理ツールの傾向

- XベースからWebベースへ
  - **Perlベース**
  - **Tcl/Tkベース**
  - **Webベース**
- MRTGが流行？
- 老舗Expect,Scotty
  - **機能が豊富で強力な分、敷居が高い慣れるのに時間がかかるかも**
- 商用NMSとの違い
  - **オートコレクト、VLANなどベンダー実装機能への対応**

ネットワーク監視・設定のためのベース

- Ping
- traceroute
- nslookup
- ifconfig
- arp
- netstat
- tcpdump(snoop)
- route
- telnet port番号
- SNMP tools

開発・自動化

- Shell(awk,sedなど)、cron
- スクリプト言語
  - Perl
  - tcl/tk
- Expect

統合化ツール

- BigBrother *
- AngelNetwork *
- Mon *
- spon
- MRTG *
- gxsnmp
- Scotty
- RRD
- NOCOL

註:*印は今回説明するツール

Expect

- http://expect.nist.gov/
- telnet, ftp, passwd, fsck, rlogin, tip, etc ..といったインタラクティヴなアプリケーションを自動化するのに便利
  - **イベント発生後の調査、処理の自動化ツール作成には、依然強力なツール**
- Tclベースだが、Perlモジュールも登場
- 本「Exploring Expect」：A Tcl-Based Toolkit for Automating Interactive Programs (Nutshell Handbook) - by Don Libes, (December 1994) O'Reilly & Associates; ISBN: 1565920902

Scotty

- http://wwwhome.cs.utwente.nl/~schoenw/scotty/
- Tcl(Tool Command Language)ベースの管理ツール
- 2つのコンポーネント
  - Tnm ：ネットワーク管理情報へのアクセスするソフト
  - Tkined：ネットワーク管理システムフレームワークを含んだエディタ
- 本「Building Network Management Tools with Tcl/Tk」 -- Dave Zeltserman, Gerard Puoplo:(April 15, 1998) (April 15, 1998) Prentice Hall; ISBN: 0130807273

nocol

ネットワーク実オペレーション向きツール
nocolの各プログラムを取り出して利用も可能

- ICMP ping
- RPC portmapper
- OSI ping
- Ethernet load
- TCP ports
- Nameserver
- Radius server
- Syslog messages
- Mailq
- NTP
- UPS (APC) battery
- Unix host perf
- BGP peers
- SNMP variables
- Data throughput

セキュリティツールーIDSの流用

- Network Fright Recorder （NFR）
  - Network Fright Recorder, Inc.（*http://www.nfr.net/*）による侵入検知システム
  - ソースコードがフリー
  - BSD/OS、HP-UX、Linux などで動作する。
  - ネットワークベースの情報源が利用され、パケットの収集と解析が実施される
  - Java 対応のWebブラウザによって収集データを参照することも可能

SLA達成的ネットワーク監視

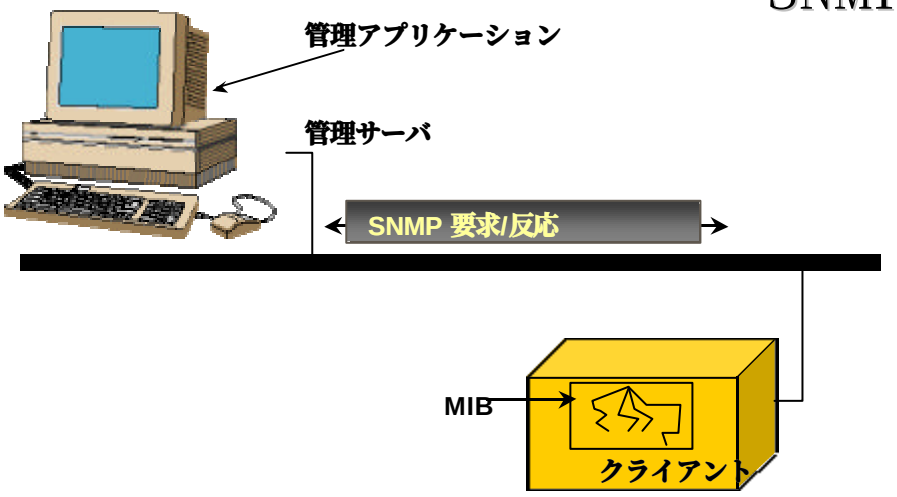
- A地点からB点までの
  - ping パケットロス
  - RTT(Round Trip Time)
  - NTP(Network Time Protcol)
  - FAQ
    - できるだけサーバへ確認する
      - SLA用にレスポンスを返すだけのサーバ(PC)を用意するとベター
    - ルータへの確認は一歩間違うとDoSアタックになる
      - LongPacket-->アッパー
      - ShortPacket-->ボディブロー

SNMP

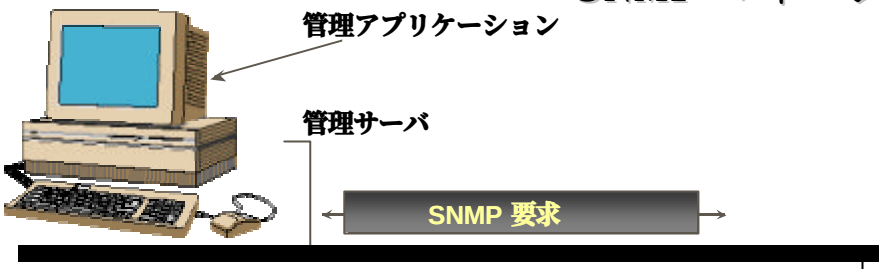
(Simple Network Management Protocol)

- マネージャ（クライアント）／エージェント（サーバ）型プロトコル
- 要求／応答型プロコトル
  - UDP　161
  - PDU(Protocol Data Units)

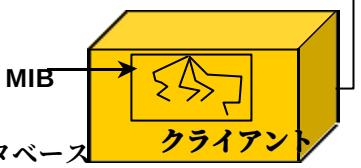
SNMP



SNMP マネージャ



- 管理アプリケーション
- データベース
  - MIB データベース
  - ネットワーク・エレメント・データベース
  - 管理アプリケーションデータベース
    - トポロジデータベース、履歴ログ、モニターログ



SNMPプロトコル

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

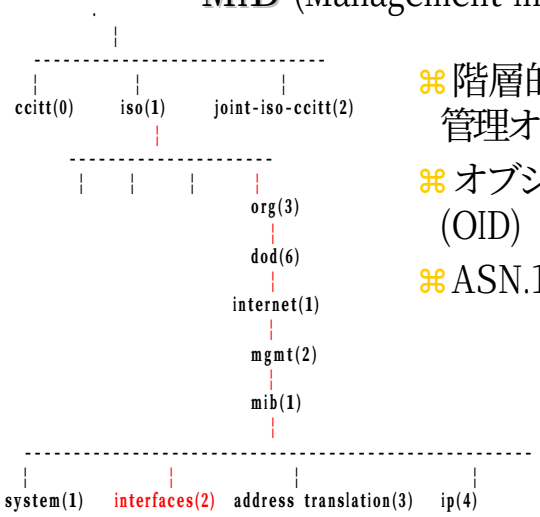
SNMPによる情報収集

- マネージメントステーションを使ってポーリングする
- ネットワーク全体の状況を把握するのみ
  - **どのような通信が行われているかはわからない**
- 転送されているデータ量だけはわかる
  - **インターフェースごとの情報**
- UDPをつかう
- エージェントから情報が確実にくるとは限らない
  - **ネットワーク状況とSNMP搭載機器の実装と負荷状況による**

trap direct polling

- trap
  - **Polingを待たずにSNMPエージェントからマネージャに情報を送る**
  - **トラップを利用してエージェントがマネージャに異常イベントの発生を知らせる**
  - **ポーリングの制御権は、マネージャが保持**
  - **SNMPのトラップ情報はUDPポート162に送られる**

MIB (Management Information Base)



- 階層的な命名体系で管理オブジェクトを定義
- オブジェクト識別子 (OID)
- ASN.1をもとにした定義

トラフィック測定MIB

- RFC2063,2064,2123
  - トラフィック・フロー測定とMeterMIB
  - NeTraMet
  - NeMac

第1部まとめ

- ネットワークの把握が基本
  - 規模、ユーザの挙動
- 効果的なポイントを監視する
- コネクションとトラフィックの把握
- フリーのツールで状況を把握し、再設計や拡張の指針をつくる

第２部：フリーソフトによる ネットワーク監視

矢萩 茂樹
インテリジェント・テレコム株式会社
1999/12/15

話の進行について

- ネットワーク機器の監視方法
- 監視する手段
- 監視システムへの要件
- 監視ソフトの分類と紹介
  - 状態監視ツール
  - 状態検知ツール
  - トラフィック確認ツール
- 今後の話

ネットワーク機器の監視方法1

- 個別監視
  - Client Base
- 集中監視
  - Server Base
  - Server-Client Base

ネットワーク機器の監視方法2

- 個別監視 - Client Base
  - 監視対象で個別に状態監視を行い、問題があった際にアラートをあげる。
    - 運用者が個別にサーバーをみて回る。。。
  - 他の要因に影響を受けないため、より詳細な情報を採りやすい。
  - が、遠隔地に設置している装置の監視はできない。

ネットワーク機器の監視方法3

- 集中監視 - 共通
  - 監視サーバーを立ち上げ、外部より監視を行う。
  - 相手が完全に切り離されてしまった場合でも検知可能
    - 突然のシステムダウンなどにも対応できる。
  - ルーター・スイッチはそれ自体でユーザに対して直接アラーム通知することはないため、監視サーバーによる方法でしか効果的な監視はできない。
  - 集中監視することができるので、少ない人数で多くの装置を面倒みることが可能。

ネットワーク機器の監視方法4

- 集中監視1 - Server Base
  - 監視対象に手を加えず、サービスポートをpollingにより監視する。
    - ICMP, dns, smtp, pop3, http, ...
  - 監視サーバーを個別に立ち上げるだけで、監視可能。
    - 簡単！

ネットワーク機器の監視方法5

- 集中監視2 - Server-Client base
  - 監視対象に詳細情報を収集するためのprobe programを設置。serverがprobeから情報を収集する。
    - SNMPでの管理も同類。
  - インストールが面倒であるが詳細情報も取得できるので、高度な管理ができる。
  - 各プラットフォーム毎に適応したprobeが用意されていないと、対応できない。
    - Windows9X/NT, NetWare, MacOS,...

監視する手段 - Polling 1

- ICMP Polling
  - ICMP echoによる監視。必要最低限どんなノードでもサポートしているために、最低限の監視に使用可能。
- TCP Port Polling
  - 各サービスポートを直接監視する方法。実際に稼動しているかを直接判断できるので、サーバープロセスの監視には有効。

監視する手段 - Polling 2

- SNMP Polling
  - 標準プロトコルベースでの管理方法。
  - 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。
  - サーバーでは個別にSNMP daemonを追加する必要がある場合が多い。
  - 商用製品が多い。
    - 例：HP OpenView

監視する手段 - Event Trap

- Local Event Trap
  - システム状態監視(CPU,disk,process)
  - Process個別処理状況監視
  - Log file監視
- Remote Event Trap
  - syslog によるメッセージ伝達
  - SNMP trap

ユーザへの通知方法

- アラームメッセージ
- 画面・アイコンの点滅
- アラーム音
- Pager Call !
  - これからはi-Modeなどの携帯にe-mail
  - しかしネットワーク的に完全に孤立したら……
    - 専用線接続ISPのAPや他のISPにダイアルアップ。
    - ダイアルアップで障害通知メールを送信。

監視における切り口 - 現状と経過

- 現状監視 - 今を見る
  - 現在のシステム状態を監視する。
- 現状検知 - 今を検知する
  - システムからの自律アラームをとらえる
- 経過監視 - これまでを見る
  - トラフィックの推移を監視。トレンドを把握する。

- これらは密接にからんだ独立事象
  - どれがぬけても片手落ち

監視システムへの要件1

- 集中監視・一斉通知
  - 人は分散できない・できるだけしたくない。
  - 監視は１カ所で。通知はそこから一斉に。
- 監視画面は各自の手元で
  - …でも、自分の机で自分のPCでみたい。

監視システムへの要件2

- 出先でも状態確認
  - リモートで対応するために遠隔で情報取得したい。
  - 他の人のPCでも確認できるようになっていないと…
    - だれでも持っているソフト → Web Browser
    - 専用クライアントはいざというときには使い物にならない。

監視システムへの要件3 - 結論

- WEB画面でリモート監視・リモート確認
- E-mailで通知。携帯へPager Call !
- WEB画面でトラフィック監視

フリーソフトで作る監視システム

- 全てを一つで満足することはできない。
  - 満足するものに作り上げるための努力は無視できない
  - なんでも(そこそこに)できるは何もできないの法則
- なら、適材適所の組み合わせで簡単に作る！
- 監視システムを構成する３つのアイテム
  - 状態監視ツール
  - 状態検知ツール
  - トラフィック監視ツール
- システムへの統合はWEBで

監視の切り口と使えるツール

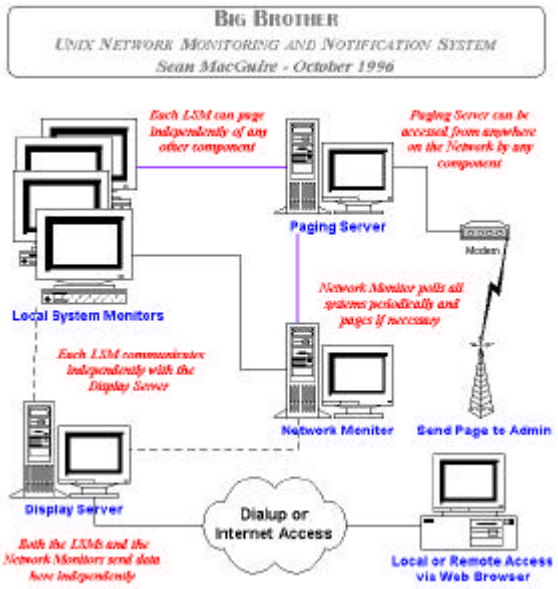
- 今を見る → Polling Base状態監視
  - Big Brother, NOCOL, SPONG, mon, Angel, NetSaint, Scotty,...
- 今を検知する → Trap Base状態検知
  - Syslog + Swatch
  - Snmptrapd(UCD, CMU),...
- これまでを見る → トラフィック監視
  - MRTG, PyNG, RRDTools+(Remstat,Cricket,ORCA,NRG), ...

Polling base 状態監視ツール

- Big Brother
- SPONG
- Angel
- NOCOL
- mon

状態監視ツール - Big Brother

- http://maclawren.ca/bb-dnld/
- WEB Baseの監視システム。
- Server-Clientタイプ
- 監視・表示・通知機能を分割しており分散監視管理可能
- ICMP/TCPベースの監視を行う。
  - 監視可能サービス：ping, smtp, http, pop3,dns,ftp.telnet, ssh,...
  - サーバー個別監視：CPU, disk, processes, logs, ….
- NT/NetWare用の監視クライアントがあり、統合監視可能



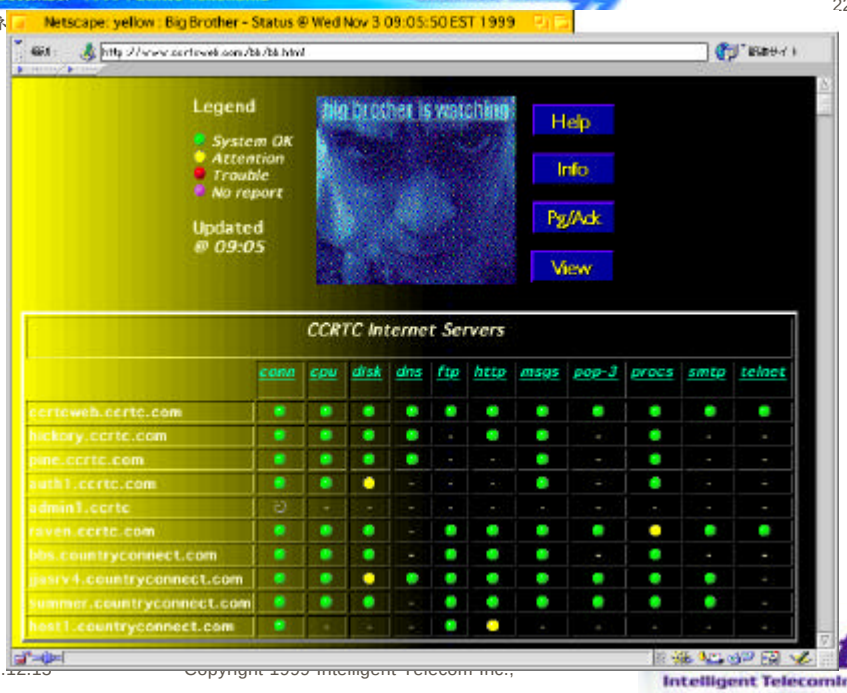
Big Brother
サーバー構成

状態監視ツール - Big Brother 続き

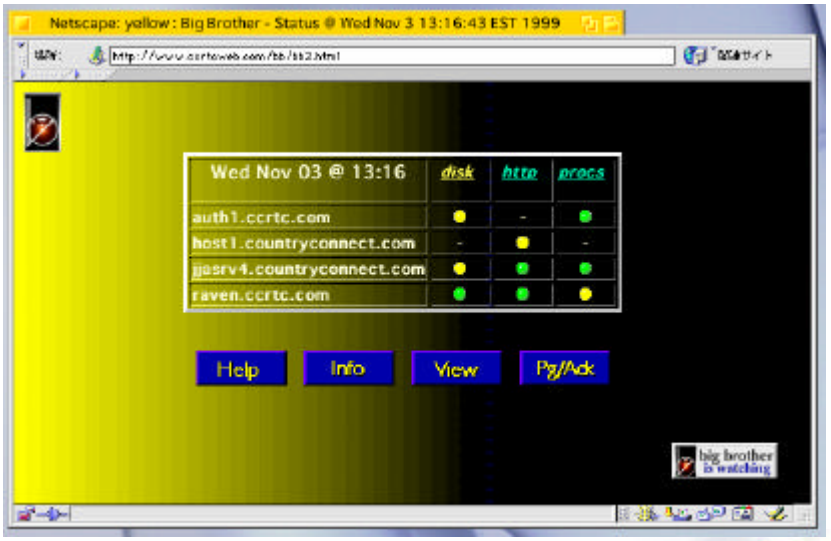
- 監視対象のグループ化機能
- ホスト単位にシステムの停止時間を設定。自動で監視対象から除外可能。
- ホスト単位で障害通知先を変更可能
- 簡単な障害履歴機能を持つ
- アラームの検出されている機器のみ抽出した画面を標準で生成。→ 便利！

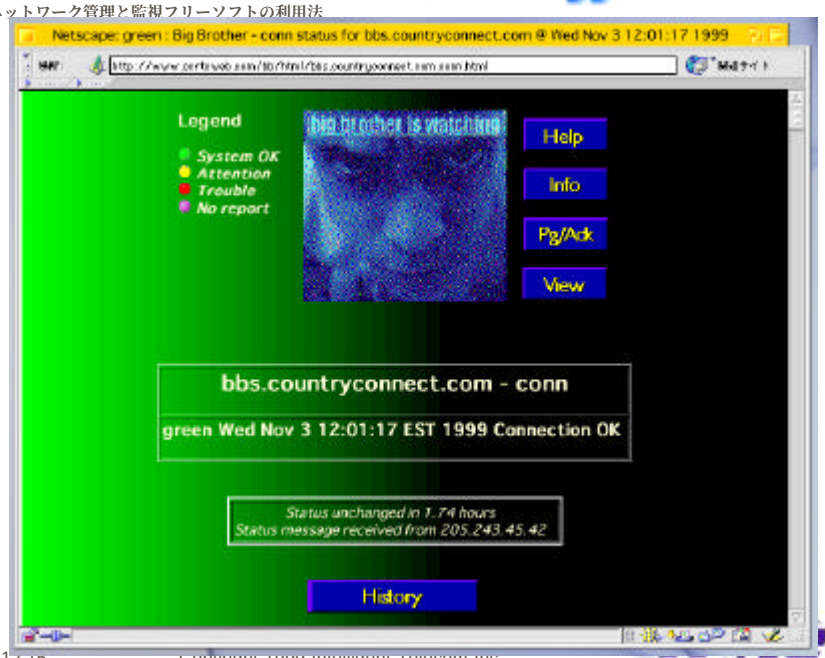
- 非常に簡単！

IT Net Intelligent Telecominc.

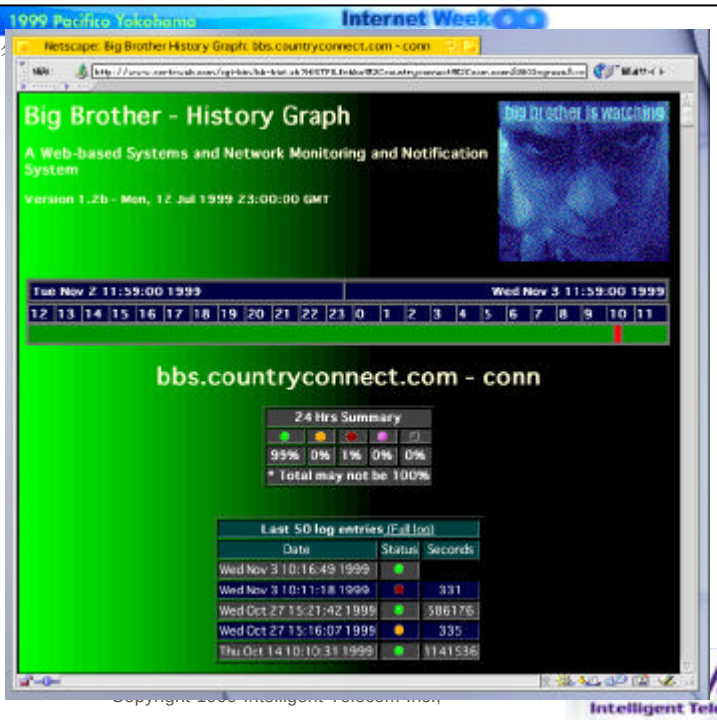


IT Net Intelligent Telecominc.

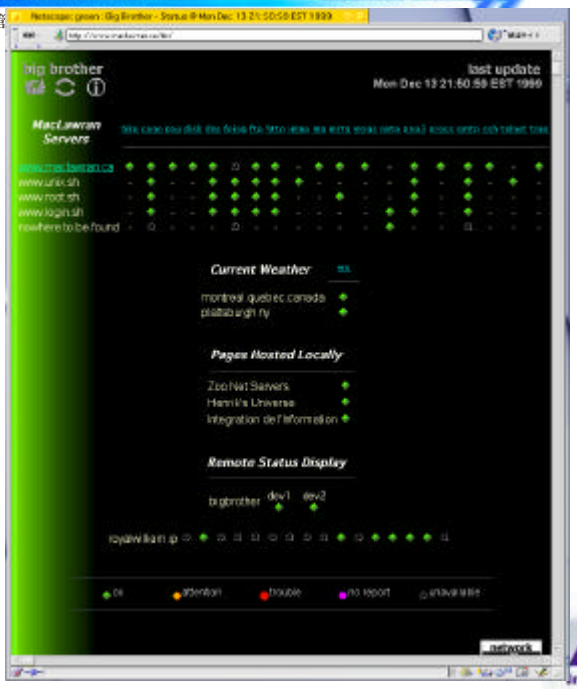


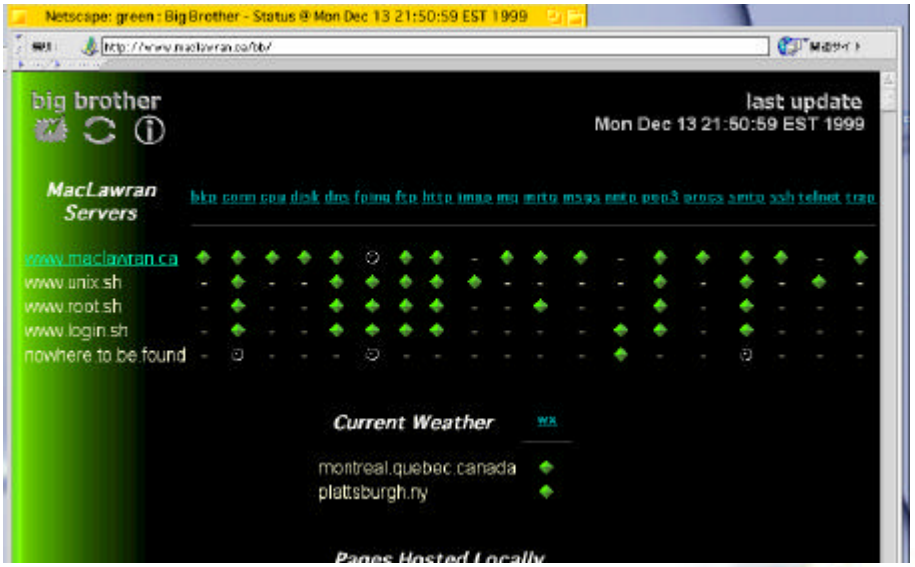


C10 ： ネットワーク



C10 ： ネットワーク管





Big Brotherの設定 - bb-hosts

```
$ cat bb-hosts
#
# THE BIG BROTHER HOSTS FILE
#
192.168.0.10 kansi1.foo.co.jp # BBPAGER BBNET BBDISPLAY http://kansi1/

group-compress <H3><I>foo.co.jp Servers</I></H3>
192.168.0.2  ns1.foo.co.jp  # dns ssh
192.168.0.3  mail.foo.co.jp # dns smtp pop3 ssh
192.168.0.5  www.foo.co.jp  # telnet ssh ftp http://www.foo.co.jp/

# router interface entry
group-compress <H3><I>Router Interface</I></H3>
192.168.0.1  gw1.foo.co.jp
192.168.0.50 gw2.foo.co.jp
192.168.1.2  tok-yok-ma30.wan.foo.co.jp
192.168.1.6  tok-osa-dr15.wan.foo.co.jp
$
```

Big Brotherの設定 - bbwarnrule

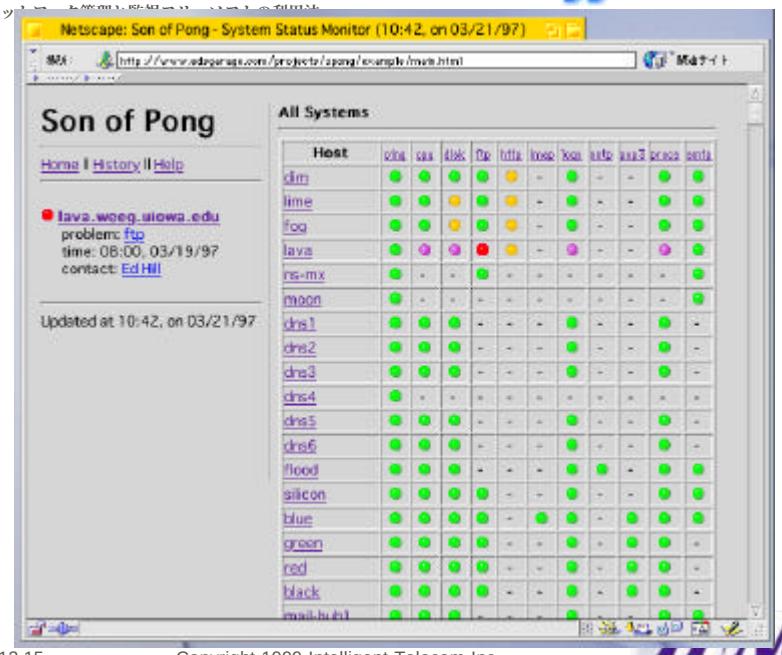
```
$ cat bbwarnrules.cfg
# bbwarnrules.cfg
#
# Rules are written in the following format:
#   hosts;exhosts;services;exservices;day;time;recipients
# hosts: match on these hosts (* is a wildcard for all hosts)
# exhosts: exclude these hosts
# services: match on these services (* is wildcard for all hosts)
# exservices: exclude these services
# day: 0-6 (sunday-saturday)
# time: 0000-2359
# recipients: email address, numeric pager, sms number
ns1.* mail.*;;*;;*;*;server-admin@foo.co.jp
www.*;;http;;*;*;server-admin@foo.co.jp @foo.co.jp
*;;*;;*;*;admin@foo.co.jp
unmatched-*;;*;;*;*;root@localhost
$
```

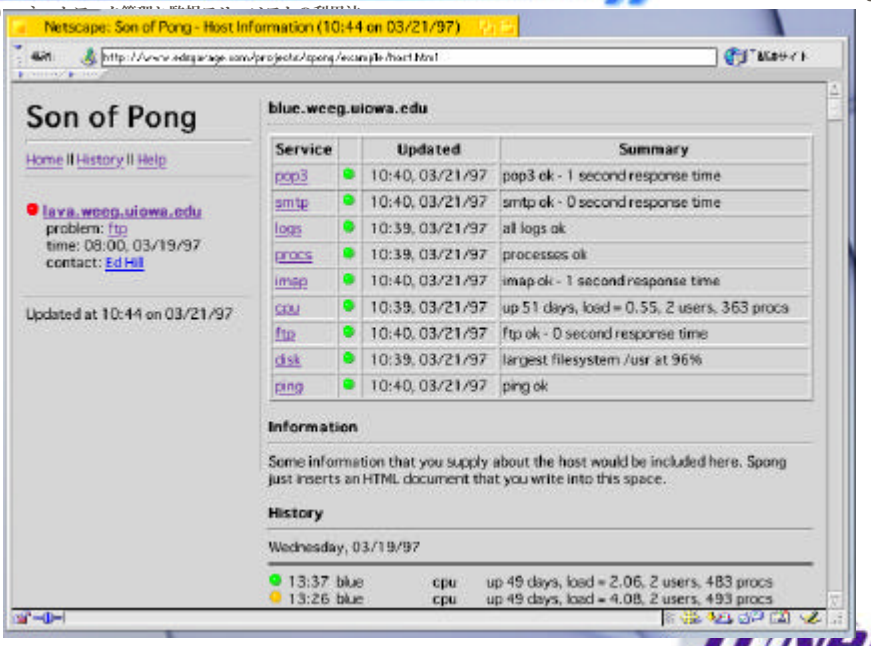

状態監視ツール - SPONG

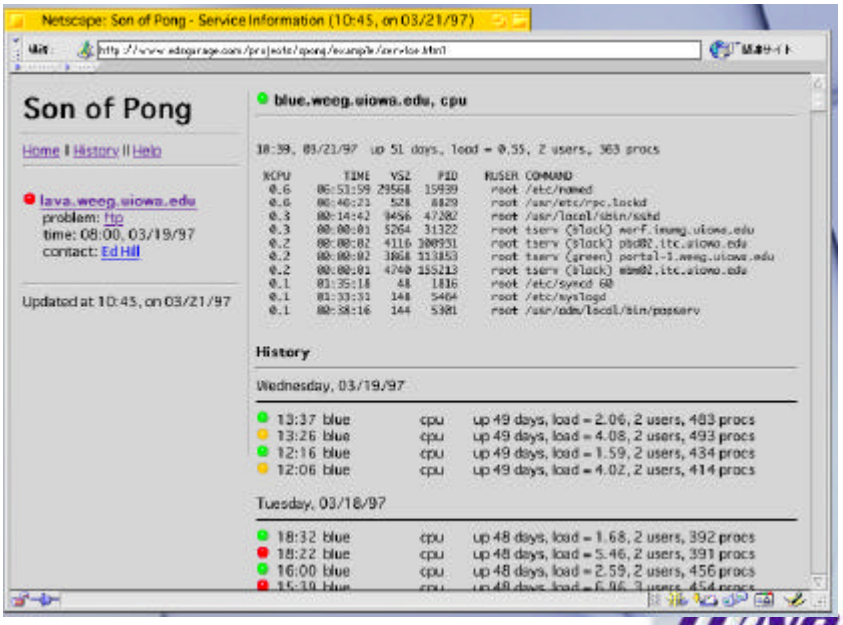
- http://www.edsgarage.com/projects/spong/
- WEB Baseの監視システム。システム監視／障害ログ解析機能に特化。簡単な設定で使える
- Server-Clientタイプ。
- Big Brotherをベースに開発。独自に発展。
  - 開発が止まっていたが、新たな開発者の元で開発再開！
- ICMP/TCPベースの監視を行う。
  - 監視可能サービス：smtp, http, ping, pop,dns,ftp.telnet, ...
  - サーバー個別監視：CPU, disk, processes, logs, ….

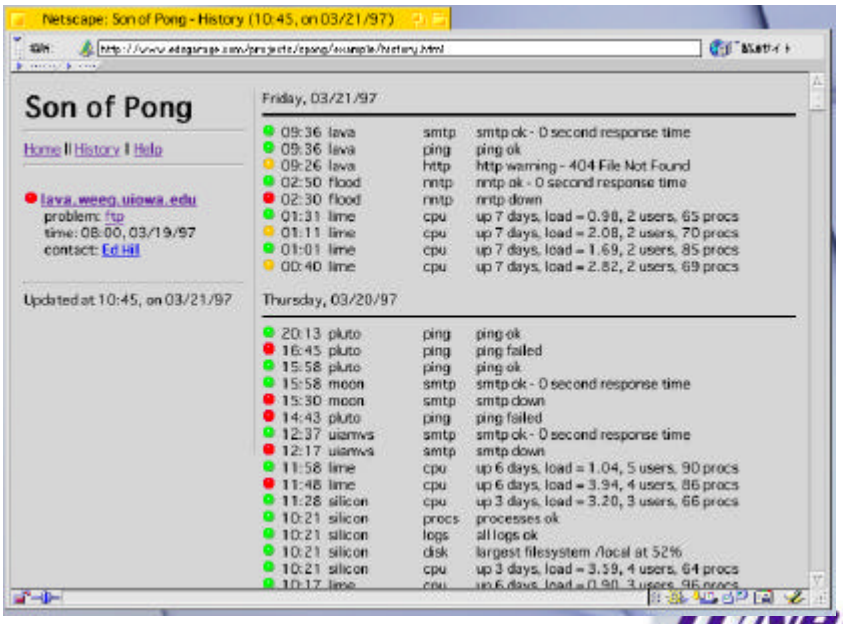
状態監視ツール - SPONG 続き

- 監視対象のグループ化機能
- ホスト単位にシステムの停止時間を設定。自動で監視対象から除外可能。
- ホスト単位で障害通知先を変更可能
- 障害ログ管理がしっかりしており、多面的に障害ログを表示することが可能。
  - 全体
  - ホスト単位
  - サービス単位
  - ....









SPONGの設定 - spong.hosts

```
%HUMANS = (
 'admin'    =>  { name  => 'Engineer', email => 'admin@foo.co.jp' },
 'tomoharu' =>  { name  => 'TOMOHARU', email => 'tomoharu@foo.co.jp' },
 'yahagi'   =>  { name  => 'YAHAGI',   email => 'yahagi@foo.co.jp' },
);
%HOSTS = (
 'kanshi.foo.co.jp' => { services => 'pop smtp',
       contact  => 'admin', group => 'unix' },
 'ns.foo.co.jp'     => { services => 'pop telnet',
       contact  => 'admin', group => 'unix' },
 'regist.foo.co.jp' => { services => 'dns telnet',
       contact  => 'admin', group => 'unix',
       down     => [ '*:04:00-05:00' ] },
 'www.foo.co.jp'    => { services => 'ftp telnet http',
       contact  => 'tomoharu', group => 'unix' },
 'mail.foo.co.jp'   => { services => 'pop smtp telnet',
       contact  => 'admin', group => 'unix' },
 'gw1.foo.co.jp'    => { services => '',
       contact  => 'yahagi', group => 'router' },
 'gw2.foo.co.jp'    => { services => '',
       contact  => 'yahagi', group => 'router' },
);
```


SPONGの設定 - spong.group

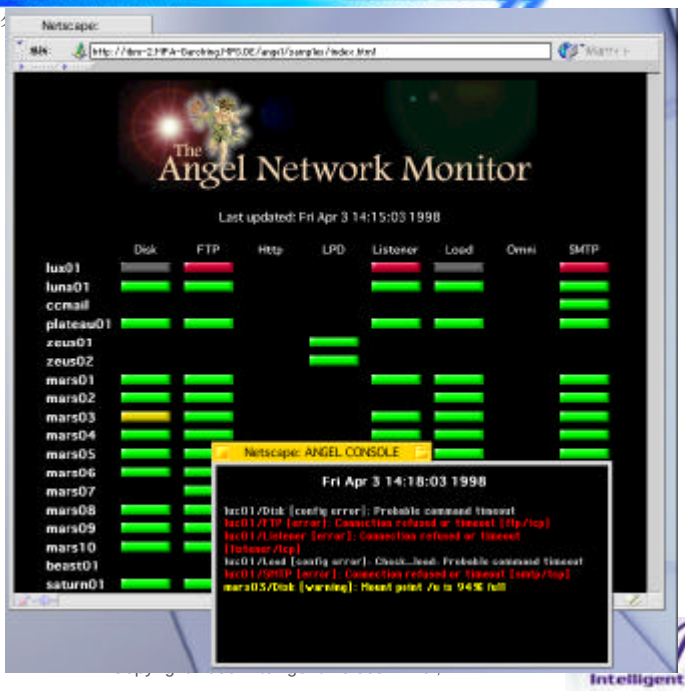
```
%GROUPS = (
  “all”     => { name   => “All Systems”,
                 summary => “all hosts monitored by spong" },

  “unix”    => { name    => "Unix - All",
                 summary => "All Unix Systems and Servers",
                 members => [ “kanshi.foo.co.jp”,
                              “ns.foo.co.jp”,
                              “regist.foo.co.jp”,
                              “www.foo.co.jp”,
                              “mail.foo.co.jp” ] },

  “router”  => { name    => “ROUTER”,
                 summary => “router group”,
                 members => [ “gw1.foo.co.jp”,
                              “gw2.foo.co.jp” ] }
);
```


状態監視ツール - Angel

- http://www.ism.com.br/~paganini/angel
- WEB Baseの監視ツール。
- Serverタイプ
- 簡単な設定で使え、画面がきれい。
- ICMP/TCPベースの監視を行う。
  - 監視可能サービス：smtp, http, ping, pop,nntp,dns, ...
  - サーバー個別監視：CPU, disk, ...
- Perlで記述。監視機能モジュール形式となっており以下のものが使用可能。
  - Check_tcp, Check_ping, Check_load, Check_disk
- httpは個別URLを指定可能



Angelの設定 - hosts.conf

```
$ cat hosts.conf
#
# Check server
#
ns:Check_ping:ns.foo.co.jp!100!200!5!15:PING:alertred!alertyellow!alertblack

mail:Check_ping:mail.foo.co.jp!100!200!5!15:PING:alertred!alertyellow!alertblack
mail:Check_tcp:mail.foo.co.jp!80:smtp:alertred!alertyellow!alertblack

www:Check_ping:www.foo.co.jp!100!200!5!15:PING:alertred!alertyellow!alertblack
www:Check_tcp:www.foo.co.jp!80:ftp:alertred!alertyellow!alertblack
www:Check_tcp:www.foo.co.jp!80:Http:alertred!alertyellow!alertblack

#
# Check the gateway routers
#
gw1:Check_ping:192.168.0.1!100!200!5!15:PING:alertred!alertyellow!alertblack
gw2:Check_ping:192.168.0.100!100!200!5!15:PING:alertred!alertyellow!alertblack

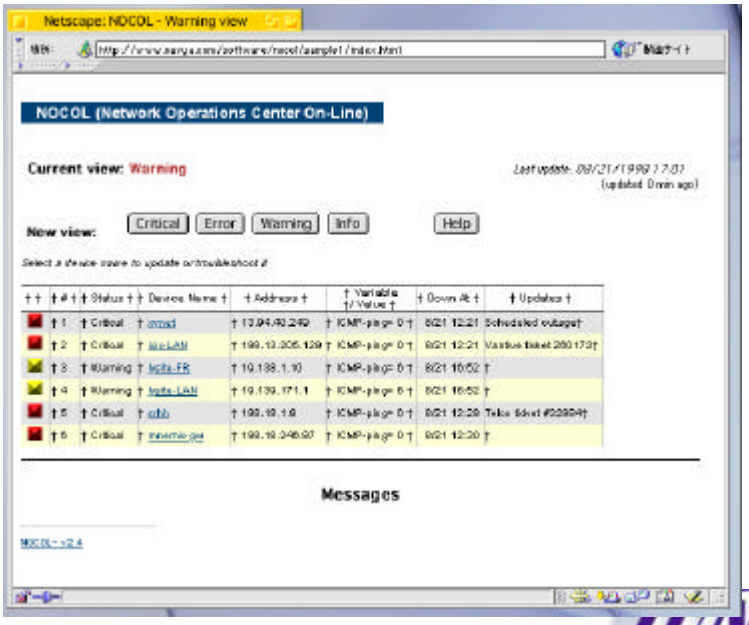
$
```

状態監視ツール - NOCOL

- http://www.netplex-tech.com/software/nocol
- WEB Baseの高機能監視システム。
- ICMP/TCPポーリングの他にSNMPベースでの監視をサポート。
  - CMU-SNMP Packageの拡張版
- 単体システム／サービスにとどまらず、モデム、UPS、はてはネットワークのスループットまで監視可能
- ネットワーク機器に直接telnet loginし、データを自動取得。内容監視することも可能。
  - expect的auto-pilot機能

状態監視ツール - NOCOL 続き

- 監視可能項目
  - Ping, Ethernet load, radius, ntp, bgp peer, rpc portmapper, tcp ports, syslog mesg, ups battery, snmp variables, OSI ping, dns, mailq, unix host perf, data throughput, ….
- 同一の対象に対して複数のアラーム通知レベルを設定でき、別々に通知することが可能。
- 設定可能な項目がとにかく多い。
  - 挑戦しがい十分。

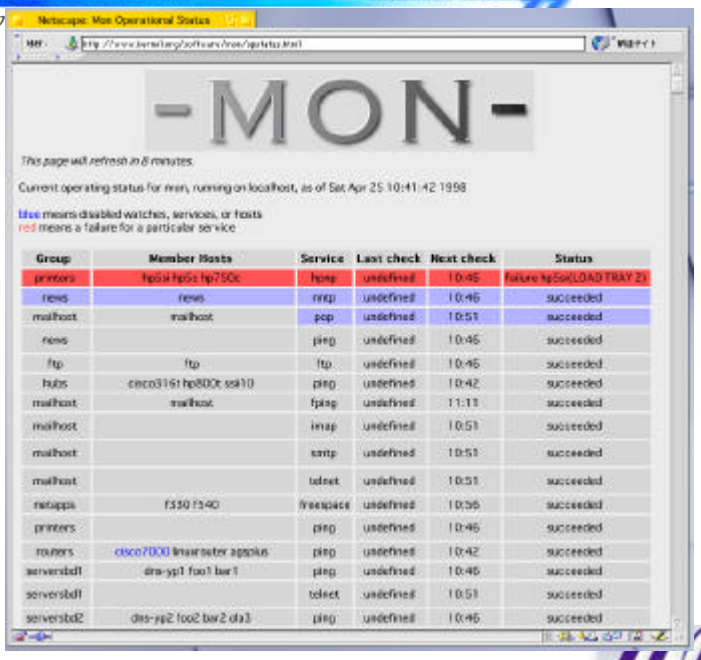


NOCOLの設定ファイル

- apcmon-confg
- armon-confg
- bgpmon-confg
- bpmon-confg
- etherload-confg
- hostmon-confg
- ippingmon-confg
- modemmon-confg
- noclogd-confg
- notifier-confg
- novellmon-confg
- nsmon-confg
- ntpmon-confg
- pm3dmmon-confg
- pm3t1e1mon-confg
- portmon-confg
- radiusmon-confg
- rpcpingmon-confg
- snmpmon-client-confg
- snmpmon-confg
- syslogmon-confg
- tpmon-confg

状態監視ツール - mon

- http://www.kernel.org/software/mon
- WEB Baseの高機能監視システム。
- ICMP/TCPポーリングの他にSNMPベースでの監視をサポート。(UCD-SNMP Package Base)
- 監視可能項目
  - **Ping, SMTP, telnet, ftp, nntp, http, pop3m imap, tcp ports, disk space, snmp variables, ldap, dns, modems**
- 監視はかなり細かくカスタマイズできるが、通知機能は簡単になっている
  - Big Brother/SPONGレベル。



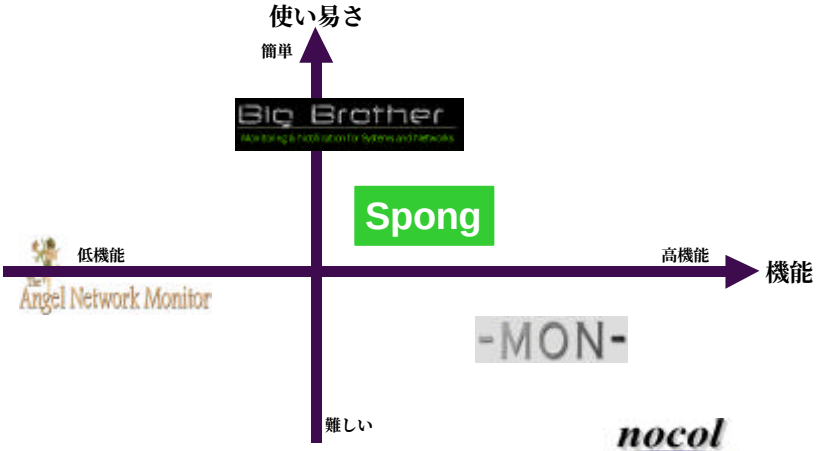
mon - mon.cfの例

```
#
alertdir  = /usr/lib/mon/alert.d
mondir    = /usr/lib/mon/mon.d
maxprocs  = 20
histlength = 100
randstart = 60s
#
# define groups of hosts to monitor
#
hostgroup servers localhost

hostgroup mail mailhost

watch servers
        service ping
                interval 5m
                monitor fping.monitor
                period wd {Mon-Fri} hr {7am-10pm}
                        alert mail.alert root@localhost
                        alertevery 1h
                period wd {Sat-Sun}
                        alert mail.alert root@localhost
        service telnet
                interval 10m
                monitor telnet.monitor
                period wd {Mon-Fri} hr {7am-10pm}
                        alertevery 1h
                        alertafter 2 30m
                        alert mail.alert root@localhost
```


状態監視ツールのセグメント分類



- 機能（高機能ー低機能、監視範囲）
- とっつき易さ（使いやすい、設定簡単）

Trap base 状態検知ツール

- Syslog+SWATCH
  - Swatch: the simple WATCH dog
- snmptrapd
  - UCD-SNMP Package
  - CMU-SNMP Package

SWATCH

- http://www.engr.ucsb.edu/~eta/swatch/
- ログファイルに対してパターンマッチングによる監視を行う
- 適合イベントが発生した際にアラームをあげる・外部コマンドを起動するなどのアクションを実行
- セキュリティー向上のために使われることが多いが、システム・サーバープロセスの出力するメッセージを監視することで、定常運用のためのシステム監視ツールとして適用可能。
- syslogを取りまとめるloghostにて実行すると効果的

SWATCHのconfigと実行例

```
$ cat ~/swatchrc

# Swatch configuration file for constant monitoring
# Bad login attempts
watchfor = /LOGIN FAILURES/
      echo
      bell=3
      exec="finger $10 | mail -s ¥"LOGIN-FAILURE:$10¥" admin@foo.co.jp"

# System crashes and halts and reboots
watchfor = /panic|halt/
      echo
      bell
#     mail=admin@foo.co.jp:yahagi@foo.co.jp
      exec="echo $0 | mail -s SYSTEM-HALT pager@foo.co.jp"

$
$ swatch --config-file=~/swatchrc --tail-file=/var/log/messages
```


UCD-SNMP Package

- http://ucd-snmp.ucdavis.edu/
- 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

UCD-SNMP snmptrapd

- 外部からの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の例

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

snmptrapd - ciscoでのsnmp関連config例

```
access-list 30 permit 192.168.100.1

snmp-server contact  admin@foo.co.jp
snmp-server location YOKOHAMA-IW99
snmp-server community himitsu RO 30
snmp-server enable traps config
snmp-server host 192.168.100.1 NAISHO tty config envmon snmp
```

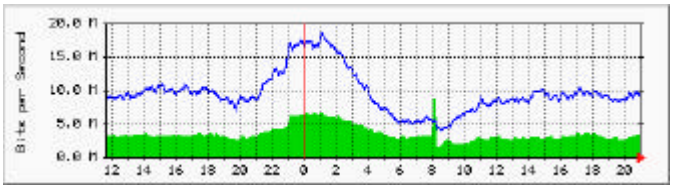

snmptrapd - 通知結果

```
From: log-admin <root@log.foo.co.jp>
To: admin@foo.co.jp
Date: Mon, 1 Nov 1999 22:01:49 +0900 (JST)
Subject: linkDown Trap

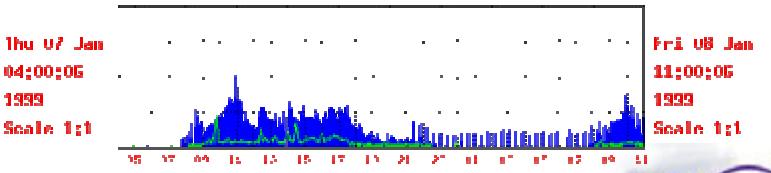
nspixp2-gw.foo.co.jp
192.168.244.21
system.sysUpTime 24:10:03:09.12
.iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTrap.snmpTrapOID .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObjects.snmpTraps.linkDown
interfaces.ifTable.ifEntry.ifIndex.1 1
interfaces.ifTable.ifEntry.ifDescr.1 "Fddi1/0/0"
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.snmpTrapEnterprise enterprises.9.1.48
```

これまでを見る監視ツール

- MRTG (Multi Router Traffic Grapher)



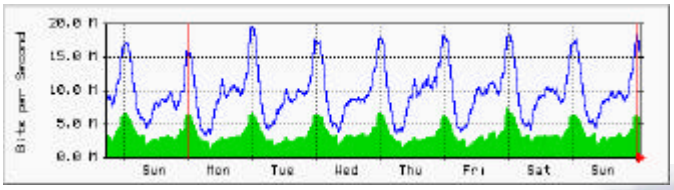
- PyNG (the Python Network Grapher)



IT Net Intelligent Telecominc.

MRTGとは

- http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
- http://www.ceres.dti.ne.jp/~riocat/webtools/mrtg/
  (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。



IT Net Intelligent Telecominc.

MRTGの特徴

- ほとんどのUnixプラットフォームとWindowsNT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。
  - 便利！便利！便利！
- コンフィグからindexを簡単に生成するツールが付属。

MRTG - cfgmaker

- mrtg付属の簡易設定ツール
  - `cfgmaker <community>@<target> > mrtg.cfg`
  - `<community> : snmp community string`
  - `<target> : target address or hostname`
  - 例： `$ cfgmaker himitsu@ix-gw.foo.co.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。

MRTG - cfgmaker の出力結果

```
# Add a WorkDir: /some/path line to this file

######################################################################
# Description: Cisco Internetwork Operating System Software IOS (tm) GS ...
#     Contact: admin@foo.co.jp
# System Name: ix-gw.foo.co.jp
#    Location: PA, CA, US
#.....................................................................…

Target[ix-fddi.foo.co.jp]: 1:himitsu@192.168.98.133
MaxBytes[ix-fddi.foo.co.jp]: 12500000
Title[ix-fddi.foo.co.jp]: ix-gw.foo.co.jp (ix-fddi.foo.co.jp): Fddi1/0/0
PageTop[ix-fddi.foo.co.jp]: <H1>Traffic Analysis for Fddi1/0/0
 </H1>
 <TABLE>
   <TR><TD>System:</TD><TD>ix-gw.foo.co.jp in Otemachi 5F</TD></TR>
   <TR><TD>Maintainer:</TD><TD></TD></TR>
   <TR><TD>Interface:</TD><TD>Fddi1/0 (1)</TD></TR>
   <TR><TD>IP:</TD><TD>ix-fddi.foo.co.jp (172.16.0.2)</TD></TR>
   <TR><TD>Max Speed:</TD>
       <TD>12.5 MBytes/s (fddi)</TD></TR>
  </TABLE>
```


MRTGの使い方

- 独立コマンドとして作成されており、単独では周期起動しない。
  - cronにて定期的に起動する。(default：5分間隔)
  - ```
    # crontab -l
    0,5,10,15,20,25,30,35,40,45,50,55 * * *
      /usr/local/sbin/mrtg /usr/local/etc/ix-foo.cfg
    #
    ```
- データ収集指定はconfigファイルのTargetレコードにて指定。
    ```

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.foo.co.jp`
  - `Target[gw1-err-3]: ifInErrors.3&ifOutErrors.3:himitsu@gw1.foo.co.jp`
  - `Target[gw1-if-1]: -/10.0.0.101:himitsu@gw1.foo.co.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

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.foo.co.jp**
  - gw1.foo.co.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.foo.co.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.foo.co.jp**
  - gw1.foo.co.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.foo.co.jp**
  - 上の例のOID指定

ちょっと脇道 - MIB Group

- **RFC-1213 インターネット標準 MIB-2**
- **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

ちょっと脇道 - よく使う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
  - 1.3.6.1.2.1.2.2.1.10 : 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

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.foo.co.jp
  - gw1.foo.co.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.foo.co.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 addres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.foo.co.jp
  - **gw1.foo.co.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.foo.co.jp
  - 例7の**OID**パターン。

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 -f -c 100 ftp.foo.co.jp
PING ftp.foo.co.jp (192.168.101.238): 56 data bytes
.
--- ftp.foo.co.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
#
```

  - `-f : flood mode (supervisor only option).`
    ネットワークに高負荷を強いることから取り扱い注意
    - Flood modeの挙動はplathomeにより異なるのでやはり注意が必要。今回のスクリプトはFreeBSDで稼動確認している。
    - ＃Linuxは挙動がちょっと変？

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 $0 $*
# /usr/local/bin/check_loss2.sh ftp.foo.co.jp
5
5
0
/usr/local/bin/check_loss.sh ftp.foo.co.jp

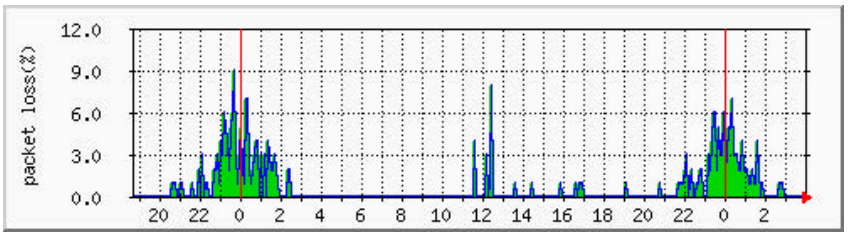
#
```


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.foo.co.jp`
Title[pingloss-ftp]: ftp.foo.co.jp - pingloss
MaxBytes[pingloss-ftp]: 100
PageTop[pingloss-ftp]: <H1> ftp.foo.co.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



MRTGでのTIPS

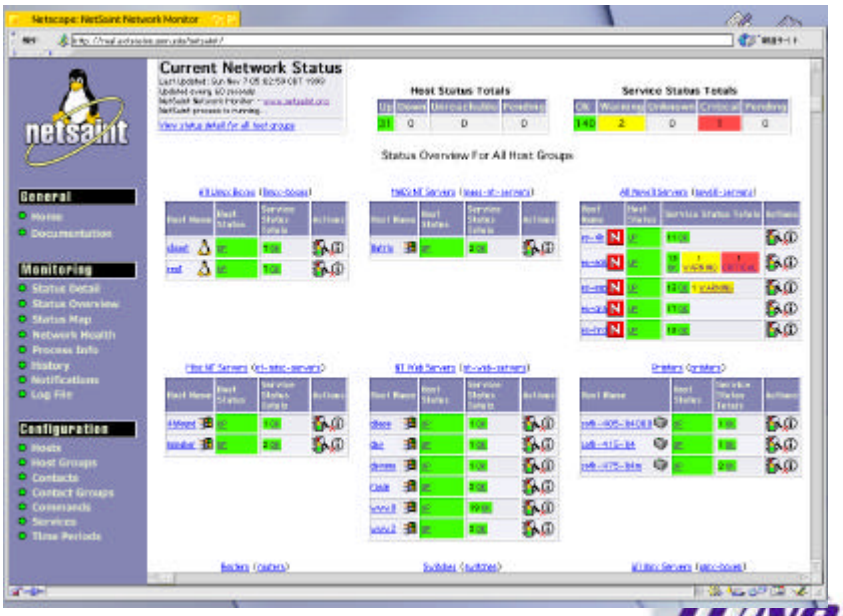
- データの方向性に注意
  - 対抗している装置で同じポートを測定するとIn/Outが逆の結果がでる。
  - 対外線を出口として、ここを起点にデータが流れるように設定すると考えやすい。
- データの単位に注意。
  - `ifInOctets/ifOutOctets`はOctet単位系。
  - 回線・物理接続速度はbps。つまりbit単位系
    - `Options[hoge] bits`した上で`Maxbytes[hoge]`を1/8する。
- Interface address指定を効果的に使う

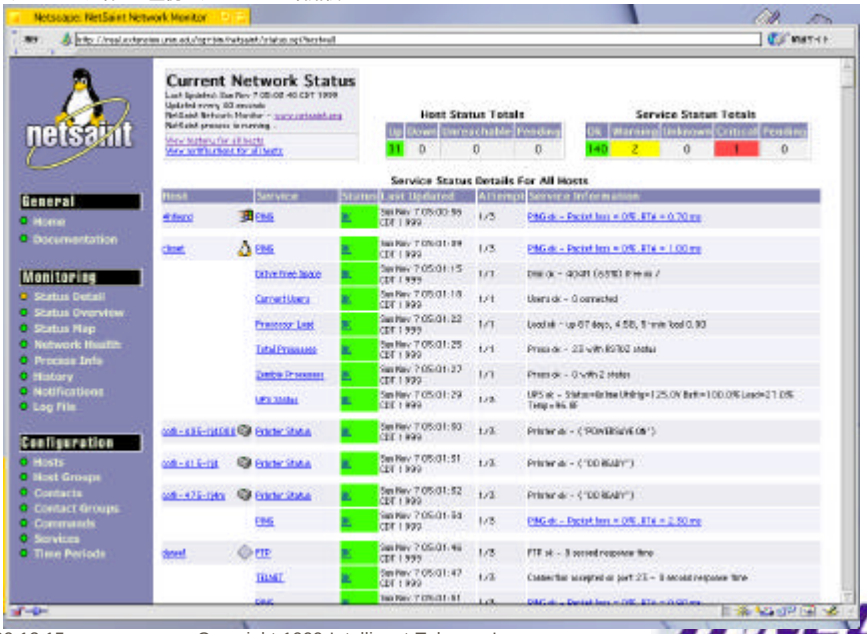
今後の期待

- NetSaint
- RRDTools → MRTG3(?)

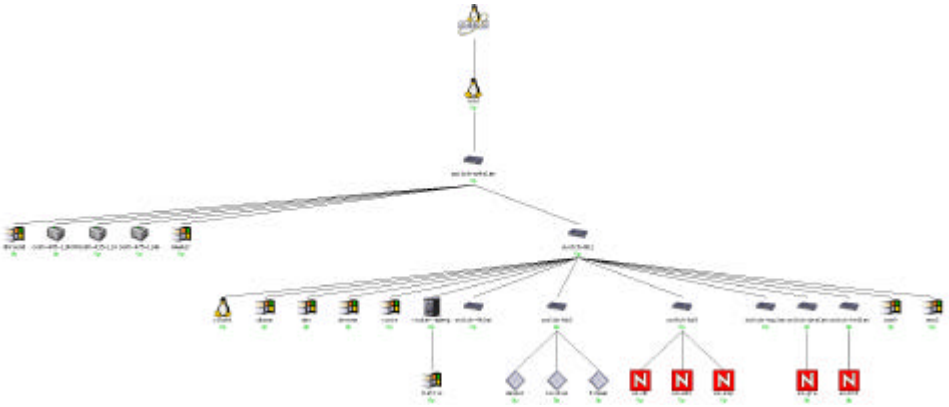
今後の期待 - NetSaint

- http://www.netsaint.org/
- WEB Baseの監視システム。現在、version 0.0.5
- Web Baseでの管理・変更が可能。
- ICMP/TCPベースの監視を行う。
  - 監視可能サービス
    - ping, smtp, http, pop3,dns,ftp.telnet,...
- Plugin形式をとっており、外部拡張可能
  - MRTG plugin
  - remote server management plugin
  - ...





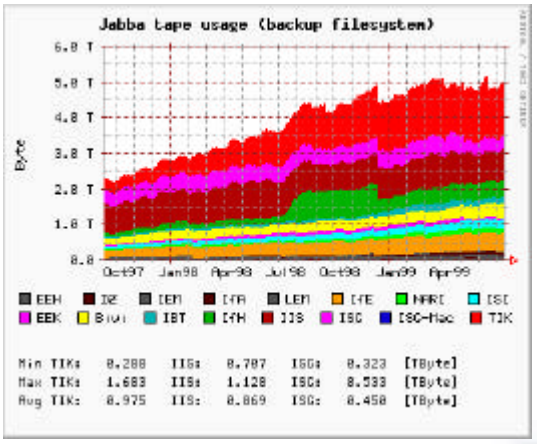
NetSaint - Object Map



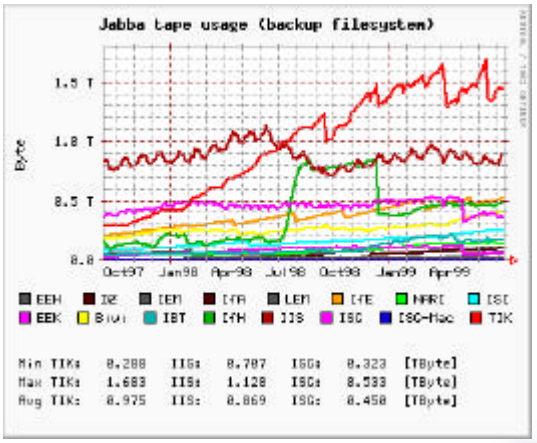
今後の期待 - RRDTools

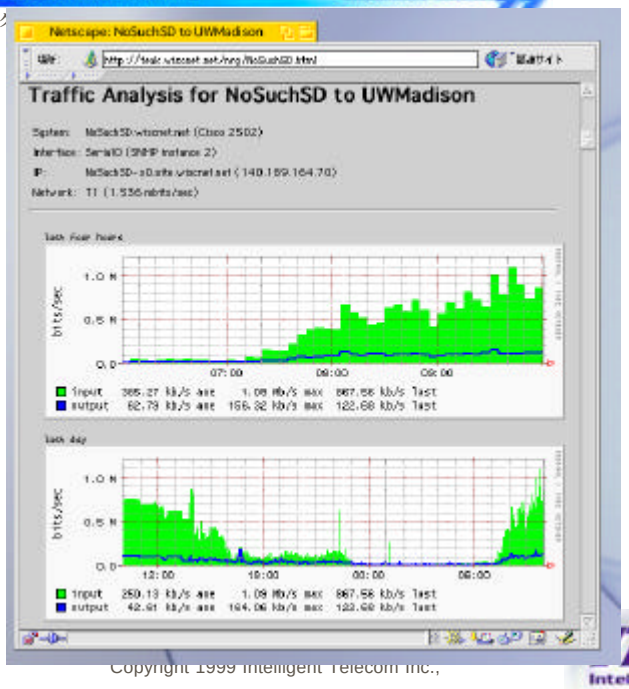
- http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
- RRDTools :Round Robin Database Tools
- MRTGの作者Tobi OetikerによるMRTGの後継プロジェクト。
- MRTGのログサイズが変わらないという利点を受け継ぎつつ、より柔軟に、より高速に、より多彩な表現ができるように、をコンセプトに開発。
- データベース管理、グラフ作成に特化。
  - RRDToolsだけではMRTGのようなWEB画面はできない。
  - FrontEnd Programが必要。
    - Remstat, ORCA, Cricket, NRG, ...

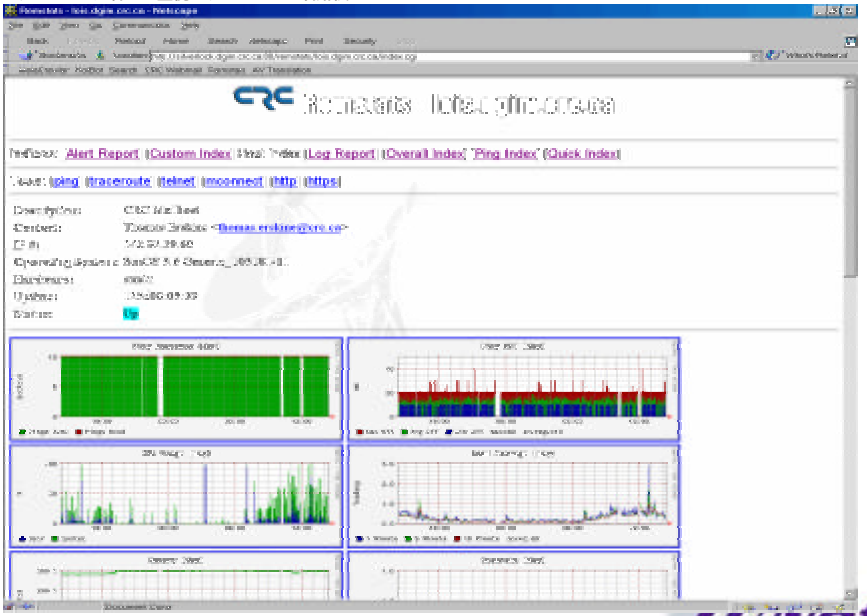
RRDTools - 例1

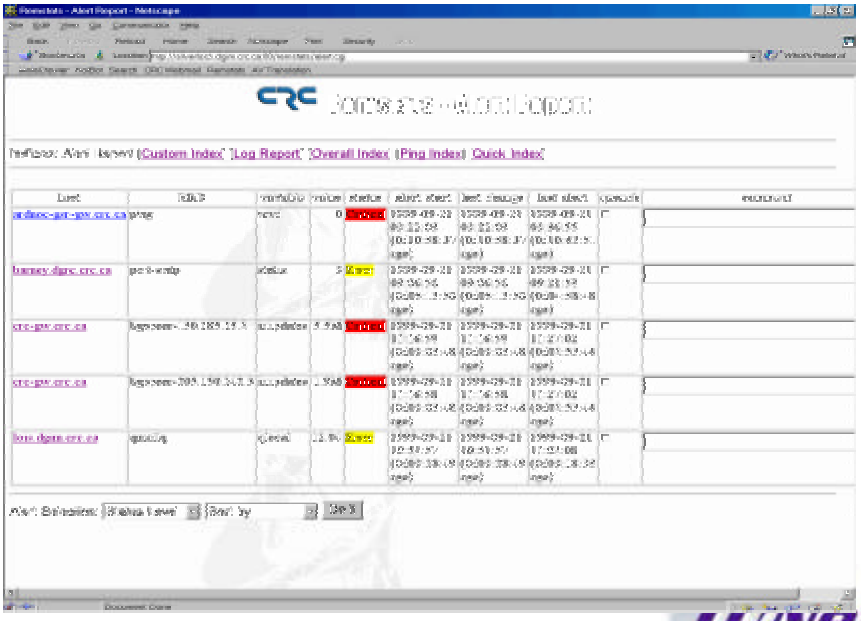


RRDTools - 例2









まとめにかわり

- 今回照会ツールの一部をデモとして公開
- **Ref: http://rouge.itjit.ad.jp:3000/**
- **InternetWeek99期間中のみ**
  - **Always under-construction (^^;**

第3部：ネットワーク管理に関するTIPS...

佐藤 友治　株式会社インターネット総合研究所
矢萩 茂樹　インテリジェント・テレコム株式会社

TIPS - まずは

- ツールの挙動確認をまずオフラインで確認
  - 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/traceroute編

- Pingはほどほどに。
  - ルーターとかスイッチは以外に弱い。
  - Flood Pingに注意。
    - 監視なの? DoS attackなの？
- tracerouteは絶対ではない
  - 行きと帰りは非対称。通るとは限らない
  - ソースルートオプションは絶対ではない
- *＃バージョンと機器毎の使用の確認が必要だが#*
  - *2ndary Addressへのpingは通らないかもしれない(cisco)?*
  - *ciscoはprimary addressをソースに返事をするものがある。*

TIPS - 監視サーバーの置き方とか編1

- 監視サーバの置き場所には注意
  - より詳細な監視をするためにはもっともコアになる装置のそばに置く。
- 監視サーバーの画面は外に公開するものか？
  - .htaccess規制もやっておきましょう。
  - Proxyに注意。
    - 『htaccess』で規制していても、proxyが中にいてopenな状態だと意味がない。
    - ACLでno-cacheにしましょう。
- http portを変更する (http port != 80)

TIPS - 監視サーバーの置き方とか編2

- 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。
  - 適性範囲に分割が必要。
    - 規模拡大時に見落としやすいので注意。

TIPS - 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
  - **商用のNMS運用との兼ね合い、マニュアルの鵜呑み**

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
- Cisco
  - SNMPアクセス規制用access-listの設定
- そんな機能のないホスト
  - 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:SwitchingHub, ...)

TIPS - SNMP編3: Interface高速化に伴う問題

- カウンター **一周問題**
  - ifInOctets/ifOutOctes は32bit正数
  - 5分ごとに各数値を集計する場合、約110Mbpsを越えるトラフィックが生成されるネットワークではカウンターが**一周**する。
  - **測定周期**の調整が必要となる。

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UCD-SNMPコマンド例

- `$ snmpwalk 10.0.0.1 himitsu 1`
- `$ snmpwalk 10.0.0.1 himitsu 2`
- `$ snmpwalk 10.0.0.1 himitsu ifDescr`
- `$ snmpwalk 10.0.0.1 himitsu ifType`
- `$ snmptranslate -n -R ifInDiscards`
- `$ snmptranslate -R ifInDiscards`
- `$ snmptranslate -n -f -d -R ifInDiscards`

TIPS - MRTG1
概論(再び)

- 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が逆の表示になる。
  - 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1/8する。
- Interface address指定を効果的に使う
  - SNMP ifIndex問題。

TIPS - MRTG2
MRTGで計測するCPU Load1

- $ /usr/local/etc/mrtg/cpu.sh

```
#!/bin/csh
top -d 2 | grep 'CPU states' | awk '{
   print $(NF-1)
}' | cut -f1 -d"." | awk '{
   a=100-$1 ; print $1 "¥n" a "¥n"
}'
uptime | awk '{print $3 " " $4; }' | cut -f1 -d","
hostname
$ /usr/local/etc/mrtg/cpu.sh
93
7
6:19
myhost.foo.co.jp
$
```

但しこのshellスクリプトがサーバでどのように処理されるか各自で確認し、よいshellかどうか勉強してみてください

TIPS - MRTG3
MRTGで計測するCPU Load2

- 次のような mrtg.cfg を使ってidle/active CPU を赤のドットラインでグラフ化。

```
Target[cpu]:     `/usr/local/etc/mrtg/cpu.sh`
MaxBytes[cpu]:   80
AbsMax[cpu]:     100
Options[cpu]:    gauge, nopercent
Title[cpu]:      CPU State
PageTop[cpu]:    <h1>CPU State</h1>
YLegend[cpu]:    CPU State
ShortLegend[cpu]: %
LegendI[cpu]:     Idle:
LegendO[cpu]:     Active:
Legend1[cpu]:    Current Idle
Legend2[cpu]:    Current Active
Legend3[cpu]:    Peak Idle
Legend4[cpu]:    Peak Active
```


- ご清聴ありがとうございました。

参考資料:文献/URL

参考：書籍1

- UNIX MAGAZINE
  - 連載「Unix Communication Notes」山口　英　1998.3～
  - 「倉敷芸術科学大学のネットワーク構築」小林和真　1997.12
- OPEN DESIGN No.10
  - 「ネットワーク管理技術のすべて」
- Software Design 1999.9
  - 「フリーソフトウェアでネットワークをチェック
    `trafshow, MRTG, ntop`の導入」　田村吉章

参考：書籍2

- SNMP
  - "Snmp, Snmpv2, Snmpv3, and Rmon 1 and 2" -- William Stallings; 3rd edition (January 1999) Addison-Wesley Pub Co; ISBN: 0201485346 ;
  - "Practical Guide to SNMPv3 and Network Management, A" -- David Zeltserman, Dave Zeltserman; (May 4, 1999) Prentice Hall; ISBN: 0130214531
  - 「SNMPバイブル - インターネット管理への実践ガイド -」William Stallings著、大鐘久生、Addison-Wesley Publiching Company; ISBN-7952-9651-0

参考：性能評価

- Communication Traffic Project
  - http://www.mmlab.tnl.ntt.co.jp/
- Distributed Benchmark System
  - http://shika.aist-nara.ac.jp/member/yukio-m/dbs/index-j.html

Network Management

- http://wwwsnmp.cs.utwente.nl/Docs/software/pubdomain.html
- http://netman.cit.buffalo.edu/index.html
- http://www.nemoto.ecei.tohoku.ac.jp/~nitou/snmpdocs/tutorial1.html

ツールURL集１

- Angel Network Monitor
  - http://ibm-2.MPA-Garching.MPG.DE/angel/
- Big Brother
  - http://maclawran.ca/sean/bb-dnld/new-info.html
- Expect
  - http://expect.nist.gov/
- IPTraf
  - http://cebu.mozcom.com/riker/iptraf/index.html
- MRTG
  - http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/

ツールURL集２

- mon
  - http://www.kernel.org/software/mon
- NeTraMet
  - http://www.auckland.ac.nz/net/Accounting/ntm.Release.note.html
- NetSaint
  - http://www.netsaint.org/
- nocol
  - http://www.netplex-tech.com/software/nocol
- ntop
  - http://www-serra.unipi.it/~ntop/

ツールURL集３

- RRDTool
  - http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
  - Frontend - CRICKET
    - http://www.munitions.com/~jra/cricket/
  - Frontend - NRG
    - http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/frontends/nrg.html
  - Frontend - ORCA
    - http://www.gps.caltech.edu/~blair/
  - Frontend - Remstats
    - http://silverlock.dgim.crc.ca/~terskine/remstats/
- SPONG
  - http://www.edsgarage.com/projects/spong/

ツールURL集４

- Scotty
  - http://wwwhome.cs.utwente.nl/~schoenw/scotty/
- SWATCH
  - http://www.engr.ucsb.edu/~eta/swatch/
- statscout
  - http://www.statscout.com
- Treno
  - http://www.psc.edu/~pscnoc/treno.html
  - Experimental TCP Implementations
    http://www.psc.edu/networking/tcp.html
- UCD-SNMP
  - http://ucd-snmp.ucdavis.edu/

参考：URL集１

- General network management portal
  http://netman.cit.buffalo.edu/index.html
- Another good network management portal
  http://compnetworking.miningco.com/msubmanage.htm?terms=network+management&cob=home&TMog=5006366091143m&Mint=56534342191358&FFV=1
- “The Simple Times”
  http://www.simple-times.org/pub/simple times/issues/
- SNMP FAQ
  http://www.cis.ohio-state.edu/hypertext/faq/usenet/snmp-faq/part1/faq.html

参考：URL集２

- Sample Cisco device security configs
  http://www.cisco.com/warp/public/700/tech_configs.html#SECURITY
- Cisco device SNMP configuration tips
  http://www.cisco.com/warp/public/490/index.shtml

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/
  - http://www.caida.org/Tools/
    - cflowd ,RRD …etc
- LBNL's Network Research Group
  - http://ee.lbl.gov/
    - tcpdump, libpcap , arpwatch, traceroute, pathchar
- Solaris Freeware Project
  - http://sunsite.sut.ac.jp/sun/solbin/
- Fresh Meat - Linux Software Index
  - http://www.freshmeat.net/