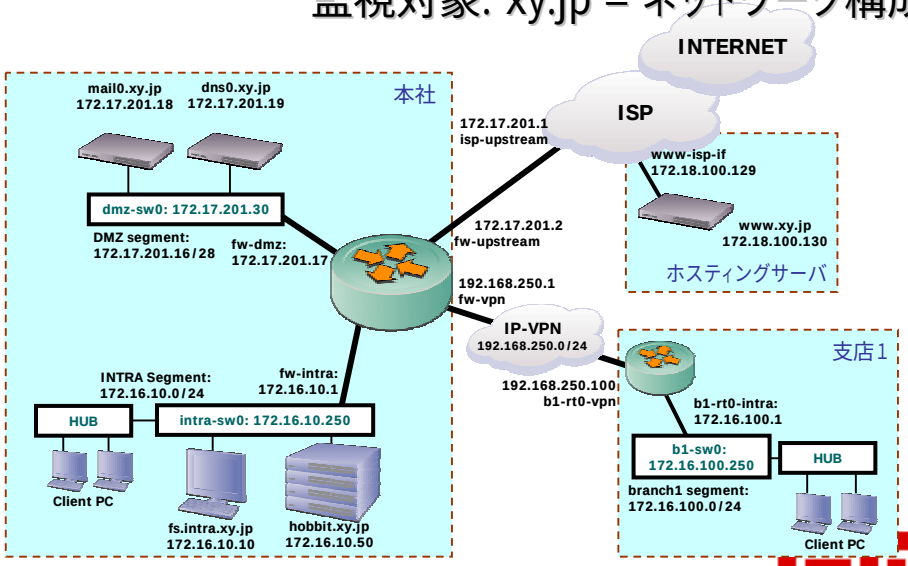
トラブルシュートを想定したネットワーク監視
～オープンソースソフトウェアによる実践～

2006/12/7
イー・モバイル株式会社
矢萩茂樹
(yahagi@emobile.jp)

本セッションの目的

- 本チュートリアルでは、ネットワーク監視と障害発生時のフローに着目し、効率的なネットワーク監視について考察する。
- 小規模ネットワークを仮想し、そのネットワークを監視するオープンソースベースツールの設定を元に、どのように監視を行えばトラブルシュートが行いやすくなるかを検討する。
- 取り上げるのは以下のツール
  - Hobbit Network Monitor
  - Nagios
  - その他

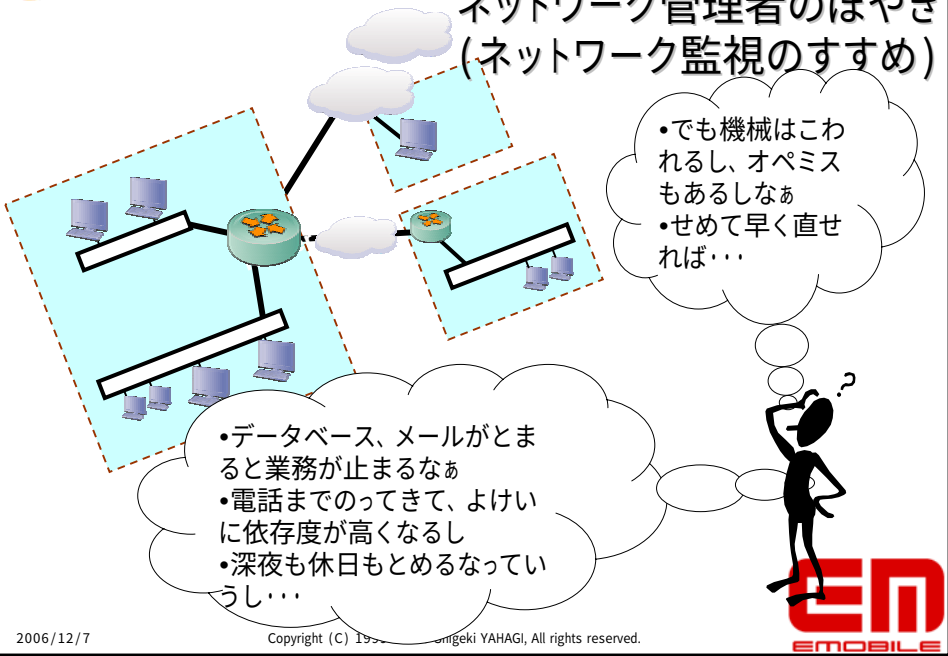
監視対象: xy.jp – ネットワーク構成



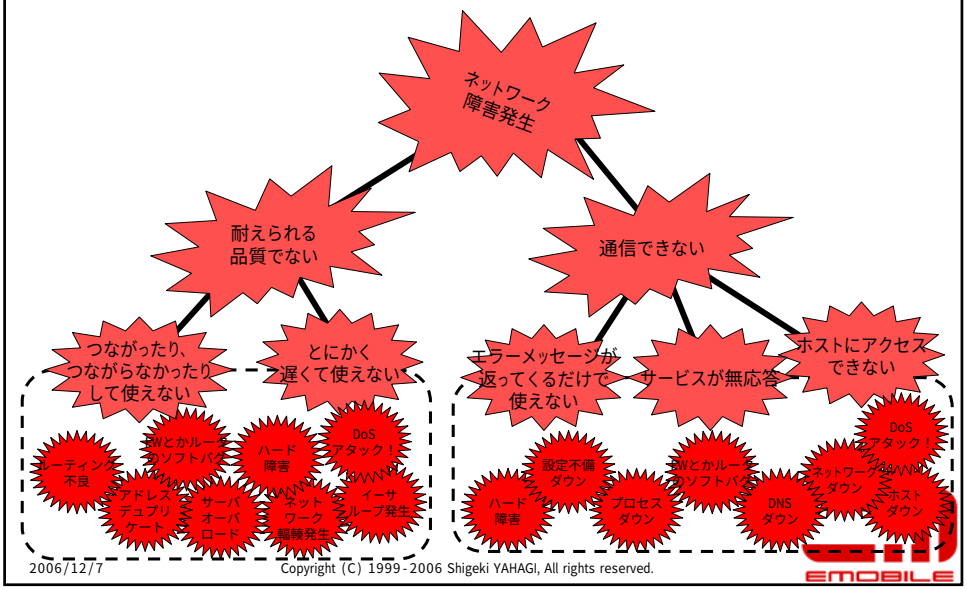
INDEX

ネットワーク管理者への文句

ネットワーク管理者のぼやき (ネットワーク監視のすすめ)



様々なネットワーク障害の例



きっかけは？

- トラブルシューティングのきっかけは二種類
  - ユーザクレームから始まる障害対応
    - 受動的きっかけ
  - ネットワーク監視から始まる障害対応
    - 能動的きっかけ

		ネットワーク監視システムの障害検知	
		あり	なし
ユーザからの申告	あり	ネットワーク共通部大規模障害	ユーザPCもしくは小規模ネットワーク障害
	なし	ネットワーク共通部障害	障害なし

監視できるもの･監視できないもの

- 監視できるもの･監視すべきもの
  - ルータ･L3スイッチなどL3機器
  - インテリジェントスイッチ･無線LAN APなどL2機器
  - 共通サービス提供機器
    - 各種サーバ･プリンタ･ロードバランサー･その他
  - 上流ISPとのインタフェース
  - ホスティングサーバと上流アドレス

リモートで情報が取得できるもの
共通使用される機材

- 監視できないもの（努力すればできるが大変なものも含む）
  - ユーザＰＣ
  - ノン･インテリジェント･スイッチ（馬鹿ハブ）

リモートで情報が取得できないもの。
個人特定なもの

大 ← 障害の影響度 → 小

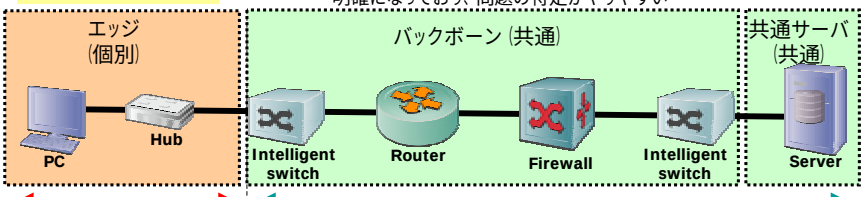
能動的に対処可能

基本的に受動的に対処

障害切り分けのしやすいネットワーク しずらいネットワーク

障害切り分け
しやすいネットワーク

重要度に応じて監視可能な部分と、しなくても良い部分が明確になっており、問題の特定がやりやすい

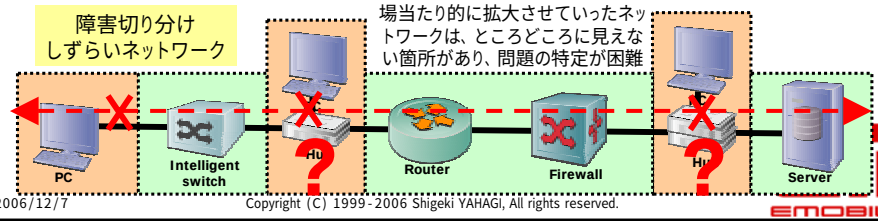


稼動を掌握できない部分：
エッジ部の装置は能動的にIPに答えないものが多く、監視が困難

稼動を掌握できる部分：
共通設備部分は全設備について死活確認可能なもので組み、全ポイント確認が必要

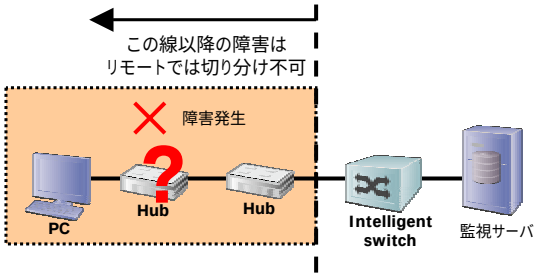
障害切り分け
しずらいネットワーク

場当たり的に拡大させていったネットワークは、ところどころに見えない箇所があり、問題の特定が困難



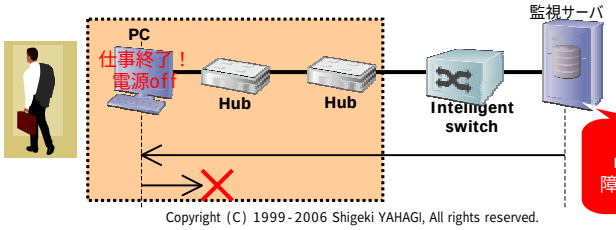
監視できる部分とできない部分：1

- エッジ部分の問題 → （機能的に･利用状況的に）監視が困難
- エッジ部分のスイッチは馬鹿ハブ使うのが一般的なのでリモートで状況把握ができない
  - 上位階層機器がインテリジェント型なら、リンクダウン･デュプレックスミスマッチなどの検出でその区間の異常判断はつくが、ハブから先の装置の障害はみれない

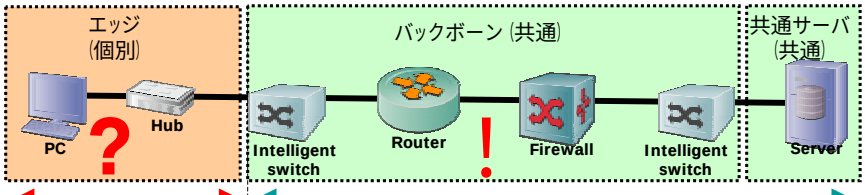


監視できる部分とできない部分：2

- リモート監視は落とさないことが前提の装置にしか適用できない
  - 仕事がおわると電源が落とされるPCは能動的な監視はできない
    - 仕事が終わって落としたのか、障害なのかの判断がつかない
    - 誤報はネットワーク監視する上での重大問題
    - 最近のPCってデフォルトping受け付けないので、設定変更も結構大変。
  - やるとするとクライアントからkeep aliveするようなプローブをいれるしかない
    - 商用ソフトによる対応
    - ここまでやるかどうかはネットワークの運用ポリシーに大きくかかわります
    - ユーザPCの監視ってむしろセキュリティ問題の方が大切なような気が･･･



監視できる部分とできない部分：3



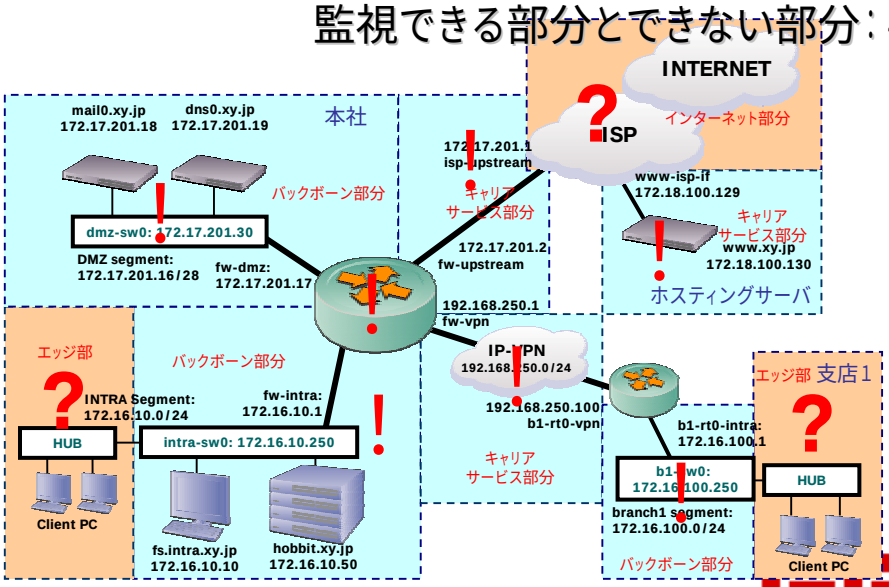
稼動を掌握できない
セグメント

稼動を掌握できる
セグメント

ユーザからの申告で対応
受動的対応

能動的にネットワーク監視を行い
稼働率をあげることが可能

監視できる部分とできない部分：4



受動的対応：ユーザクレームから始まるトラブルシューティング：1

- エッジ部分の障害って
  - ケーブル障害（切断、抜け、ルーズコネクト、ポートの挿し間違いetc）
  - 電源障害（故障、電源ケーブルぬけ）
  - ネットワーク障害
    （ブリッジループ、デュプレックスずれ、etc）
  - ＰＣハード障害（イーサIF故障、）
  - ＰＣソフト障害（設定不良、ドライバ不良、）
  - その他（DHCPプール不足とか？）
- エッジ部分の障害対応では事前準備が重要
  - ネットワーク構成図（ポート情報、接続先、etc）
  - 接続しているホスト情報（IPアドレス、MACアドレス、OS、ハード情報、etc）
    - クライアント情報の管理はかなり必要

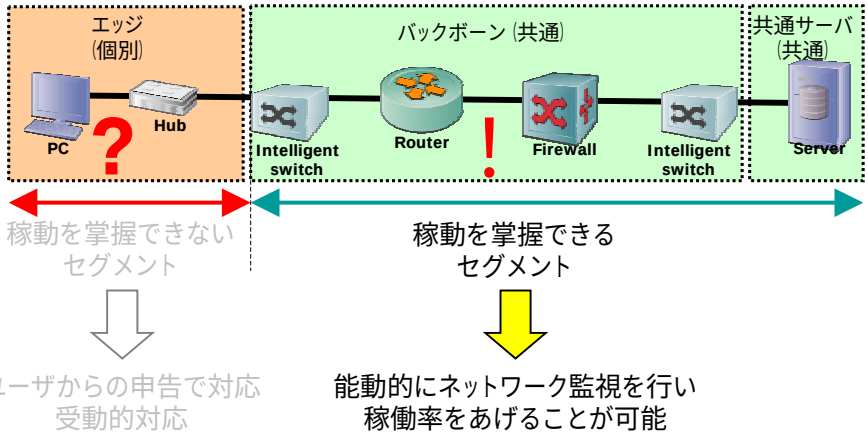
受動的対応：ユーザクレームから始まるトラブルシューティング：2

- 実際のトラブルシュートは申告場所で実施
  - 物理的な確認：ランプ状態、ケーブル接続状況、
  - 各種ツールの活用：
    - ネットワークスキャナ：look@lan など
    - 高速pingツール：ExPing など

look@lan

ExPing

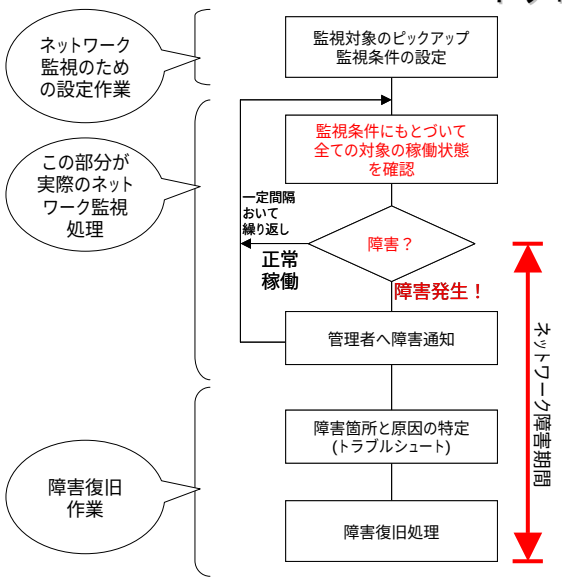
能動的対応：ネットワーク監視



能動的対応：ネットワーク監視とは

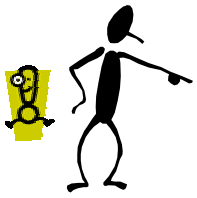
- ネットワークの稼働状態を監視する業務
  - 障害を的確に検知し、迅速な障害復旧を行うために行う
- フローを文章化すると
  - 監視すべきノードをピックアップし、
  - 監視条件を設定し、
  - 全ての監視対象を漏れなく定期的に稼働監視し、
  - 問題があったら障害を管理者に通知する。
- ネットワーク監視の後には障害復旧処理が続く
  - 障害原因を特定(トラブルシュート)し、
  - 障害ノードの復旧を行う

ネットワーク監視のフロー

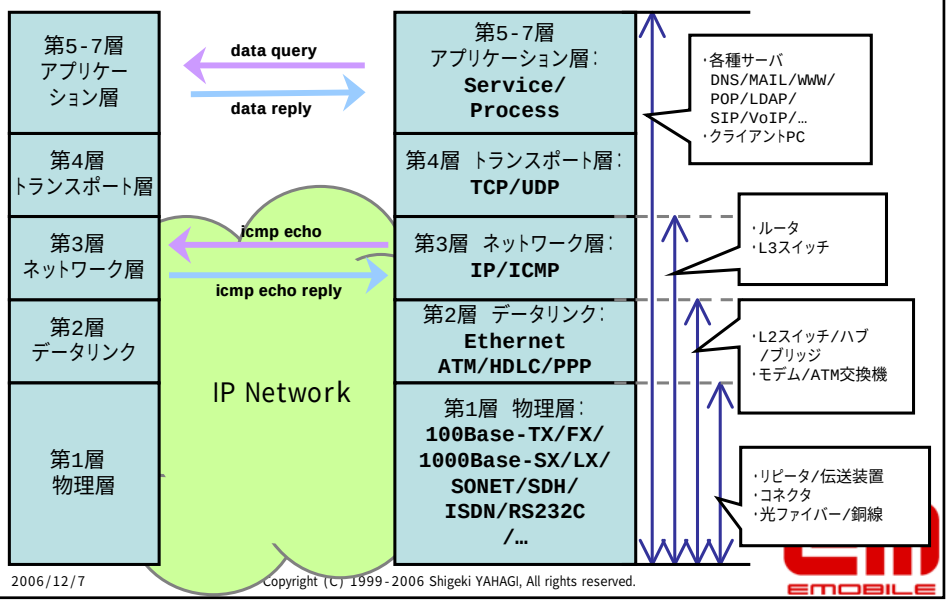


ネットワーク障害を早期復旧させるには

- 漏れのない稼働状態確認
- 的確な障害通知
- 速やかな障害原因調査開始
- 迅速な障害原因の特定
- 障害復旧のための準備

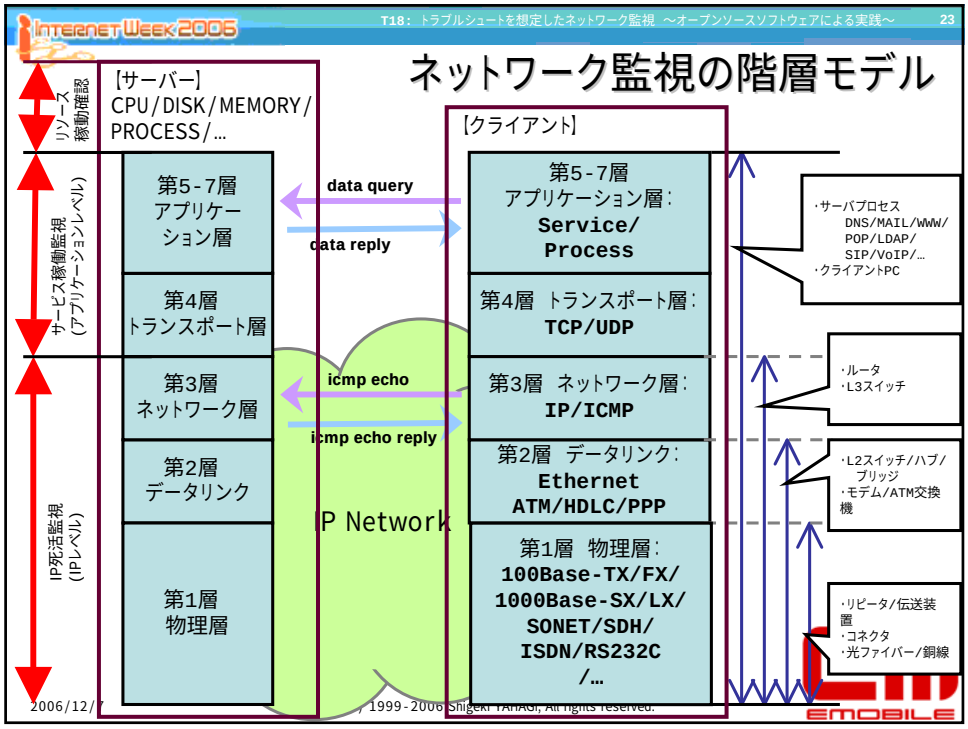


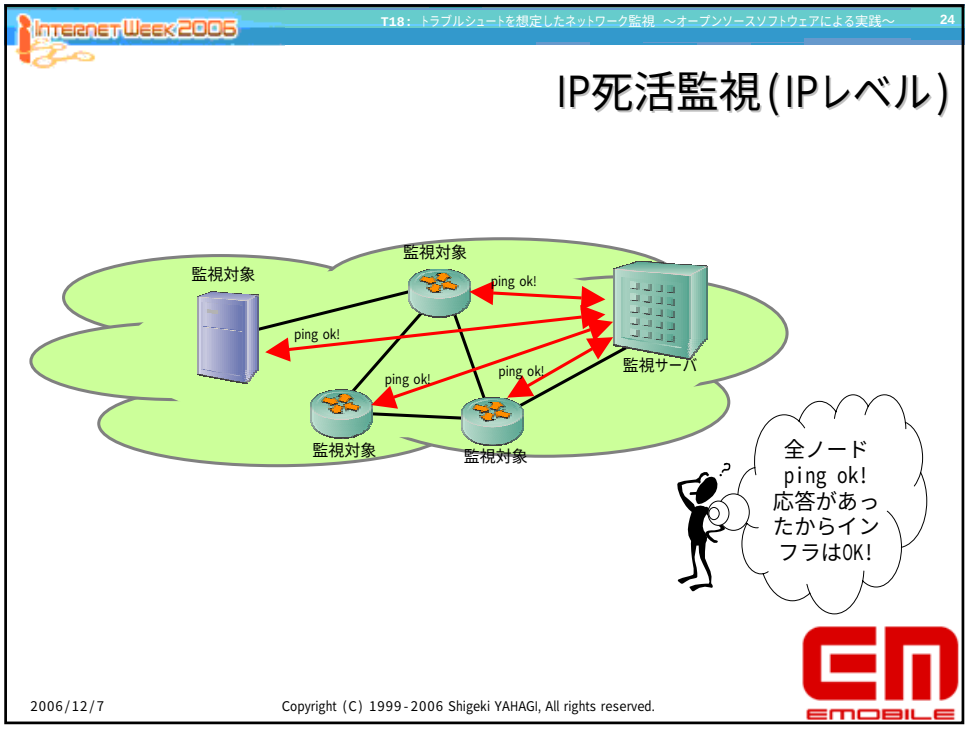
ネットワーク監視対象



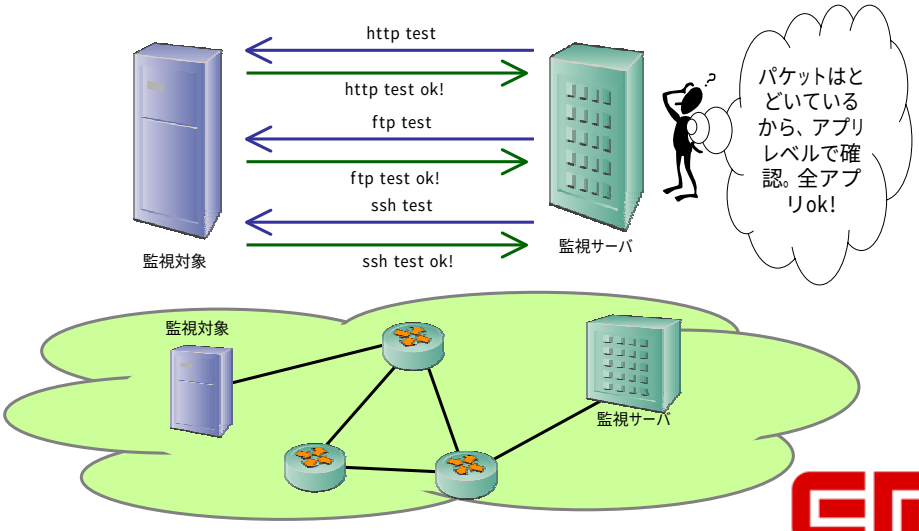
プロトコルスタックに準じたネットワーク監視の種類

- IP死活監視：IP基盤確認
  - 監視対象がIP的に生存していることを確認する
- サービス提供状態監視：提供サービス確認
  - サービスが問題なく稼動していることを確認する
- リソース監視：システム稼働状態確認
  - プロセスが正常に起動していることを確認する。
  - 十分なリソースが確保されていることを確認する
    - CPU/DISK/MEMORY/PROCESS

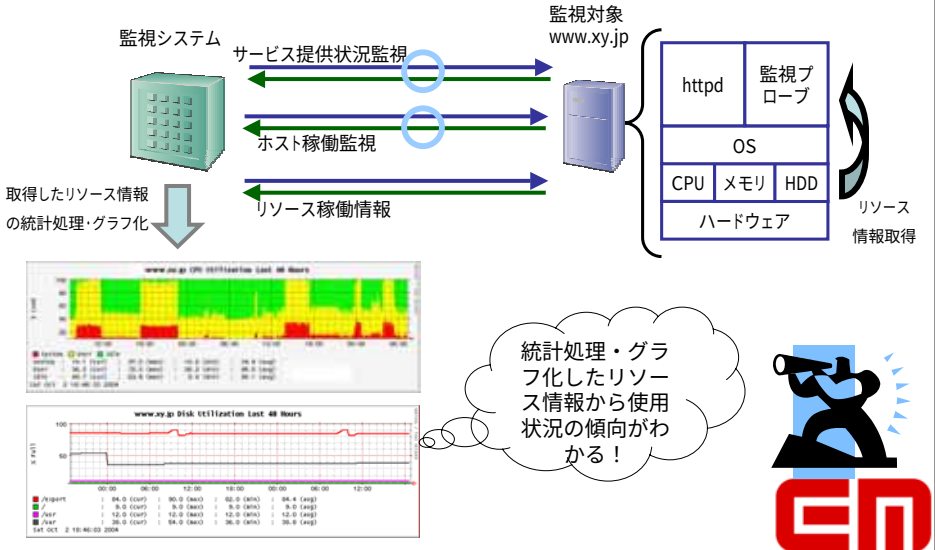

INTERNET WEEK 2006
T18: トラブルシュートを想定したネットワーク監視 ～オープンソースソフトウェアによる実践～ 23
ネットワーク監視の階層モデル
リソース稼動確認
サービス稼働監視(アプリケーションレベル)
IP死活監視(IPレベル)
[サーバー]
CPU/DISK/MEMORY/PROCESS/…
第5-7層 アプリケーション層
第4層 トランスポート層
第3層 ネットワーク層
第2層 データリンク
第1層 物理層
data query
data reply
icmp echo
icmp echo reply
IP Network
[クライアント]
第5-7層 アプリケーション層: Service/Process
第4層 トランスポート層: TCP/UDP
第3層 ネットワーク層: IP/ICMP
第2層 データリンク: Ethernet ATM/HDLC/PPP
第1層 物理層: 100Base-TX/FX/1000Base-SX/LX/SONET/SDH/ISDN/RS232C/…
・サーバプロセス DNS/MAIL/WWW/POP/LDAP/SIP/VoIP/…
・クライアントPC
・ルータ
・L3スイッチ
・L2スイッチ/ハブ/ブリッジ
・モデム/ATM交換機
・リピータ/伝送装置
・コネクタ
・光ファイバー/銅線
2006/12/7
1999-2006 Shigeki YAHAGI, All rights reserved.
EMOBILE


INTERNET WEEK 2006
T18: トラブルシュートを想定したネットワーク監視 ～オープンソースソフトウェアによる実践～ 24
IP死活監視(IPレベル)
監視対象
監視対象
監視サーバ
監視対象
監視対象
ping ok!
ping ok!
ping ok!
ping ok!
ping ok!
全ノード ping ok! 応答があったからインフラはOK!
2006/12/7
Copyright (C) 1999-2006 Shigeki YAHAGI, All rights reserved.
EMOBILE

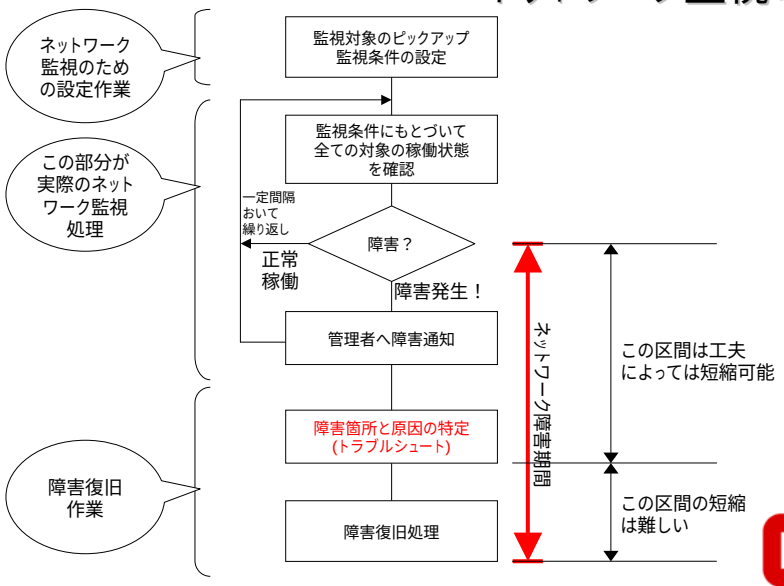
サービス稼働監視(アプリケーションレベル)



リソース監視 (システムレベル)



ネットワーク監視のフロー



ネットワーク障害の分類

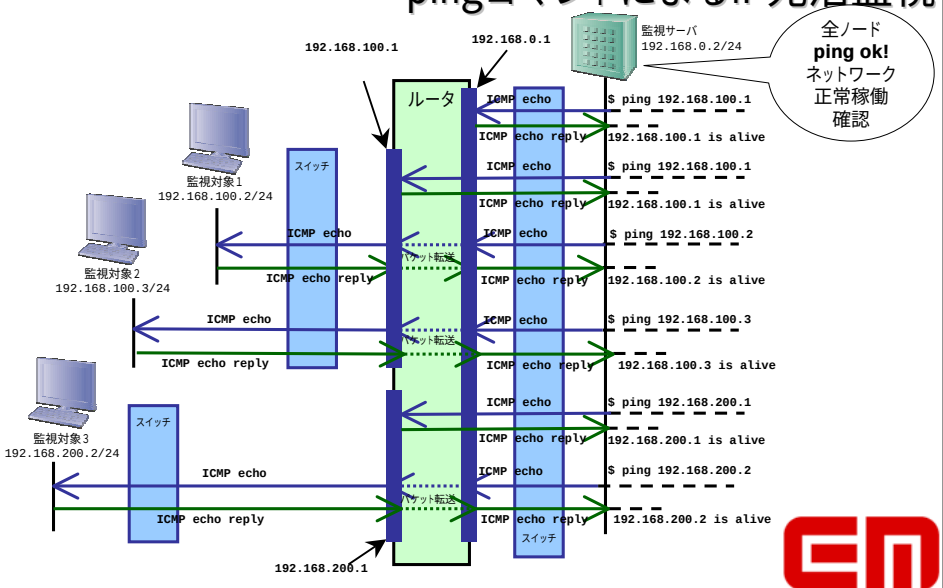
- ネットワーク監視では”単体障害”・”複数障害”で対応がわかれる
- 単独ホスト障害は切り分けは見ての通りでわかりやすい
- 複数ホスト障害の場合にはトラブルシュートによる原因追求が必要
  - tracerouteによる障害ポイントの把握
  - ネットワーク接続図と障害ホストのマッピングによる障害ポイントの絞込み
- 絞込みの条件
  - 単独？：
    - 対象確定！
  - 複数？：
    - 同一ルータ？, 同一ネットワーク？. 同一セグメント？, 同一スイッチ？, …

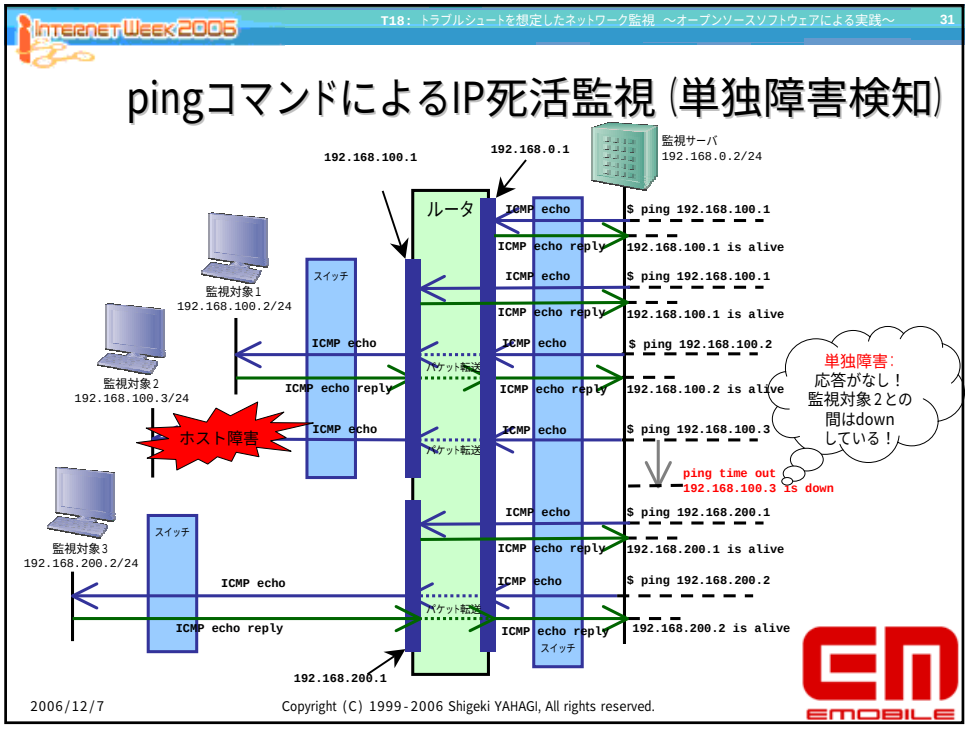
迅速なトラブルシュートのための準備

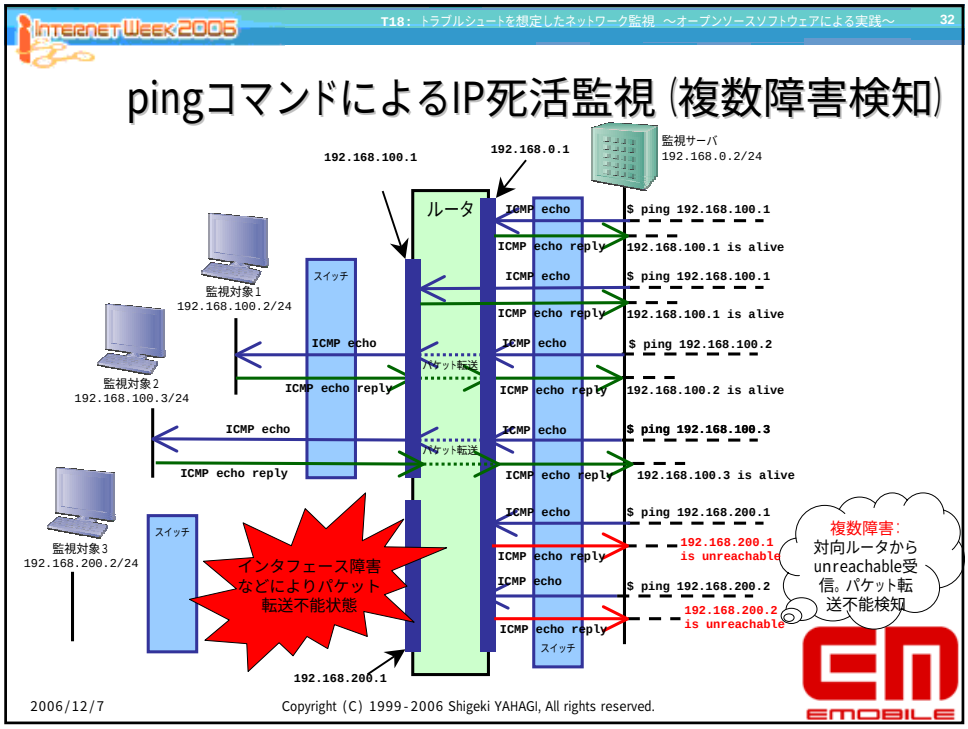
- 時間がかかるのは複数障害の切り分け作業
  - トラブルシュートの手順は基本的に同一
    - tracerouteによる障害ポイントの把握
    - ネットワーク接続図と障害ホストのマッピングによる障害ポイントの絞込み
  - この情報整理ができているかどうかが、早期解決のわかれめ
    - これらの情報はネットワーク構築時確定している
    - トラブルになってから調べるのでは遅い

  - 整理した情報そのまま、トポロジーごとネットワーク監視システムに設定するのが肝！

pingコマンドによるIP死活監視




INTERNET WEEK 2006
T18: トラブルシュートを想定したネットワーク監視 ～オープンソースソフトウェアによる実践～
31
pingコマンドによるIP死活監視（単独障害検知）
192.168.100.1
192.168.0.1
監視サーバ
192.168.0.2/24
ルータ
監視対象1
192.168.100.2/24
スイッチ
監視対象2
192.168.100.3/24
ホスト障害
監視対象3
192.168.200.2/24
スイッチ
192.168.200.1
ICMP echo
ICMP echo reply
$ ping 192.168.100.1
192.168.100.1 is alive
$ ping 192.168.100.2
192.168.100.2 is alive
$ ping 192.168.100.3
ping time out
192.168.100.3 is down
$ ping 192.168.200.1
192.168.200.1 is alive
$ ping 192.168.200.2
192.168.200.2 is alive
パケット転送
単独障害：
応答がなし！
監視対象2との
間はdown
している！
2006/12/7
Copyright (C) 1999-2006 Shigeki YAHAGI, All rights reserved.
EMOBILE


INTERNET WEEK 2006
T18: トラブルシュートを想定したネットワーク監視 ～オープンソースソフトウェアによる実践～
32
pingコマンドによるIP死活監視（複数障害検知）
192.168.100.1
192.168.0.1
監視サーバ
192.168.0.2/24
ルータ
監視対象1
192.168.100.2/24
スイッチ
監視対象2
192.168.100.3/24
監視対象3
192.168.200.2/24
スイッチ
192.168.200.1
インタフェース障害
などによりパケット
転送不能状態
ICMP echo
ICMP echo reply
$ ping 192.168.100.1
192.168.100.1 is alive
$ ping 192.168.100.2
192.168.100.2 is alive
$ ping 192.168.100.3
192.168.100.3 is alive
$ ping 192.168.200.1
192.168.200.1
is unreachable
$ ping 192.168.200.2
192.168.200.2
is unreachable
パケット転送
複数障害：
対向ルータから
unreachable受
信。パケット転
送不能検知
2006/12/7
Copyright (C) 1999-2006 Shigeki YAHAGI, All rights reserved.
EMOBILE

様々な障害と想定される原因の例

- 特定の端末が通信不能
  - 端末障害
  - 上位ブリッジ障害
  - ・・・
- 特定のサブネットが通信不能
  - サブネット収容ルータ障害
  - サブネット収容スイッチ障害
  - ・・・
- 特定の拠点が通信不能
  - 拠点間回線障害
  - ルータ障害
  - ・・・
- まったく通信できない
  - DNS障害
  - インターネットゲートウェイ障害
  - ルーティング障害
  - ・・・
- 特定のサービスが通信不能
  - サーバ障害
  - その他、アプリケーションシステム障害
  - ファイヤーウォール障害
  - ・・・

障害ポイントの割り出しの基本

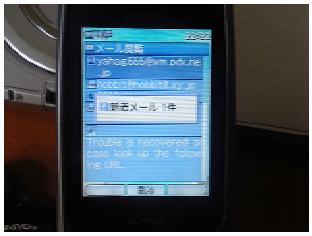
- サービス稼働監視･リソース監視とIP死活監視の複合障害の関係
  - サービスはIP基盤の上で稼動している
  - IPレイヤで障害となっている場合、サービスも当然障害となる
- その上で障害情報解析の基本は共通障害ポイントの割り出し
  - 単独障害 → 障害ポイント確定！
  - 多重障害 → 共通障害ポイントの割り出しが必要
- 多重障害での分析
  - IP死活監視多重障害 → ネットワーク障害
  - サービス稼動監視多重障害 → ネットワーク障害 or 共通設備障害
  - リソース監視多重障害 → ネットワーク障害 or 共通設備障害

サービス稼働監視・リソース監視とIP死活監視の複合障害の関係

			IP死活監視		
			警報無し	単独ノード警報	複数ノード警報
サービス稼働監視	単独ノード警報	単独サービス警報	単独ノード・単独サービス障害	単独ホスト障害	ネットワーク障害
		複数サービス警報	単独ホスト障害	単独ホスト障害	ネットワーク障害
	複数ノード警報	単独サービス警報	複数ノード・単独サービス障害	①単独ノード障害に起因する単独サービス障害 ②ネットワーク障害	ネットワーク障害
		複数サービス警報	①ネットワーク通信品質異常などによるサービス障害 ②NMS処理異常	①単独ノード障害に起因する複数サービス障害 ②NMS処理異常	①ネットワーク障害 ②NMS処理異常
リソース監視	単独ノード警報	単独リソース警報	単独ノード・単独リソース障害	単独ノード障害	ネットワーク障害
		複数リソース警報	単独ノード障害	単独ノード障害	ネットワーク障害
	複数ノード警報	単独リソース警報	障害リソースにまつわるネットワーク障害	①単独ノード障害に起因する単独リソース障害 ②ネットワーク障害	ネットワーク障害
		複数リソース警報	①ネットワーク通信品質異常などによるリソース障害 ②NMS処理異常	①単独ノード障害に起因する複数リソース障害 ②NMS処理異常	①ネットワーク障害 ②NMS処理異常

障害通知について：1

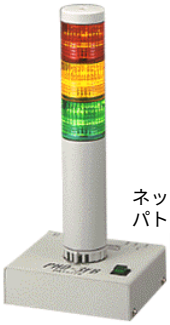
- 適切な障害通知ポイントの設定と確実な通知方法
- 障害検知後、管理者に対して速やかにイベントの報告を行う
  - 障害システム／イベント／時間により障害通知先を判断し、通知を行う。
  - 確実に届いて確実に反応するエスカレーション体制が一番重要
  - 通知されても対処開始が遅れれば、遅れた分だけ障害が重大化していく！
- メールによる障害発生通知
  - 通知には以下の情報を含めると迅速な対応が可能
    - 障害発生時刻
    - 障害発生個所・機器
    - 障害状況
    - 障害サマリーページへのＵＲＬ情報



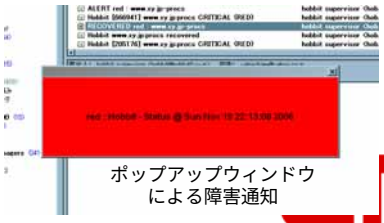
携帯メール
による障害通知

障害通知について：2

- 監視者への積極的な通知方法
  - パトランプ（音・光）による警報
  - POPUP WINDOWなどによる通知
- 定期メンテナンスやエスカレーション対象外の通知を抑制
  - 計画停止を障害検知しない仕組みは対応速度や精度を上げるためにも重要！

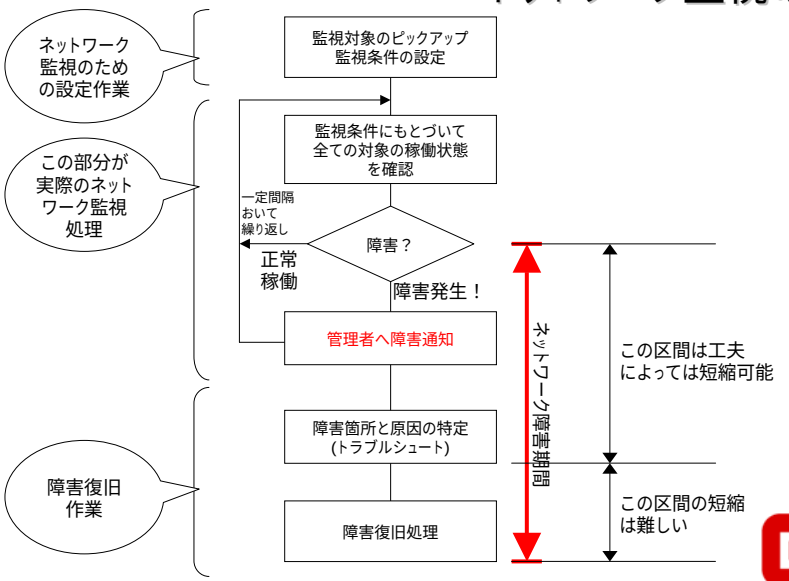


ネットワーク対応
パトランプによる
障害通知



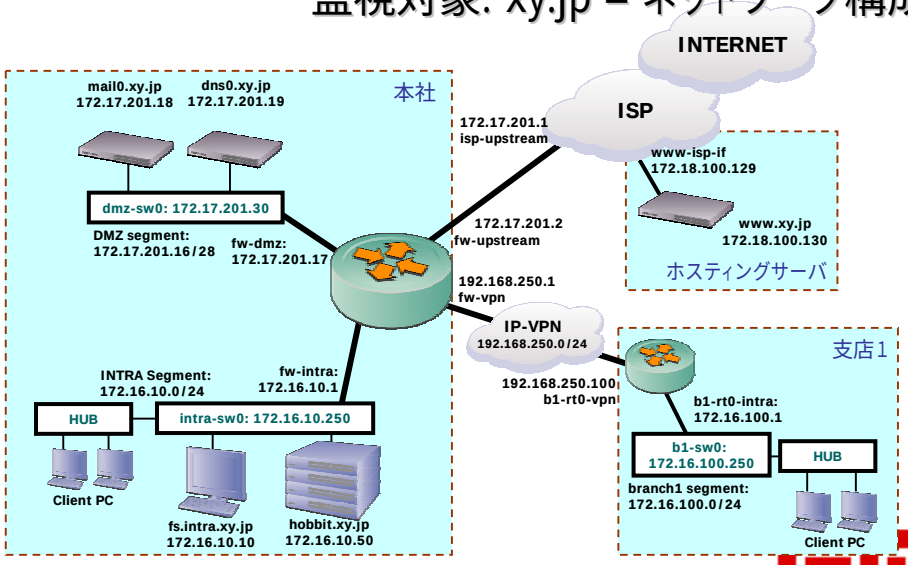
ポップアップウィンドウ
による障害通知

ネットワーク監視のフロー



INDEX

監視対象: xy.jp – ネットワーク構成



監視対象 – 概要1

- 小規模企業のエンタープライズネットワークを想定。
- 仮想ネットワークはGlobal Address/Domainを取得／管理しており、ISPを経由してThe Internetとの接続を行っている。
- 本社側にファイヤーウォールを導入しており、社内からインターネットへの接続はすべて本社ファイヤーウォールを経由する
- ISPとの接続は本社FWからFTTH経由で接続
- 本社ー支社間はIP-VPN経由で接続
- 本社はこのほかにSub Allocation Block（172.17.201.16/28）の割当を受ける
- グローバルアドレスが振られるサーバはすべて本社ファイヤーウォールのDMZ配下に配置
- www.xy.jp(172.18.100.130)はISPのサーバホスティングを使用し、ISPデータセンターに設置
- ファイヤーウォール配下のネットワークはPrivateアドレスを使用し、Firewallにて NAPT(Network Address/Port Translation)している

2006/12/7

監視対象分析 – IPアドレスブロック割当

セグメント	アドレスブロック	用　途
本社DMZセグメント	172.17.201.16/28	ISP割当グローバル
www.xy.jpセグメント	172.18.100.128/30	www.xy.jpホスティンググローバル
本社イントラセグメント	172.16.10.0/24	イントラ向けプライベート
本社WANセグメント	172.17.201.0/30	ISP割当グローバル
本社ー支社間セグメント	192.168.250.0/24	WAN機器チェック用プライベート
支社イントラセグメント	172.16.100.0/24	イントラ向けプライベート

2006/12/7

監視対象分析 - 提供サービス1

- ネットワーク提供サービス
  - 社外向けサービス
    - DNS/MAIL(SMTP)/WWW
  - 社内向けサービス
    - DNS/MAIL(SMTP/POP)/WWW(Intra)
    - DHCP
    - File Server/Print Server
  - 共通ポート
    - メンテナンスはTELNETは使用せず、SSHのみ。
    - FTPサービスも社外向けには開いていない
    - SMTPサービスは必要なサーバのみに限定
    - 社外へはポートはあけておらず、IPsec/PPTP VPN経由で内部からのみLOGIN可能とする

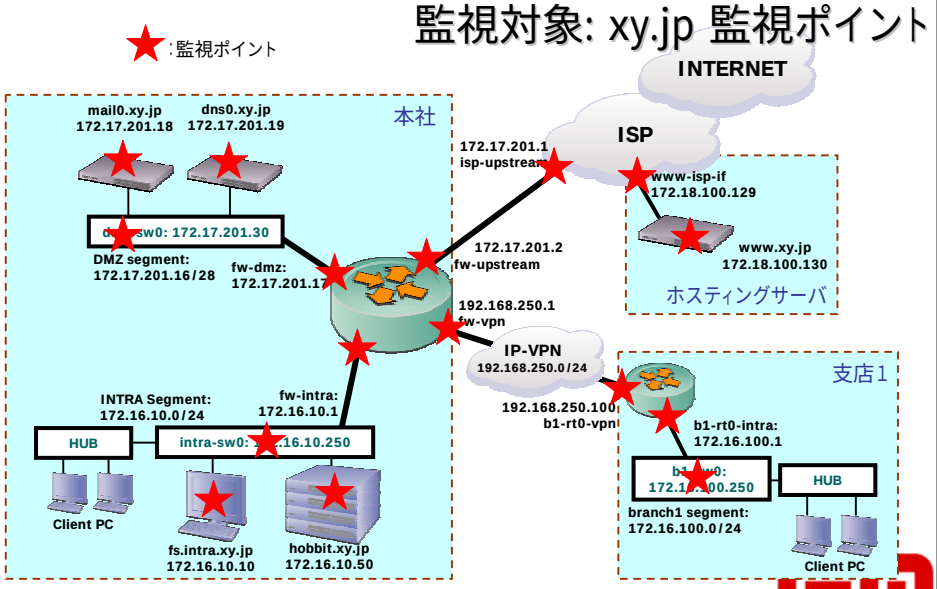
監視対象分析 - 提供サービス1

- ネットワーク提供サービス2
  - DNS設定
    - Primary: dns0.xy.jp（172.17.201.18）
    - Secondary: mail0.xy.jp（172.17.201.19）
  - メール設定
    - Primary: mail0.xy.jp
    - Secondary: dns0.xy.jp
  - POPは社内のみに制限。
  - 社外からのアクセスはVPNを経由してのみ可能

監視ポイントの割り出し

- 監視ポイントは以下の設備を対象とする
  - 共用サービスを提供している設備：各種サーバ
  - ネットワーク設備：ルータ・スイッチ・ファイアウォール
    - イーサネット区間は障害となると切り分けが難しいことから、基幹部分のスイッチはSNMP対応のインテリジェントスイッチ導入が好ましい
    - ファイアウォールはデフォルトではICMPには答えない。監視するためには返答するようなポリシー適用が必要
- 監視対象外の設備
  - クライアントPCやオンラインと関係のない開発サーバなどは対象外
  - 稼働時間が規定されるノードの設定には注意が必要
- 該当するノードと提供サービスを漏れなく、ピックアップすることが重要
  - ぬけていては管理されていないのと同じ

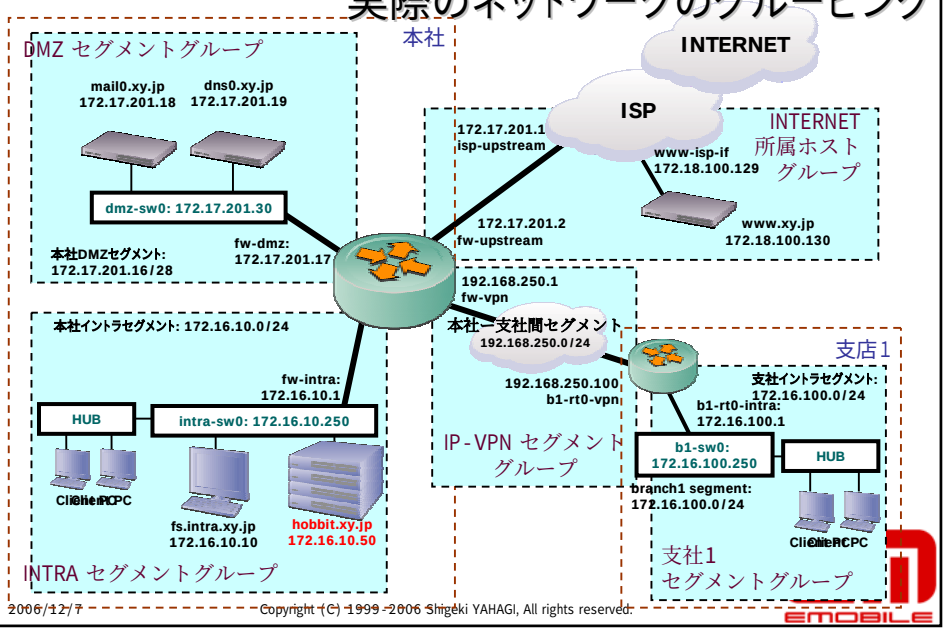
監視対象: xy.jp 監視ポイント



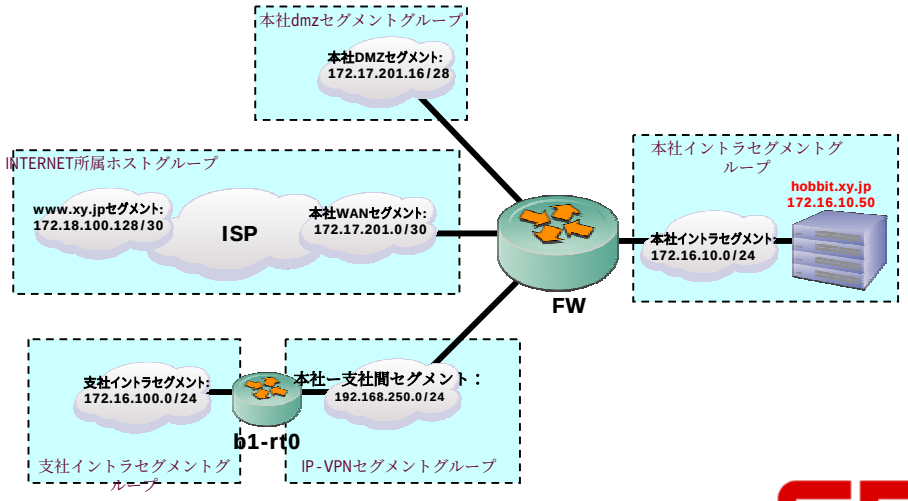
監視ポイントのグルーピング

- ネットワーク監視を効率的に行う上で整理整頓が重要
- 障害発生時のフローは以下の手順が基本
  - IP死活監視が失敗！
  - 失敗したノードの物理トポロジーに基づいて、直前のポイントまでからのping確認を実施。詳細の切り分けを行う
- IPネットワークの依存関係はあくまでもサブネットセグメントとそれをつなぐルータの木構造
- セグメント毎にグルーピングし、監視サーバをトップノードとしたセグメントツリーを作る

実際のネットワークのグルーピング



各グループのトポロジー整理



監視対象分析 – 監視ホスト一覧

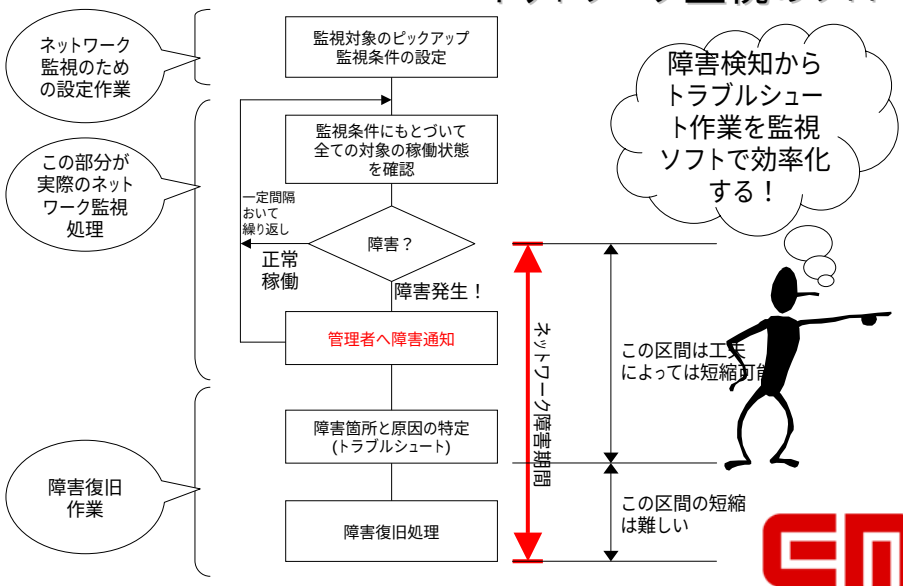
セグメント	IP address	ホスト名称	URL	提供サービス
本社イントラセグメント (172.16.10.0/24)	172.16.10.1	fw-intra	- - -	firewall
	172.16.10.10	fs.intra.xy.jp	- - -	FileServer
	172.16.10.50	bb0.xy.jp	bb0.xy.jp	http, ssh
	172.16.10.250	intra-sw0	- - -	switch
本社DMZセグメント (172.17.201.0/28)	172.17.201.17	fw-dmz	- - -	firewall
	172.17.201.18	mail0.xy.jp	mail0.xy.jp	dns, smtp, pop, ssh
	172.17.201.19	dns0.xy.jp	dns0.xy.jp	dns, smtp, ssh
	172.17.201.30	dmz-sw0	- - -	switch
本社WANセグメント (172.17.201.0/30)	172.17.201.1	isp-upstream	- - -	ISP-Router
	172.17.201.2	fw-upstream	- - -	firewall
www.xy.jpセグメント (172.18.100.128/30)	172.18.100.130	www.xy.jp	www.xy.jp	http, ftp, ssh
	172.18.100.129	www-isp-if	- - -	ISP-Router
本社ー支社間セグメント (192.168.250.0/24)	192.168.250.1	fw-vpn	- - -	firewall
	192.168.250.2	b1-rt0-vpn	- - -	router
支社イントラセグメント (172.16.100.0/24)	172.16.100.1	b1-rt0-intra	- - -	firewall
	172.16.100.250	b1-sw0	- - -	switch

監視対象分析 – 監視時間と通知先

- 全ての機器の障害情報は障害受付窓口であるalert@xy.jpに通知
- 独自のイントラ系と支社ネットワークの部分については以下の監視･障害通知ポリシーを適用
  - 本社ファイルサーバ fs0.intra.xy.jp ：
    - 毎日午前4時から6時の間でデイリーバッチ処理が走り、高負荷となることから監視を停止。監視省力化
    - この機械の障害時には担当窓口：intra@xy.jpにも通知
  - 支社機器の障害対応は現地の担当に任せることが多いためにalert@branch.xy.jpへの通知を追加

INDEX

ネットワーク監視のフロー



ネットワーク監視ソフトの一例

ネットワーク監視ソフト導入のメリット

- ネットワーク監視ソフトのうれしさ
  - 監視業務の自動化 - 24x7で自動監視
    - 漏れのないネットワーク監視
    - 適切な障害通知
  - 障害原因解析の効率化
    - 障害情報の多面的な解析
    - 共通障害ポイントの把握
  - 障害の事前予測の実現
    - 障害パターンの長期把握
  - ネットワーク稼動品質の向上
    - 稼働レポート
    - 過去の稼動状態の履歴検索
- ネットワーク監視ソフトの例
  - Hobbit / Nagios/Big Brother/ OpenNMS/ Big Sister/ Mon/ Demarc/ Nocol/ Spong/ MTR/ Scotty/ SysOrb/ Nisca/ Zabbix/...

ネットワーク監視ソフト - Hobbit

- **http://hobbitmon.sourceforge.net/**
- Webベース監視システムBig Brotherの思想を受け継ぎ、劇的に高速化･高機能化した監視システム
  - Big Brotherを高速化するbbgenという拡張スクリプトを基礎にBBに依存していた部分を独自に追加
  - 完全オープンソース化
- BBのモジュール構成を受け継ぎ、監視･表示･通知機能に分割
  - C言語で完全に書き直すことで大規模ネットワークまで適用可能なパフォーマンスを持つ
  - チューニングすれば障害検知時間を1 - 2分程度までにすることも可能
- 各種Unix用プローブクライアントのほか、BBクライアントとの相互接続が可能
  - BB用Windows NT系の監視用プローブが使用可能であり、複合OS統合監視が可能

ネットワーク監視ソフト - Hobbit

- ICMP/TCPポーリングによる監視を行う
  - 監視可能サービス:
    - `ping,smtp,http、https,pop3,dns,ftp,telnet,ssh,imap,nntp, ・・・`
  - リソース監視:
    - `CPU,disk,processes`
- 監視対象のグループ化機能
- 監視画面の階層化機能(3段階以上可能)
- 特定のホストだけで構成する監視画面を仮想的に作ることも可能
- 柔軟なアラーム通知機能
  - E-mailによりアラームを通知する
  - 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
- アラームの検出されている機器のみサマリ画面を標準で生成
  - アラームメッセージに障害情報ページのURLが引用されており、迅速に障害情報に到達可能

ネットワーク監視ソフト - Hobbit

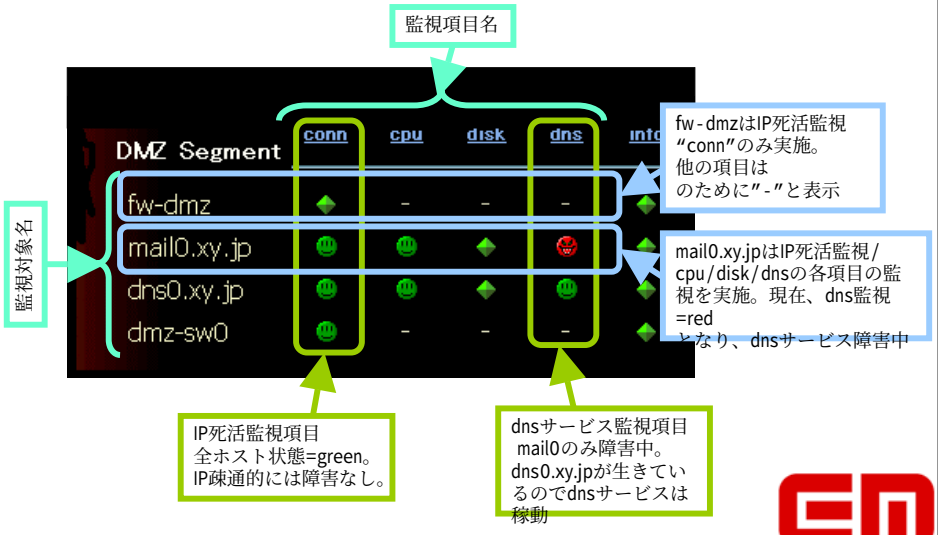
- BBの機能拡張モジュールと同じ拡張インタフェースをもち、再利用可能
  - 拡張監視モジュール: DBMS, ファイルサーバ, プリンタサーバ, ・・・
  - 他ソフトとの連係: MRTG, Snort, tripwire, ・・・
  - BBTray : Big Brother監視ツール on Windows
- BBで定番となっている以下の機能は標準機能として統合
  - RRDtoolによる監視リソースのグラフ化
  - 特定対象の監視を一時的に停止するGUI機能
  - fpingによる高速IP死活監視
  - 長期間にわたる障害履歴表示機能
- システム稼動状況レポート作成機能の充実

Hobbit：監視画面（TOP）

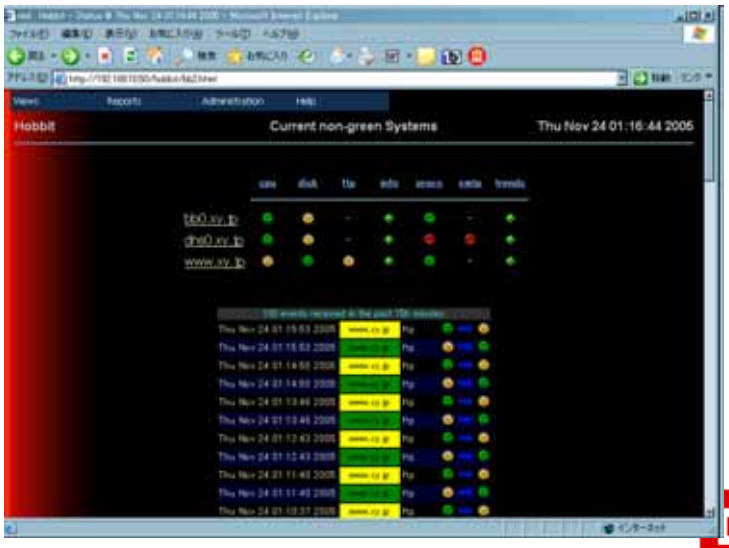
Hobbit監視サーバー：稼働状態アイコン

	24時間以内の状態変化の場合	24時間以上状態変化がない場合
Green 緑 ： 障害なし		
Yellow 黄 ： 軽微障害発生		
Red 赤 ： 重要障害発生		
Clear 無色 ： データなし		
Purple 紫 ： データ取得不能		
Blue 青 ： 監視一時中止		

Hobbit監視状態表示：全体表示



Hobbit：障害サマリ



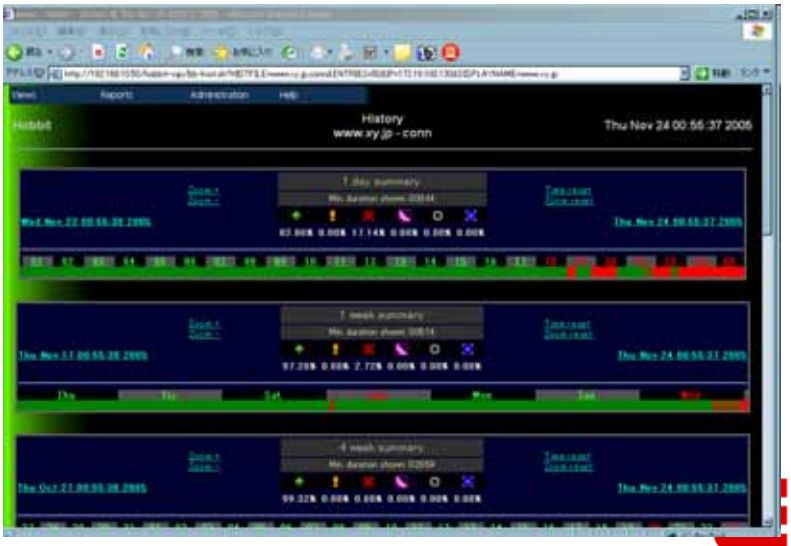
障害サマリ画面の見方

dns0.xy.jpのsmtpサービス監視とプロセス監視にてクリティカル・アラーム発生。

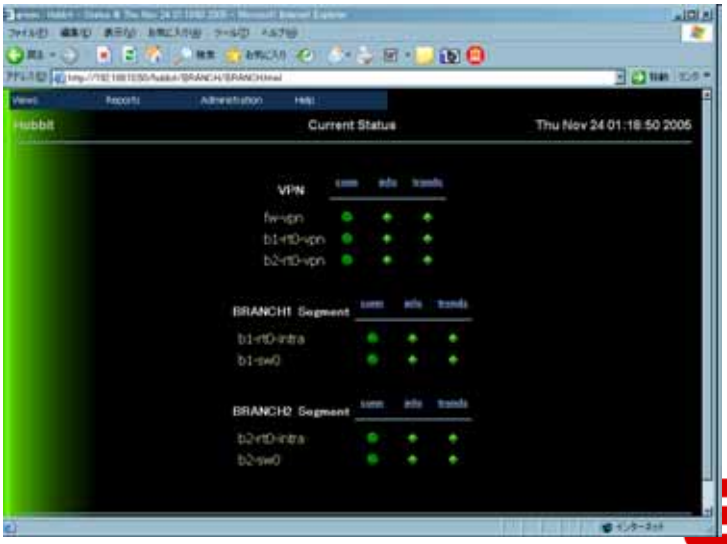
他のホストでは重大障害が発生していないことから、プロセス障害の可能性が高い。

このアイコンをクリックして詳細情報画面で確認

Hobbit：稼動履歴画面



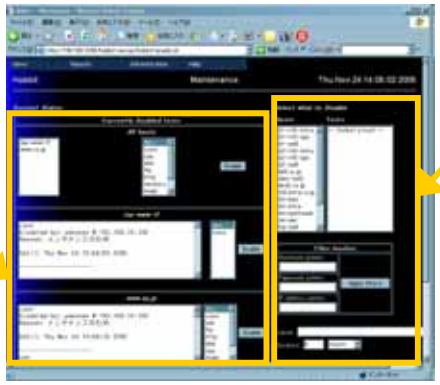
Hobbit：サブ監視画面



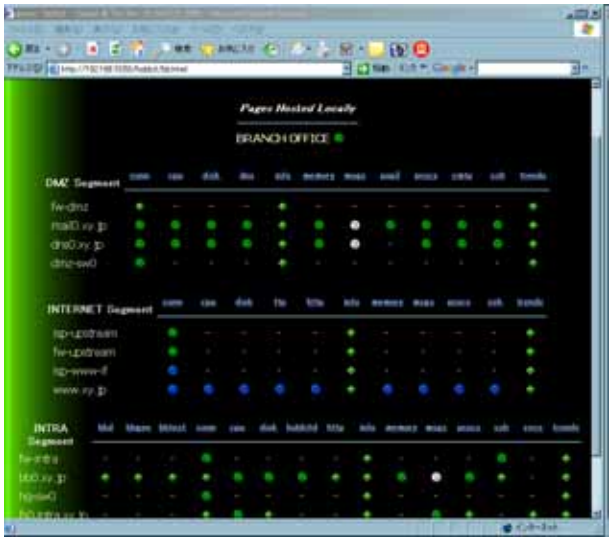
Hobbit：稼働率レポート

Hobbit：監視の一時停止機能

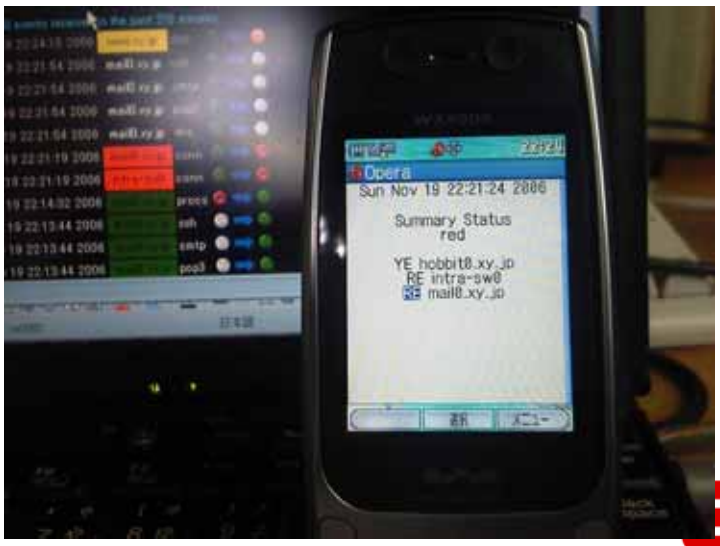
- 監視対象の保守停止といったメンテナンスの際に、一時的に対象を監視対象外にする機能
- GUIを使って簡単に監視の一時停止・開始が可能
- atコマンドと連携して、事前に監視停止・再開スケジュールを登録可能



Hobbit：監視の一時停止機能
isp-www-if/www.xy.jp監視停止中



Hobbit：WAP/WML監視画面の提供



ネットワーク監視ソフト - Nagios

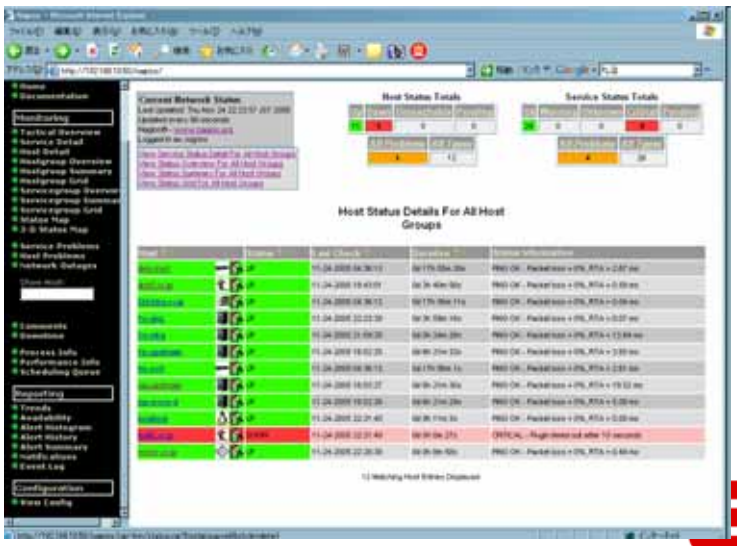
- http://www.nagios.org
- Linuxコミュニティーを中心としたネットワーク監視ソフト開発プロジェクト NetSaintがベース。マルチプラットフォームでの稼動を考慮したNagiosへと受け継がれている
- 制御パネル部と結果表示部をフレーム分割したすっきりと使いやすい画面構成
- ICMP/TCPポーリングによる監視を行う
  - 監視可能サービス：
    - `ping,tcp,udp,smtp,http、https,pop,dns,ftp,telnet,ssh,nntp, ･･･`
  - ローカルリソース監視：
    - `CPU,disk,processes,User`
- 非常に強力でかつシンプルなプラグイン機構をもっており、ユーザが簡単に独自のネットワーク試験機能を追加可能
- 拡張プラグインでIPv6対応！

ネットワーク監視ソフト - Nagios

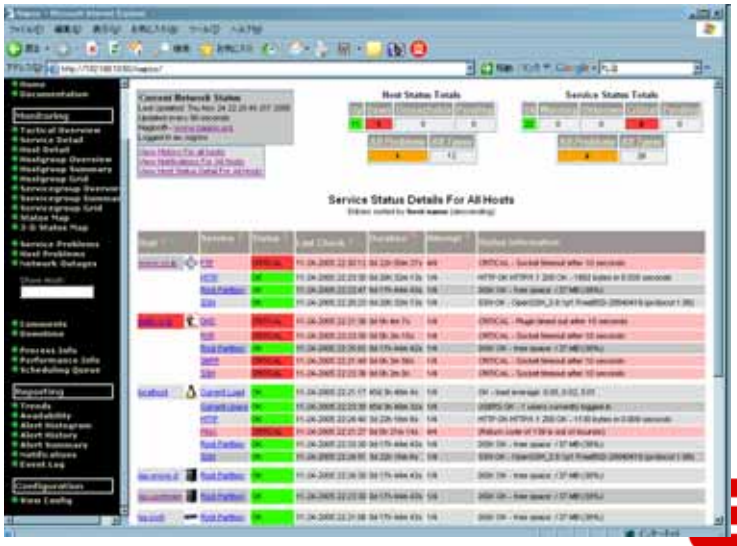
- 監視対象に対して親･子関係を定義するようになっており、ネットワークマップを自動生成可能
  - このネットワークマップに停止や未到達のホストをマッピングすることで障害ポイントの効率的割り出しが可能！
- 柔軟なアラーム通知機能
  - E-mailによりアラームを通知する
  - ホスト単位にシステムの停止時間を設定。自動で監視対象から除外可能
  - ホスト単位で障害通知先を変更可能
- アラームの検出されている機器･サービスのみのサマリ画面を生成
- ログファイルや警告、障害などの現在のネットワークの状態をブラウザ上から閲覧可能とする機能
- システム稼動状況レポート作成機能の充実

Nagios：ホストグループ一覧画面

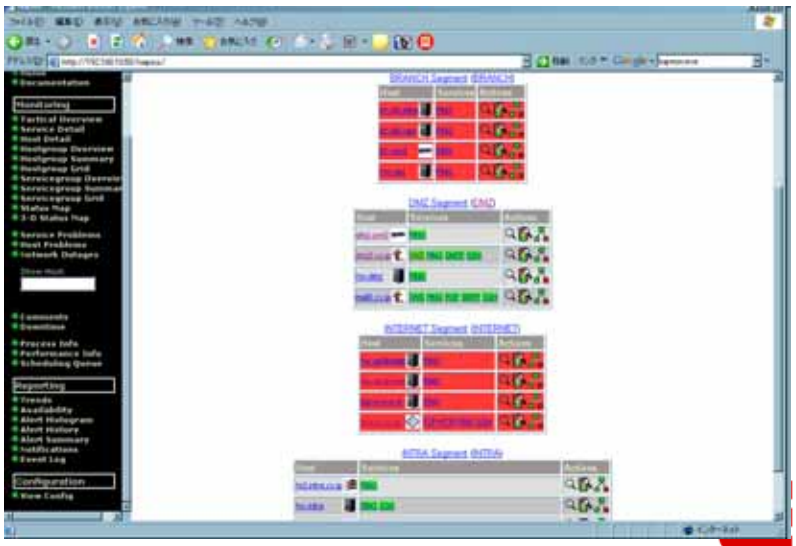
Nagios：ホスト稼動状態画面



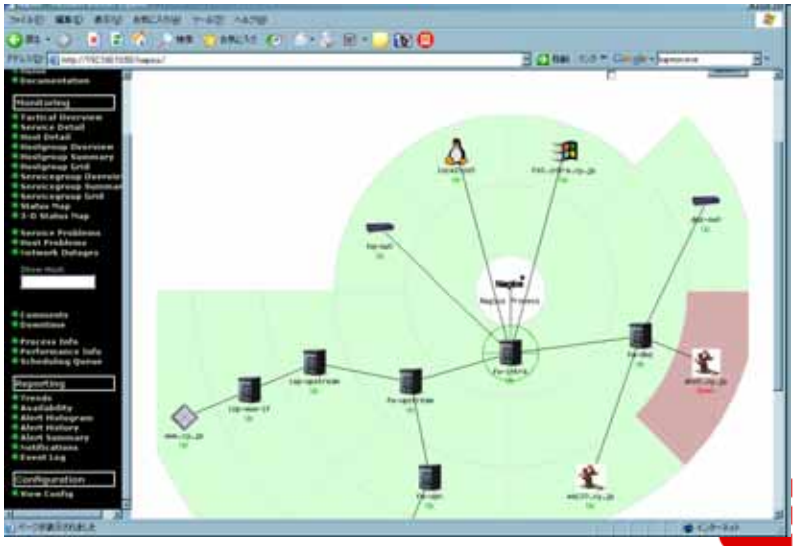
Nagios：サービス稼動状態画面



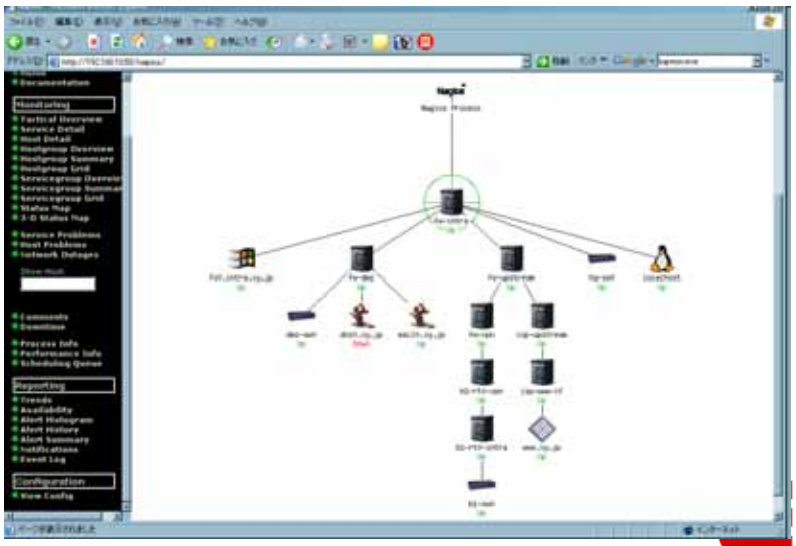
Nagios：障害サマリ



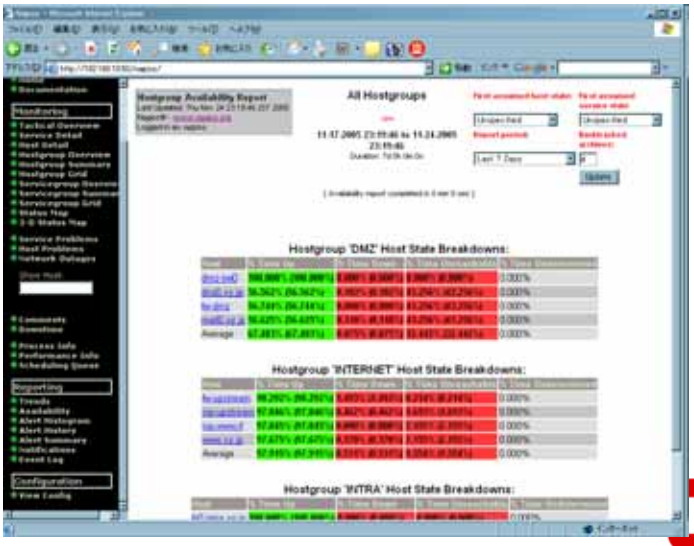
Nagios：ネットワーク稼動状況



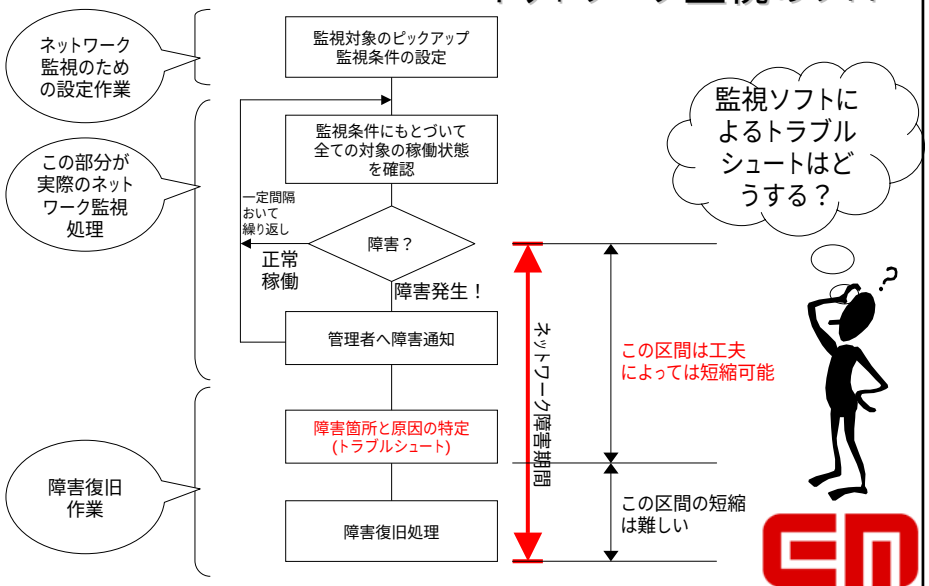
Nagios：ネットワーク稼動状況



Nagios：稼働率レポート

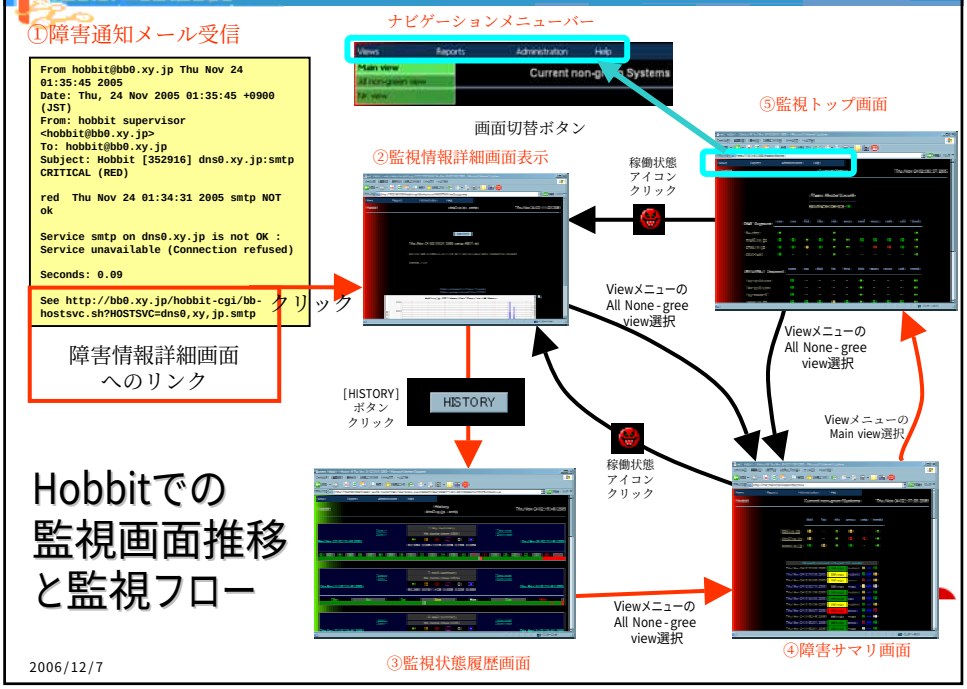


ネットワーク監視のフロー



Hobbitを使った障害トラブルシューティングフロー

① 障害通知を受信
② 障害情報詳細画面にて状況を確認
③ いつから障害が発生しているかを確認
④ 他の障害を確認
⑤ 障害を全体から把握



Hobbitでの監視画面推移と監視フロー

2006/12/7

Hobbitでの監視対象定義 etc/bb-hosts - 1

- 監視対象の定義ファイル
- 記述方法は/etc/hosts の拡張版に類似
- Hobbit/Big Brotherともに同形式のファイルフォーマットを使用
  - HobbitはLDAP/SSL試験やコンテンツ試験などの拡張指定もサポート。添付資料2を参照。
- 監視対象の記述:
  - **<IP Address> <Host Name> [# <Service> {<Service>}]**
    IP Address: 監視対象の**IP Address**
    Host Name: 監視対象のホスト名
    Service: サーバー機能及び監視サービス。

2006/12/7

Hobbitでの監視対象定義 etc/bb-hosts - 設定例

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

group-compress <H3><I>xy.jp Servers</I></H3>
192.168.0.2  ns1.xy.jp  # dns ssh !telnet
192.168.0.3  mail0.xy.jp # dns SMTP pop3 ssh !telnet
192.168.0.5  www.xy.jp  # telnet ssh ftp http://www.xy.jp/

# router interface entry
page Router-IF “Router Interface”
group-compress <H3><I>Router1 Interfaces</I></H3>
192.168.0.1  gw1.xy.jp
192.168.0.50 gw2.xy.jp
group-compress <H3><I>Router2 Interfaces</I></H3>
192.168.1.2  tok-yok-ma30.wan.xy.jp
192.168.1.6  tok-osa-dr15.wan.xy.jp
$
```


Hobbitでの監視対象定義 etc/bb-hosts - 2

- **Service**には以下のものを記述可能。
  - サーバー機能： **BBNET,BBPAGER,BBDISPLAY**
    - **BBDISPLAY**：ネットワーク監視画面サーバが動いていることを指示
    - **BBPAGER**：ネットワーク警報通知サーバが動いていることを指示
    - **BBNET**：ネットワーク監視サーバが動いていることを指示
  - **ping**監視はデフォルトで行われる。以下のアレンジも可能
    - **noping: ping**監視を行わない。監視対象外の表示はする
    - **noconn: ping**監視を行わない。表示自体も消す
    - **dialup: ping**監視結果**:NG**にて、アラームをあげない
  - 監視サービス： **smtp,http,pop3,dns,ftp,telnet,ssh,imap**
    - **http**は**URL**指定する。例： **http://www.xy.jp/top.shtml**
    - 以下のアレンジが可能。
      - **!telnet : telnet port**が開いている際に警告を行う。
        ただし、**dns/http/https**での**”!”**指定は不可
      - **~telnet : ** 試験は通常通りに行い、逆の結果を返す。
        - 例： 試験OK:赤、試験NG:緑

Hobbitでの監視対象定義 etc/bb-hosts - 3

- 画面修飾関係の設定
  - 表示グループ指定：group, group-compress
    - group(-compress) <group name>
    - この指定以下の計測対象をひとつの表示サブグループとして固めて表示する
      - group：すべての計測項目を表示する
      - group-compress：サブグループ内にて計測される項目のみ表示する
    - <group name>にはhtmlタグが使用可能
  - サブページ指定：page/subpage
    - page <page name> <page title>
    - subpage <subpage name> <subpage title>
    - この項目以下の計測対象をサブページにまとめる
    - 画面上は<page name>の項目にまとめて表示される。状態表示アイコンからサブページにリンクがはられる
    - <page title>にはhtmlタグが使用可能

bb-hosts監視指定 基本タグ一覧

監視指定	監視指定タグ	タグの意味
BBサーバ機能指定	BBDISPLAY	BB監視画面サーバ稼動指定
	BBPAGER	BB監視通知サーバ稼動指定
	BBNET	BBネットワーク監視サーバ稼動指定
PING監視指定	noping	ping監視を行わない。監視対象外の表示はする
	noconn	ping監視を行わない。表示自体も消す
	dialup	ping監視結果NGにて、アラームをあげない
監視サービス指定	ftp	ftpサービスの監視実行
	smtp	smtpサービスの監視実行
	pop3	pop3サービスの監視実行
	telnet	telnetサービスの監視実行
	ssh	sshサービスの監視実行
	nntp	nntpサービスの監視実行
	http://URL	URLに指定されたhttp実行サービスの監視
	https://URL	URLに指定されたhttpsサービスの監視実行。(lynxが必要)
	dns	dnsサービスの監視実行
	dig	dnsサービスの監視実行。digコマンドが使用可能なら同コマンドにて実施
	bbd	BBDISPLAY/BBPAGERの監視実行

監視設定

- ネットワークノードの全IPアドレスに対してPing試験を実施
- サーバについてはサービスポートの確認を行う。
  - 提供サービス確認/規制サービス確認

セグメント	IP address	ホスト名称	URL	提供サービス
本社イントラセグメント (172.16.10.0/24)	172.16.10.1	fw-intra	---	firewall
	172.16.10.10	fs.intra.xy.jp	---	FileServer
	172.16.10.50	bb0.xy.jp	bb0.xy.jp	http, ssh
	172.16.10.250	intra-sw0	---	switch
本社DMZセグメント (172.17.201.0/28)	172.17.201.17	fw-dmz	---	firewall
	172.17.201.18	mail0.xy.jp	mail0.xy.jp	dns, smtp, pop, ssh
	172.17.201.19	dns0.xy.jp	dns0.xy.jp	dns, smtp, ssh
	172.17.201.30	dmz-sw0	---	switch
本社WANセグメント (172.17.201.0/30)	172.17.201.1	isp-upstream	---	ISP-Router
	172.17.201.2	fw-upstream	---	firewall
www.xy.jpセグメント (172.18.100.128/30)	172.18.100.130	www.xy.jp	www.xy.jp	http, ftp, ssh
	172.18.100.129	www-isp-if	---	ISP-Router
本社ー支社間セグメント (192.168.250.0/24)	192.168.250.1	fw-vpn	---	firewall
	192.168.250.2	b1-rt0-vpn	---	router
支社イントラセグメント (172.16.100.0/24)	172.16.100.1	b1-rt0-intra	---	firewall
	172.16.100.250	b1-sw0	---	switch

監視ノードの整理表

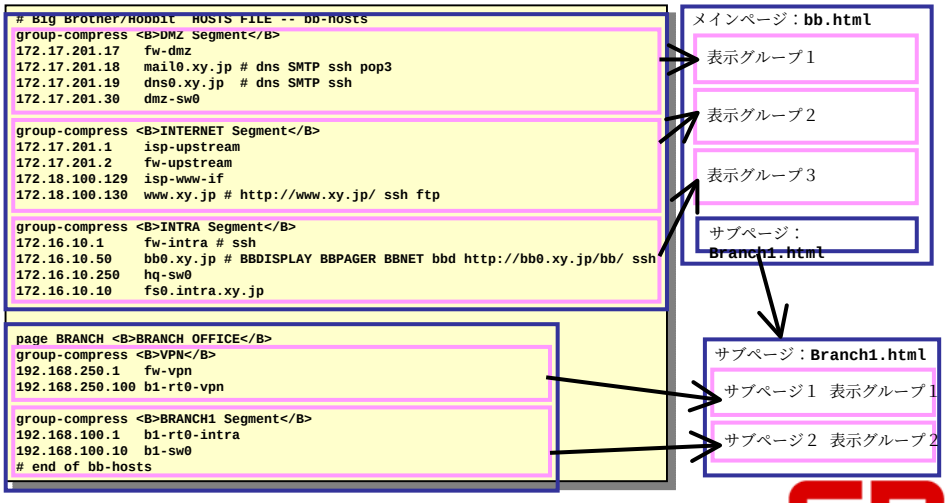
所属ページ	所属group	IP address	ホスト名称	BBServer	提供サービス	提供サービス(http)	規制サービス
top-page	INTRA segment	172.16.10.1	fw-intra				
		172.16.10.10	fs.intra.xy.jp				telnet smtp
		172.16.10.50	bb.xy.jp	BBDISPLAY BBNET BBPAGER	ssh	http://bb.xy.jp/bb	telnet smtp
		172.16.10.250	intra-sw0				
	DMZ segment	172.17.201.17	fw-dmz				
		172.17.201.18	mail.xy.jp		dns smtp pop ssh		telnet
		172.17.201.19	dns.xy.jp		dns smtp ssh		telnet
		172.17.201.30	dmz-sw0				
	Internet segment	172.17.201.1	isp-upstream				
		172.17.201.2	fw-upstream				
	www-hosting segment	172.18.100.130	www.xy.jp		ftp ssh	http://www.xy.jp	telnet smtp
		172.18.100.129	www-isp-if				
sub-page Branch1	VPN segment	192.168.250.1	fw-vpn				
		192.168.250.2	b1-rt0-vpn				
	Branch1 segment	172.16.100.1	b1-rt0-intra				
		172.16.100.250	b1-sw0				

Hobbitでの監視対象設定 etc/bb-hosts

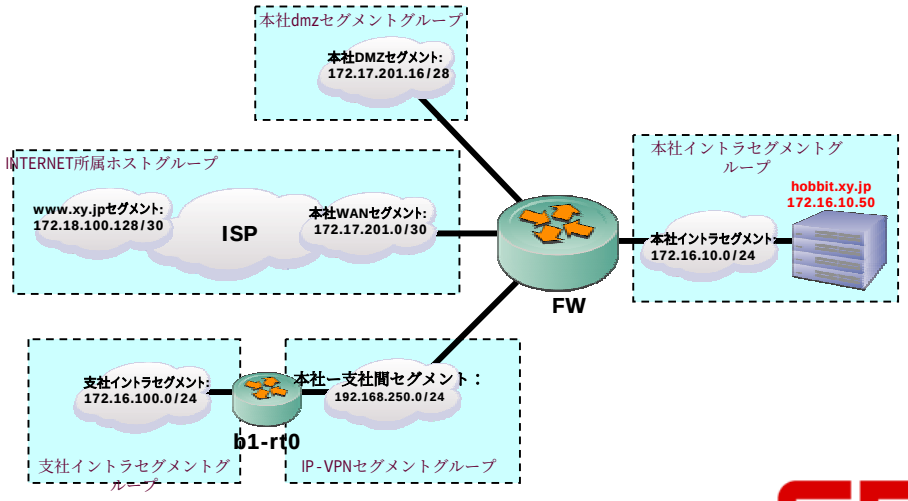
```
# Big Brother/Hobbit  HOSTS FILE -- bb-hosts
group-compress <B>DMZ Segment</B>
172.17.201.17   fw-dmz
172.17.201.18   mail0.xy.jp # dns SMTP ssh pop3
172.17.201.19   dns0.xy.jp  # dns SMTP ssh
172.17.201.30   dmz-sw0

group-compress <B>INTERNET Segment</B>
172.17.201.1    isp-upstream
172.17.201.2    fw-upstream
172.18.100.129  isp-www-if
172.18.100.130  www.xy.jp # http://www.xy.jp/ ssh ftp

group-compress <B>INTRA Segment</B>
172.16.10.1     fw-intra # ssh
172.16.10.50    bb0.xy.jp # BBDISPLAY BBPAGER BBNET bbd http://bb0.xy.jp/bb/ ssh
172.16.10.250   hq-sw0
172.16.10.10    fs0.intra.xy.jp

page BRANCH <B>BRANCH OFFICE</B>
group-compress <B>VPN</B>
192.168.250.1   fw-vpn
192.168.250.100 b1-rt0-vpn

group-compress <B>BRANCH1 Segment</B>
192.168.100.1   b1-rt0-intra
192.168.100.10  b1-sw0
# end of bb-hosts
```


bb-hostsの設定とページ構成

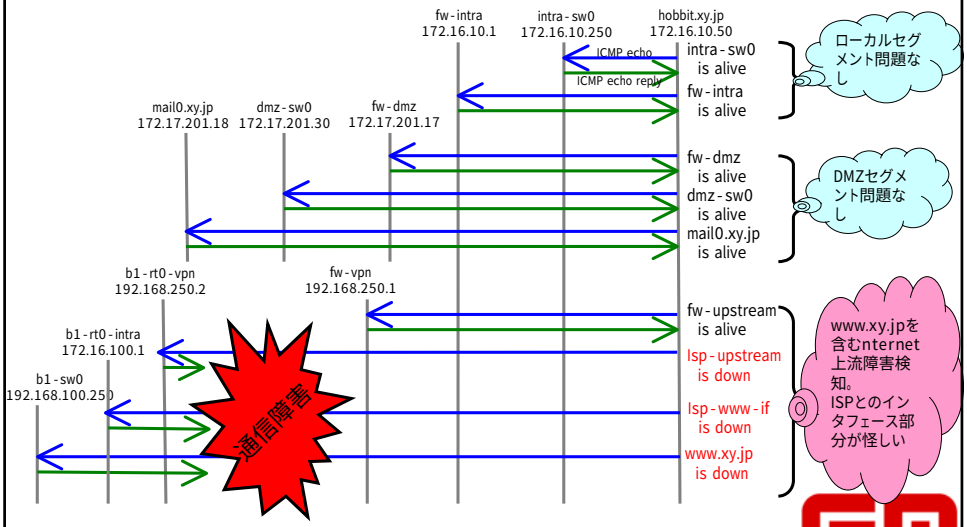
bb-hosts



各グループのトポロジー整理



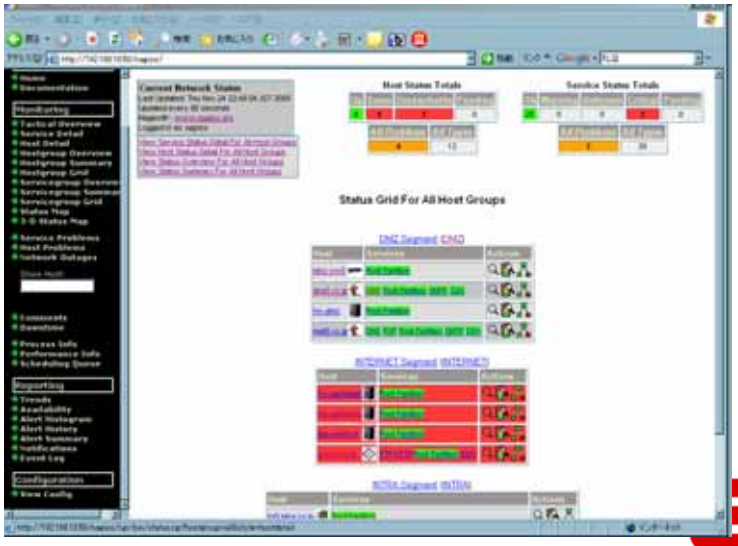
セグメントグルーピングによる障害点の把握



Hobbit/BigBrotherによる障害点の把握

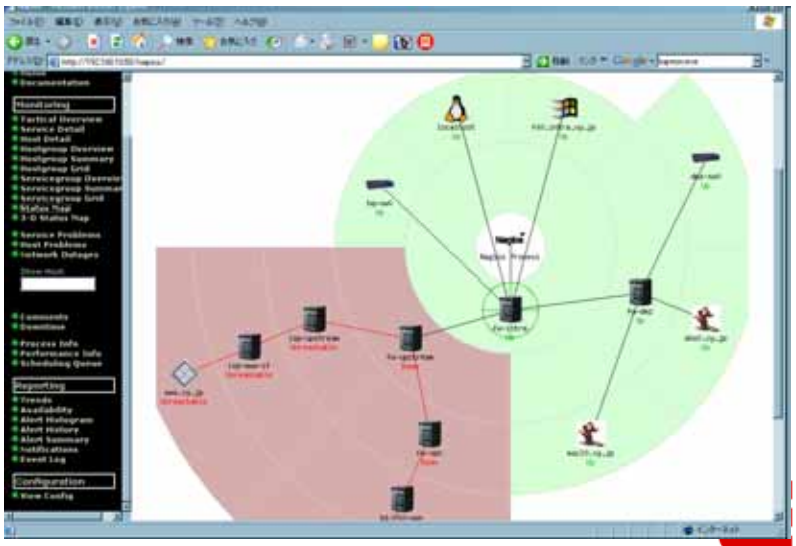
2006/12/7

Nagiosによる障害点の把握



2006/12/7

Nagiosによる障害点の把握：ネットワーク表示



Hobbit警報通知定義：etc/hobbit-alert.cfg

- 警告通知に対するルールを記述する
- 記述方法：
  - HOST=<HOST名>
    MAIL <送付先> <OPTIONS>
  - <HOST名>：イベント対象のホスト名称。"%"を先頭に置くことで正規表現形式指定も可能
  - <送付先>：障害通知を行うメールアドレスを指定
  - <OPTIONS>：送付方法に対するオプション指定。
    - SERVICE=＜SERVICE>：<SERVICE>にマッチするサービス障害の場合通知
    - REPEAT=<TIME>：<TIME>分間隔で繰り返して障害通知を行う。
    - DURATION{<|>}<TIME>：<TIME>分障害未満もしくは以上継続した場合に障害通知を行う
    - COLOR=<COLOR>：<COLOR>の場合に通知を行う。デフォルトは"red"。この他、"yellow", "purple"も指定可能。
    - RECOVERED：対象が障害回復した場合に通知する
    - その他、いろいろとあり。
  - 例：HOST=www.foo.com SERVICE=http
    MAIL webadmin@foo.com REPEAT=20 RECOVERED
    MAIL cio@foo.com DURATION>60 COLOR=red

監視対象分析 – 監視時間と通知先

- 全ての機器の障害情報は障害受付窓口であるalert@xy.jpに通知
- 独自のイントラ系と支社ネットワークの部分については以下の監視･障害通知ポリシーを適用
  - 本社ファイルサーバ fs.intra.xy.jp ：
    - 毎日午前4時から6時の間でデイリーバッチ処理が走り、高負荷となることから監視を停止。監視省力化
    - この機械の障害時には担当窓口：intra@xy.jpにも通知
  - 支社機器の障害対応は現地の担当に任せることが多いためにalert@branch.xy.jpへの通知を追加

警報通知定義

セグメント	IP address	ホスト名称	提供サービス	通知先	通知時間
本社イントラセグメント (172.16.10.0/24)	172.16.10.1	fw-intra	firewall	alert@xy.jp	24/7D
	172.16.10.10	fs.intra.xy.jp	FileServer	alert@xy.jp	24/7D, 午前4-5時台は除外
	172.16.10.50	bb0.xy.jp	http ssh	alert@xy.jp	24/7D
	172.16.10.250	intra-sw0	switch	alert@xy.jp	24/7D
本社DMZセグメント (172.17.201.0/28)	172.17.201.17	fw-dmz	firewall	alert@xy.jp	24/7D
	172.17.201.18	mail0.xy.jp	dns smtp pop ssh	alert@xy.jp	24/7D
	172.17.201.19	dns0.xy.jp	dns smtp ssh	alert@xy.jp	24/7D
	172.17.201.30	dmz-sw0	switch	alert@xy.jp	24/7D
本社WANセグメント (172172O1.0/30)	172.17.201.1	isp-upstream	ISP-Router	alert@xy.jp	24/7D
	172.17.201.2	fw-upstream	firewall	alert@xy.jp	24/7D
www.xy.jpセグメント (172.18.100.128/30)	172.18.100.130	www.xy.jp	http ftp ssh	alert@xy.jp	24/7D
	172.18.100.129	www-isp-if	ISP-Router	alert@xy.jp	24/7D
本社ー支社間セグメント (192.168.250.0/24)	192.168.250.1	fw-vpn	firewall	alert@xy.jp	24/7D
	192.168.250.2	b1-rt0-vpn	router	alert@xy.jp alert@branch.xy.jp	24/7D
支社イントラセグメント (172.16.100.0/24)	172.16.100.1	b1-rt0-intra	firewall	alert@xy.jp alert@branch.xy.jp	24/7D
	172.16.100.250	b1-sw0	switch	alert@xy.jp alert@branch.xy.jp	24/7D

警報通知定義 etc/hobbit-alert.cfg

```
[hobbit@bb0.xy.jp /home/hobbit/server/etc]$ cat hobbit-alert.cfg
### hobbit-alert.cfg

HOST=%^fs.*
    MAIL alert@xy.jp TIME=*:0000-0359:*:0600-2359
## fs*(fs.intra.xy.jpにマッチ)については24H/7Dの監視を行い、
## 障害時はaltert@xy.jpとintra@xy.jpに通知する
## ただし、AM4:00-AM5:59までの間は通知対象外とする

HOST=%^b1-.*
    MAIL alert@xy.jp
    MAIL alert@branch.xy.jp
## b1-*(支社の装置)については24H/7Dの監視を行い、
## 障害時はaltert@xy.jpとintra@xy.jpに通知

HOST=%^.*
     MAIL yahagi@eaccess.net COLOR=red REPEAT=30m
     MAIL hobbit COLOR=red REPEAT=30m
## その他の障害はすべてhobbit@bb0.xy.jpに行う
### end of hobbit-alert.cfg
$
```


警報通知定義 etc/hobbit-alert.cfg 外部スクリプトの組み込み

- 警告通知に対するルールを記述する
- 記述方法:
  - HOST=<HOST名>
    SCRIPT <外部スクリプト> <スクリプト用パラメータ> <OPTIONS>
  - <外部スクリプト>：障害通知を行うメールアドレスを指定
  - <スクリプト用パラメータ> ：外部スクリプトに引き渡す
  - <OPTIONS>：送付方法に対するオプション指定。MAIL指定と同様のものを指定可能
  - 例：
    HOST=%^www.*
    MAIL webadmin@foo.com REPEAT=20 RECOVERED
    SCRIPT /usr/home/hobbit/server/ext/mail-to-keitai.sh xyadm1n@docomo.ne.jp RECOVERED

```
From:    hobbit supervisor <hobbit@hobbit0.xy.jp>
To:      xyadm1n@docomo.ne.jp
Date:    Sun, 19 Nov 2006 21:46:40 +0900 (JST)
Subject: RECOVERED red : www.xy.jp-procs
----
Trouble is recovered! please look up the following URL.

RECOVERED : red : www.xy.jp-procs
Time  : 61sec

REF: http://cw9b39k7.corede.net/hobbit/wml/
```

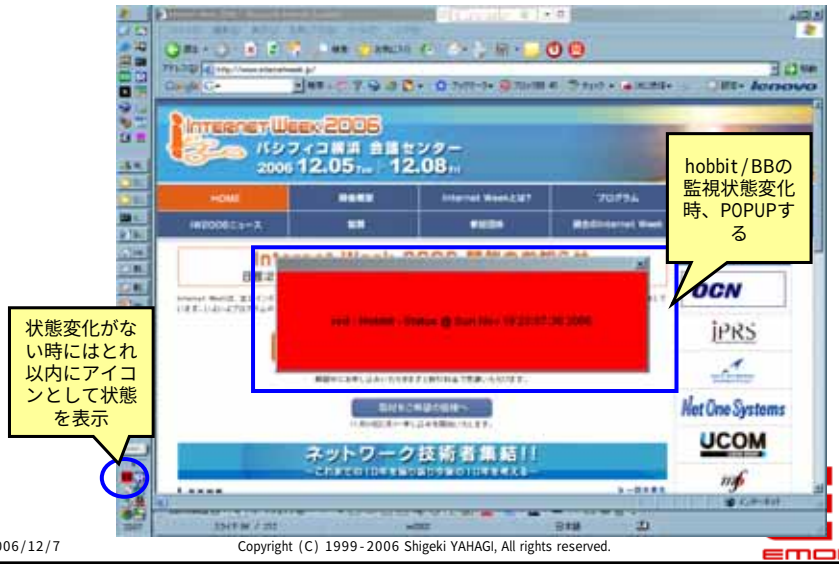
hobbit-alerts.cfg用拡張スクリプト例：mail-to-keitai.sh：携帯用WAP URL通知スクリプト

```
#!/bin/sh
### mail-to-keitai.sh <recipitents>

case "${RECOVERED}" in
   "0")
        STATUS="ALERT"
        MSG="Something happen!"
        ;;
   "1")
        STATUS="RECOVERED"
        MSG="Trouble is recovered!"
        ;;
esac

/usr/bin/mail -s "${STATUS} ${BBCOLORLEVEL} : ${BBHOSTNAME}-${BBSVCNAME}" ${RCPT} << EOF
$MSG please look up the following URL.

${STATUS} : ${BBCOLORLEVEL} : ${BBHOSTNAME}-${BBSVCNAME}
Time  : ${DOWNSECS}sec

REF: http://${BBSERVERWWWNAME}${BBWAP}/
EOF

### end of command
```


BigBrother/Hobbit機能拡張ツール BBTray - 監視サポートツール

- BigBrother/Hobbit DisplayServerを常時監視するサポートツール
  - **http://www.deadcat.net/viewfile.php?fileid=233**
  - Windows9x/NT/2000/XPで動作
  - BBを監視し、状態が変化すると音とPopup Windowにて通知
  - Windowをクリックすることで、障害サマリー画面に直接とべるので、即時に現状把握可能
    - BBサーバーとIP通信ができれば、どこでも現状が分かる
  - 類似品にtkBB(Tk-Perl版)あり

BB/Hobbit機能拡張ツール
BBTray - 監視サポートツール



BB/Hobbit機能拡張ツール
BBTray - 監視サポートツール

Green Window
- this is normal status

Yellow Window
- this is warning status.

Red Window
- this is critical status!!

機能拡張スクリプト BBTrayのコンフィグ

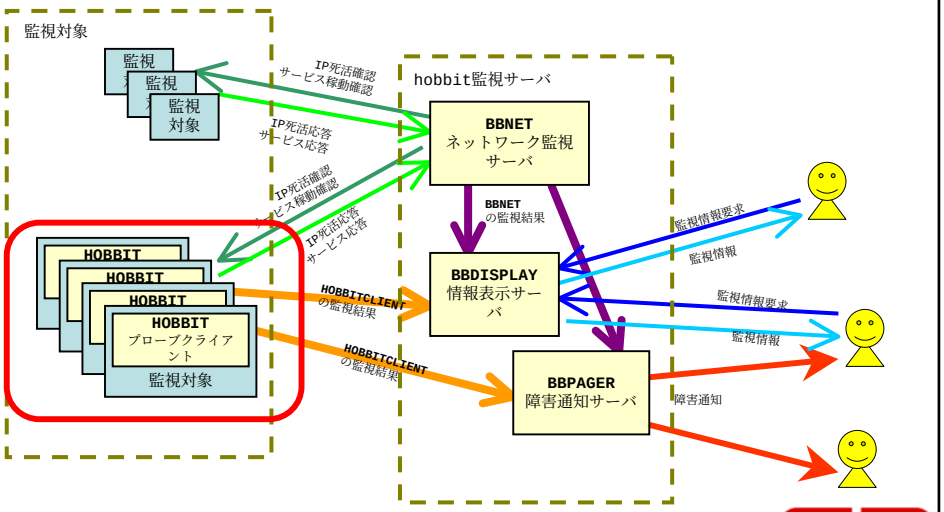
```
; BBTRAY.INI - BBTray Configuration File
;   This file must be in the same directory as the BBTRAY.EXE.
;   Changes will only take effect on restart of BBTray
;---------------------------------------------------------------------------
; Default options

[General]
DisplayURL=http://bb0.xy.jp/hobbit/bb2.html
SoundsPath=C:¥Program Files¥BBtray¥Sounds¥
IconsPath=C:Program Files¥¥BBtray¥Icons¥
;ProxyName=192.168.0.200:3128
PollFrequency=15
PageDelay=900
PopupLevels=r,p,y,g

; String for tray icon's hint and pop-up window. Can include the following
; fields identifiers:
;   %U   BBDISPLAY URL
;   %T   BBDISPLAY title
;   %c   color letter (ex: 'g' for 'green')
;   %C   color string
;   ¥n   NewLine
; For the old URLOnHint format, use HintString=%C: %U
; OBS: Max HintString size is 63 chars.
HintString=My Servers: %T
PopupString=My Servers: %U¥n%T

;---------------------------------------------------------------------------
; These are the messages displayed by BBTray
[Messages]
VERIFY=Verifying...
NOCONN=It was not possible to connect to the monitoring system!
INVSTATUS=Invalid status received!
```


INDEX

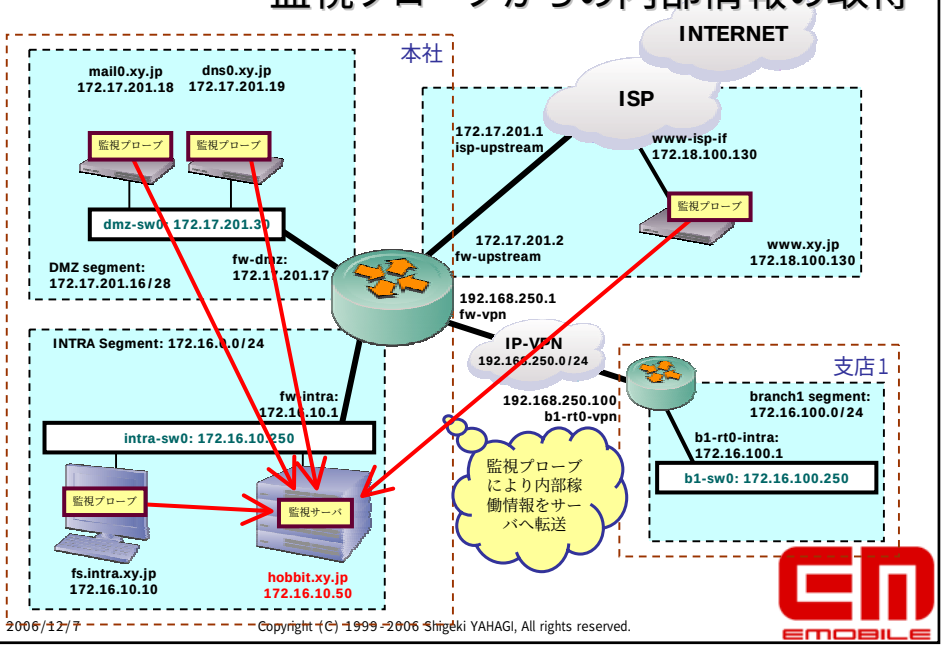
監視プローブによるリソース監視

- Hobbit/Big Brother/Nagios監視サーバー単体では、各監視対象のIP死活監視とサービス稼働監視までが限界
- より踏み込んで、監視対象の稼働状態を把握するためには、監視対象にて自身のリソースをチェックするプローブクライアントのインストールが必要となる
  - Hobbit/BBでは標準でプローブクライアントを添付
  - NagiosではNPREP/NRPE/NSClientなどの外部拡張にて機能を実現
- 監視プローブで可能となるリソース監視：
  - CPU使用率監視
  - プロセス稼働監視
  - ディスク容量監視
  - 自律メッセージ監視（BBCLIENTでのみサポート）
  - などなど

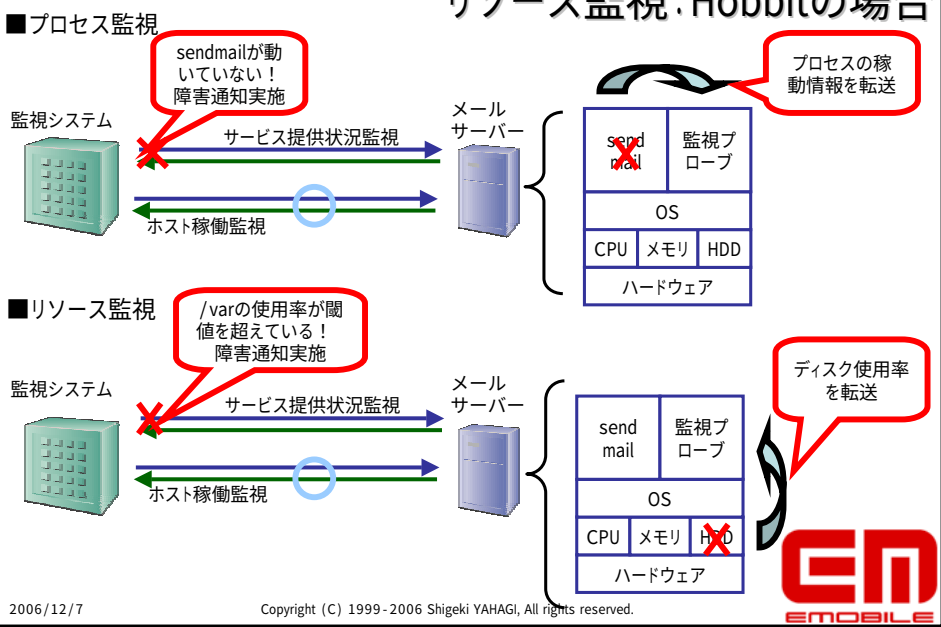
Hobbit監視プローブによるリソース監視



監視プローブからの内部情報の取得



リソース監視：Hobbitの場合

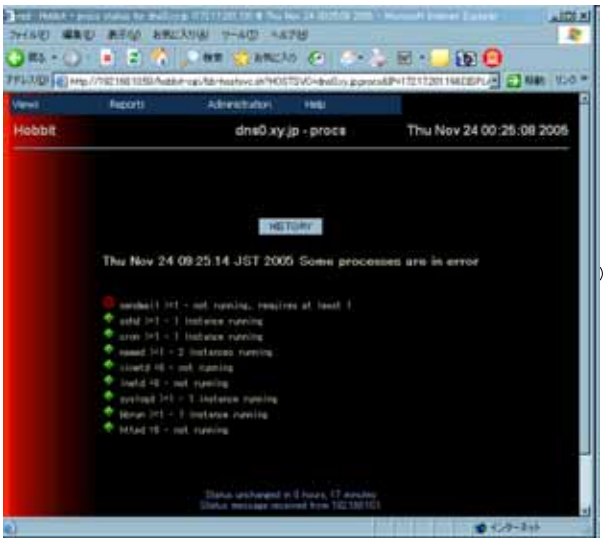


プローブクライアントを追加した監視項目

- dns0.xy.jpで障害発生。IP死活監視はOK。
- サービス監視でSMTPのみ障害でプロセス監視でも障害がでている他のサービスでは警告があがってないので、SMTP回りの障害と判断
- DISK使用率監視で注意がでているが、注意レベルなので後回しにして、まずはプロセス監視の詳細情報チェックする！

2006/12/7

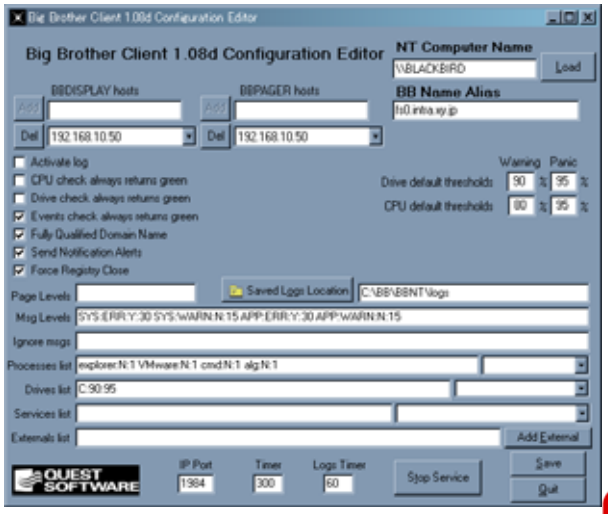
プロセス監視画面



sendmailがなぜか落ちている！これが原因らしい。

2006/12/7

Windows版BBCLIENT



Hobbit:プローブクライアント閾値の設定

- Hobbitにおいてはプローブクライアントは情報を転送するだけ
- 結果判定条件はすべてHobbitサーバに設定する
- 設定ファイル：etc/hobbit-client.cfg
  - CPU使用率監視
  - UPTIME検査
  - メモリ監視指定
  - プロセス監視
  - ディスク使用率監視

[hobbitclient.cfgの例]

```
[hobbit@bb0.xy.jp ~/server/etc]$ cat hobbitclient.cfg
# the generic rules last.

UP      1h
LOAD    5.0 10.0

DISK / 70 85
DISK /var 50 70
DISK /dev 101 101
DISK /usr/compat/linux/proc  101 101
DISK /usr 75 85

MEMPHYS 100 101
MEMSWAP 50 80
MEMACT  90 97

PROC syslogd 1 1 yellow
PROC cron
PROC sshd

HOST=bb0.xy.jp
    PROC inetd 0 0 yellow
    PROC xinetd 0 0 yellow
    PROC httpd 5 20 red
```

Hobbitプローブクライアント閾値の設定：CPU使用率監視/UPTIME監視

- CPU使用率監視
- load averageを元にCPU使用率を監視
- レコードフォーマット1： デフォルト指定
  - LOAD <WARN値> <PANIC値>
    - <WARN値> - Load average warningレベル設定値
    - <PANIC値> - Load average panicレベル設定値
  - 使用例： LOAD 5 10
  - デフォルト値：<WARN値>=5, <PANIC値>=10
- レコードフォーマット2：個別ホスト指定
  - HOST=<host IP/URL>
    LOAD <WARN値> <PANIC値>
  - 使用例：HOST=www.xy.jp
    LOAD 10 20
- UPTIME監視
- システムのUPTIMEを監視する
- レコードフォーマット1： デフォルト指定
  - UP <bootlimit値> <toologlimit値>
    - <bootlimit値> ： ブート発生かいつまでを異常とするかの時間。h/d/wを追加することで時間/日/週の規定が可能
    - <toolonglimit値> ： システム稼動最大時間を規定する値
  - 使用例： UP 1d
- レコードフォーマット2：個別ホスト指定
  - HOST=<host IP/URL>
    UP <bootlimit値> <toologlimit値>
  - 使用例：HOST=www.xy.jp
    UP 1h 50w

Hobbitプローブクライアント閾値の設定：メモリ使用監視

- 物理メモリ・仮想メモリ・スワップなどの使用率を検査
  - MEMPHYS： 物理メモリの使用率
  - MEMACT： 実際に使用しているメモリの使用率
  - MEMSWAP： スワップの使用率
- レコードフォーマット1：デフォルト指定
  - MEMPHYS <WARN値> <PANIC値>
  - MEMACT <WARN値> <PANIC値>
  - MEMSWAP <WARN値> <PANIC値>
    - WARN値：警告を行う使用率をパーセントで指定
    - PANIC値：障害としてアラーム周知を行う使用率をパーセントで指定
- レコードフォーマット2：ホスト指定
  - 本指定はCPU使用率検査・UPTIME検査のフォーマット2と同様となる
- デフォルト値
  - MEMPHYS 100 101
    - この場合<PANIC値>が100を超えて指定されており、警報をださないという指定となっている
  - MEMACT 90 97
  - MEMSWAP 50 80

Hobbitプローブクライアント閾値の設定：プロセス稼動監視

- 監視対象のプロセスの存在を監視する
- レコードフォーマット1：デフォルト指定
  - PROC <PROC名> <最少起動数> <最大起動数> <COLOR値>
    - <PROC名>：レコードの対象となるホストの名称を指定
    - <最少起動数>：プロセスの最小起動数を指定。デフォルト=1
    - <最大起動数>：プロセスの最大起動数を指定。デフォルト=1
    - <COLOR値>：<最少起動数>と<最大起動数>から当該プロセス数がはずれた際に返す警報色を"yellow", "red"のいずれかで指定。デフォルト="red"
    - 非起動確認についてもサポートしており、その際には<最少起動数>、<最大起動数>ともに"0"を指定する。
    - セキュリティー上あがっているとまずいプロセスの監視に使用
      - ex: PROC inetd 0 0 yellow
- レコードフォーマット2：ホスト指定
  - 本指定はCPU使用率検査･UPTIME検査のフォーマット2と同様となる

[プロセス稼動監視指定の例]

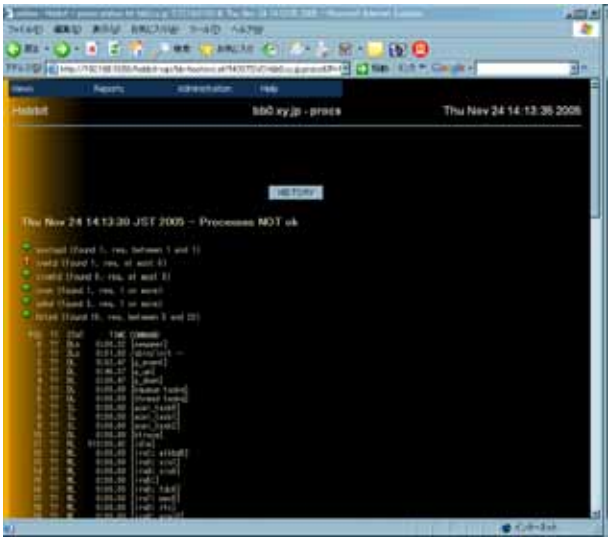
レコードフォーマット1：全体対象に共通のプロセス稼動監視指定

```
PROC syslogd 1 1 yellow
PROC cron
PROC sshd
```

レコードフォーマット2：bb0.xy.jpだけに適用するプロセス稼動監視指定

```
HOST=bb0.xy.jp
    PROC inetd 0 0 yellow
    PROC xinetd 0 0 yellow
    PROC httpd 5 20 red
```


Hobbit - プロセス監視



Hobbitプローブクライアント閾値の設定：ディスク使用率監視

- 監視対象のディスク使用率を監視する
- レコードフォーマット1：デフォルト指定
  - DISK <パティション名> <WARN値> <PANIC値>
    - <パティション名>：監視対象とするパティションを指定
    - <WARN値>：警告を行うディスク使用率をパーセントで指定
    - <PANIC値>：障害としてアラーム周知を行うディスク使用率をパーセントで指定
      - /procや/devのように監視する必要のないパティションにおいては１００以上の値を指定する
- レコードフォーマット２：ホスト指定
  - 本指定はCPU使用率検査･UPTIME検査のフォーマット２と同様となる

【プロセス稼動監視指定の例】

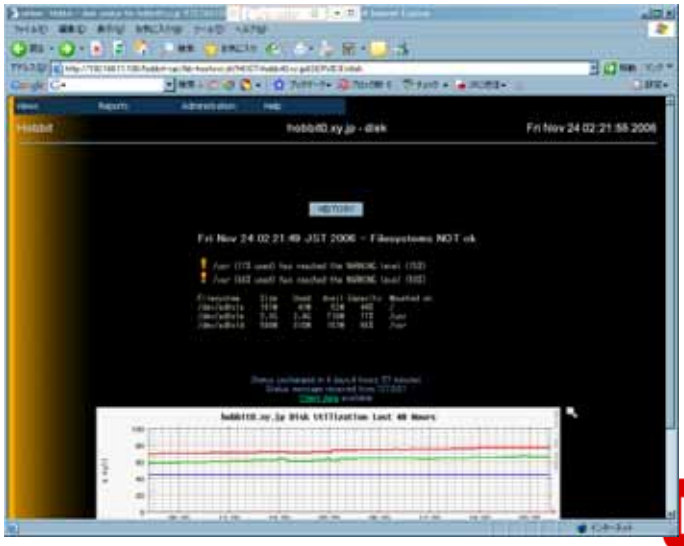
レコードフォーマット1：全体対象に共通のディスク使用率監視指定

```
DISK / 70 85
DISK /dev 101 101
DISK /usr/compat/linux/proc  101 101
```

レコードフォーマット2：bb0.xy.jpだけに適用するディスク使用率監視指定

```
HOST=bb0.xy.jp
     DISK /var 50 70
     DISK /usr 75 85
```


Hobbitディスク使用率監視画面



hobbitclient.cfgの例

```
hobbit@bb0.xy.jp:/usr/home/hobbit/server/etc[505]$
### hobbitclient.cfg

UP      1h
LOAD    5.0 10.0

DISK / 70 85
DISK /var 50 70
DISK /dev 101 101
DISK /usr/compat/linux/proc  101 101
DISK /usr 75 85

SWAP    20 40
MEMPHYS 100 101
MEMSWAP 50 80
MEMACT  90 97

PROC syslogd 1 1 yellow
PROC cron
PROC sshd

HOST=bb0.xy.jp
        PROC inetd 0 0 yellow
        PROC xinetd 0 0 yellow
        PROC httpd 5 20 red

HOST=mail0.xy.jp
        PROC inetd
        PROC sendmail 2
        PROC httpd 0 0 yellow

HOST=dns0.xy.jp
        PROC inetd 0 0 yellow
        PROC xinetd 0 0 yellow
        PROC sendmail
        PROC httpd 0 0 yellow

HOST=www.xy.jp
        PROC inetd 0 0 yellow
        PROC xinetd 0 0 yellow
        PROC sendmail 0 0 yellow
        PROC httpd 5 20 red

### end of file
hobbit@bb0.xy.jp:/usr/home/hobbit/server/etc[505]$
```


リソース監視 経過監視：RRDtool連携によるグラフ化機能

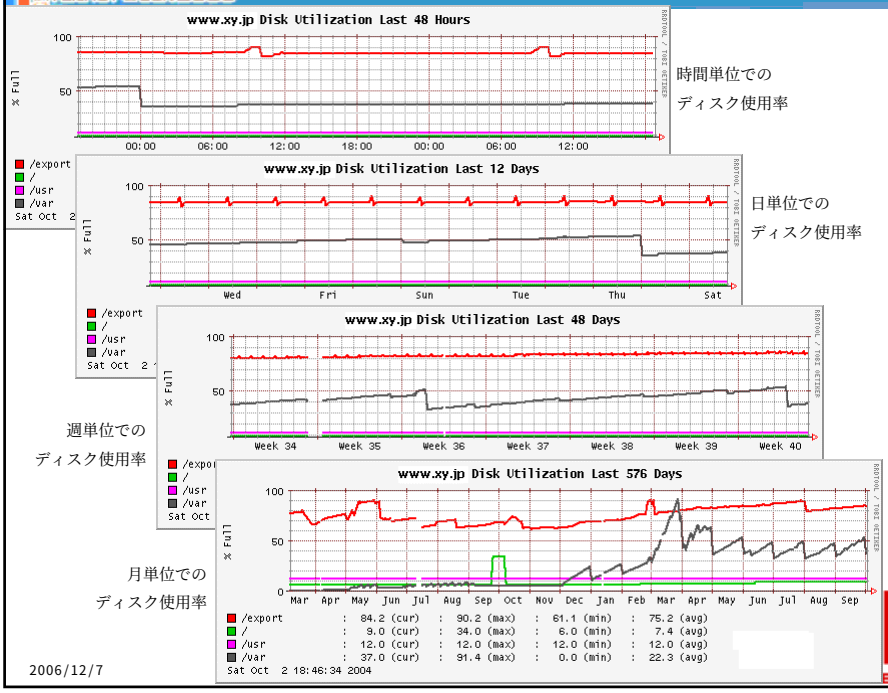
- ネットワーク監視ソフトは定期的に監視対象の稼動状況や、レスポンスなどのデータを収集することで、ネットワークの稼動状況を分析・提示する
- サービス応答時間やディスク・CPU・メモリ使用率といった監視項目をグラフ化できれば経過監視が可能となるはず。

- グラフ化ツールとの連携により経過監視による障害発生予測も可能となる
  - RRDtoolにより日次・週次・月次などのスパンでリソースの使用状況を把握できるので、障害発生予測などに有効
    - リソース不足障害の予測
    - 繰り返す障害の発生パターン
- Hobbit/BB+LARRDは各監視対象のプローブクライアントから取得したデータをRRDToolにより自動的にグラフ化する
  - グラフ化対象データ：Load Average, Disk Usage, Memory使用率, SWAP, プロセス数, TCP Connection Time, ネットワーク使用率, ...

システムリソース管理
Hobbit Trends画面

2006/12/7

disk使用率の経過監視の例



2006/12/7

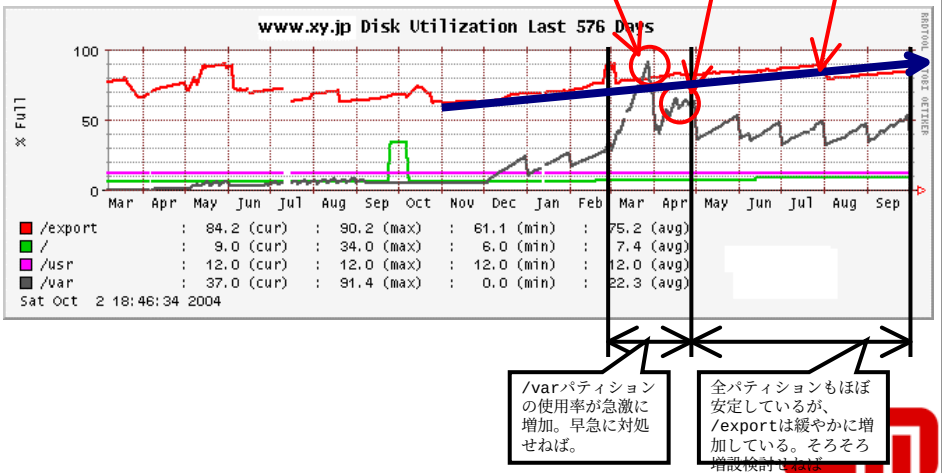
disk使用率の経過監視の例

急激なディスク使用率増加対処
ファイル配置の最適化と圧縮

ディスク増設
による本対処

緩やかな
増加傾向

図1-d 月単位でのディスク使用率

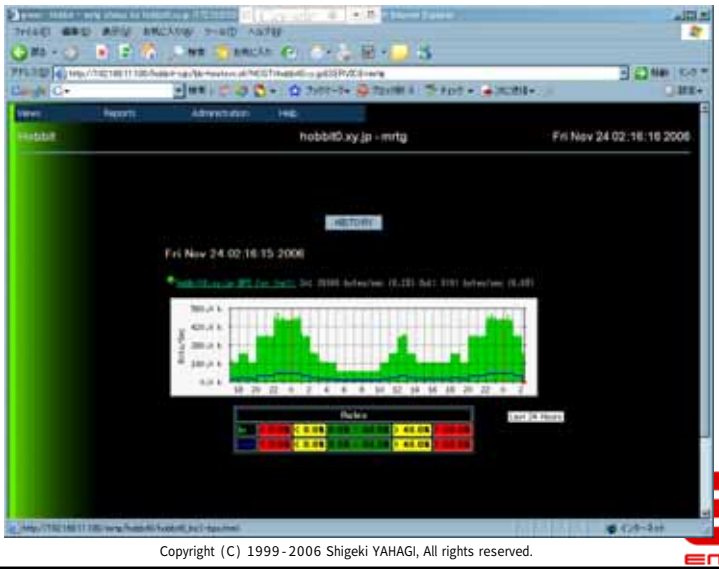


/varパティションの使用率が急激に増加。早急に対処せねば。

全パティションもほぼ安定しているが、/exportは緩やかに増加している。そろそろ増設検討せねば。

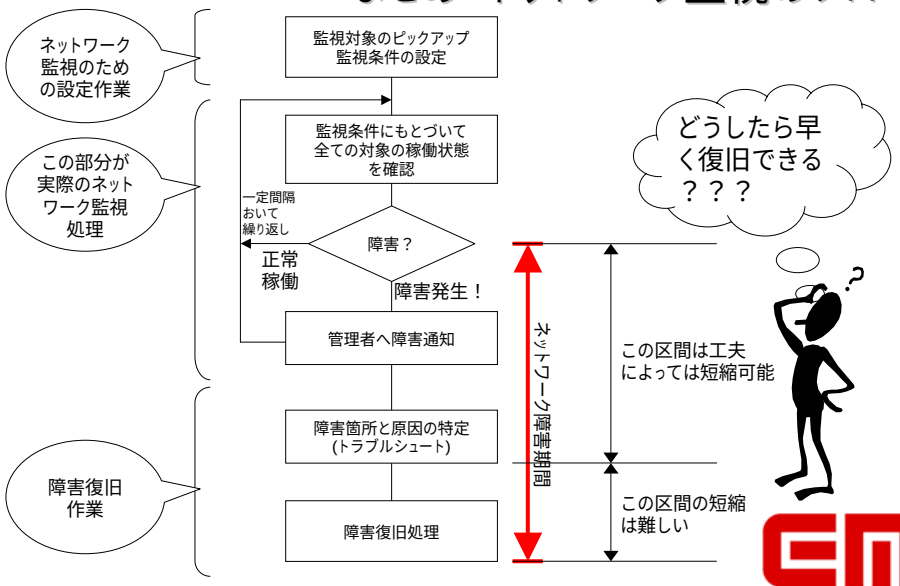
2006/12/7

トラフィックリソース監視
Hobbit/BBとMRTGの連携：bbmrtg.pl



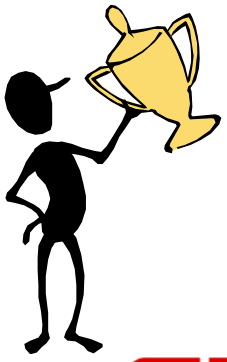
2006/12/7

まとめ：ネットワーク監視のフロー



まとめ：ネットワーク障害を早期復旧させるには

- 漏れのない稼働状態確認
- 的確な障害通知
- 迅速な障害原因調査開始
- 迅速な障害原因の特定
- 障害復旧のための準備



追加資料１：
Nagiosの設定

Nagiosの設定手順

① contactの定義
- 障害通知先の定義を行う

② contactgroupの定義
- 障害通知先グループの定義を

③ hostの定義
- ホスト情報の定義を行なう

④ hostgroupの定義
- 同じセグメントなど関係のあるホストをグループとして定義する

⑤ serviceの定義
- 監視サービスと対象ホストを定義する

⑥ servicegroupの定義（オプション）
- 監視サービスをグルーピングするサービスグループを定義する

⑦ hostextinfoの定義（オプション）
- ホストに付加するICONなど付加情報を定義する

Nagiosの設定：情報整理

セグメント	host定義			hostgroup定義	service定義						
	host	IP address	Parent		PING	ssh	dns	smtp	pop3	http	ftp
本社イントラセグメント (172.16.10.0/24)	fw-intra	172.16.10.1	fw-intra	intra segment	○						
	fs0.intra.xy.jp	172.16.10.10	fw-intra	intra segment	○						
	bb0.xy.jp	172.16.10.50	fw-intra	intra segment	○	○				○	
	intra-sw0	172.16.10.250	fw-intra	intra segment	○						
本社DMZセグメント (172.17.201.0/28)	fw-dmz	172.17.201.17	fw-intra	dmz segment	○						
	mail0.xy.jp	172.17.201.18	dmz-sw0	dmz segment	○	○	○	○	○		
	dns0.xy.jp	172.17.201.19	dmz-sw0	dmz segment	○	○	○	○			
	dmz-sw0	172.17.201.30	fw-dmz	dmz segment	○						
本社WANセグメント (17217201.0/30)	isp-upstream	172.17.201.1	fw-upstream	internet segment	○						
	fw-upstream	172.17.201.2	fw-intra	internet segment	○						
www.xy.jpセグメント (172.18.100.128/30)	www.xy.jp	172.18.100.130	www-isp-if	internet segment	○	○				○	○
	www-isp-if	172.18.100.129	isp-upstream	internet segment	○						
本社ー支社間セグメント (192.168.250.0/24)	fw-vpn	192.168.250.1	fw-intra	vpn segment	○						
	b1-rt0-vpn	192.168.250.2	fw-vpn	vpn segment	○						
支社イントラセグメント (172.16.100.0/24)	b1-rt0-intra	172.16.100.1	b1-rt0-vpn	vpn segment	○						
	b1-sw0	172.16.100.250	b1-rt0-intra	vpn segment	○						

Nagiosの設定：contactとcontactgroup定義

【/usr/local/nagios/etc/hosts.cfg - contact/contactgroup定義】

```
define contact{
        contact_name                    nagios
        alias                           Nagios Admin
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           nagios@localhost
        }

define contactgroup{
        contactgroup_name       admins
        alias                   Nagios Administrators
        members                 nagios
        }
```


Nagiosの設定：host定義

【/usr/local/nagios/etc/hosts.cfg - host templateとmail0.xy.jpのhost定義例】

```
define host{
        name                            generic-host    ; The name of this host template
        notifications_enabled           1   ; Host notifications are enabled
        event_handler_enabled           1   ; Host event handler is enabled
        flap_detection_enabled          1   ; Flap detection is enabled
        failure_prediction_enabled      1   ; Failure prediction is enabled
        process_perf_data               0   ; Process performance data
        retain_status_information       1   ; Retain status information across program restarts
        retain_nonstatus_information    1   ; Retain non-status information across program restarts
        register                        0   ; ITS NOT A REAL HOST, JUST A TEMPLATE!
        notification_interval   120
        notification_period     24x7
        notification_options    d,u,r
        max_check_attempts      10
        }

define host{
        use             generic-host            ; Name of host template to use
        host_name       mail0.xy.jp
        alias           mail server
        parents         fw-dmz
        address         192.168.10.10
        check_command   check-host-alive
        contact_groups  admins
        }
```

Nagiosの設定：hostgroup定義

【/usr/local/nagios/etc/hosts.cfg - hostgroup定義】

```
define hostgroup{
        hostgroup_name  DMZ
        alias           DMZ Segment
        members         fw-dmz, mail0.xy.jp, dns0.xy.jp, dmz-sw0
        }

define hostgroup{
        hostgroup_name  INTERNET
        alias           INTERNET Segment
        members         isp-upstream, fw-upstream, isp-www-if, www.xy.jp
        }

define hostgroup{
        hostgroup_name  INTRA
        alias           INTRA Segment
        members         fw-intra, localhost, hq-sw0, fs0.intra.xy.jp
        }

define hostgroup{
        hostgroup_name  BRANCH
        alias           BRANCH Segment
        members         fw-vpn, b1-rt0-vpn, b1-rt0-intra, b1-sw0
        }
```


Nagiosの設定：service定義1

【/usr/local/nagios/etc/hosts.cfg - service template】

```
define service{
        name                            generic-service ; The 'name' of this service template
        active_checks_enabled           1       ; Active service checks are enabled
        passive_checks_enabled          1       ; Passive service checks are enabled/accepted
        parallelize_check               1       ; Active service checks should be parallelized
        obsess_over_service             1       ; We should obsess over this service (if necessary)
        check_freshness                 0       ; Default is to NOT check service 'freshness'
        notifications_enabled           1       ; Service notifications are enabled
        event_handler_enabled           1       ; Service event handler is enabled
        flap_detection_enabled          1       ; Flap detection is enabled
        failure_prediction_enabled      1       ; Failure prediction is enabled
        process_perf_data               1       ; Process performance data
        retain_status_information       1       ; Retain status information across program restarts
        retain_nonstatus_information    1       ; Retain non-status information across program restarts
        register                        0       ; ITS NOT A REAL SERVICE, JUST A TEMPLATE!
        is_volatile                     0
        check_period                    24x7
        max_check_attempts              4
        normal_check_interval           5
        retry_check_interval            1
        notification_interval           960
        notification_period             24x7
        }
```

Nagiosの設定：service定義2

【/usr/local/nagios/etc/hosts.cfg - service定義】

```
define service{
        use                     generic-service         ; Name of service template to use
        host_name               localhost, fw-dmz,mail0.xy.jp,dns0.xy.jp,dmz-sw0,isp-upstream,fw-upstream,
isp-www-if,www.xy.jp,fw-intra,hq-sw0,fs0.intra.xy.jp,fw-vpn,b1-rt0-vpn,b1-rt0-intra,b1-sw0
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
        contact_groups  admins
        }

define service{
        use                     generic-service         ; Name of service template to use
        host_name               localhost
        service_description     Root Partition
        check_command           check_local_disk!20%!10%!/
        contact_groups  admins
        }

define service{
        use                     generic-service         ; Name of service template to use
        host_name               mail0.xy.jp, dns0.xy.jp
        service_description     DNS
        check_command           check_dns
        contact_groups  admins
        }
define service{
        use                     generic-service         ; Name of service template to use
        host_name               mail0.xy.jp, dns0.xy.jp
        service_description     SMTP
        check_command           check_smtp
        contact_groups  admins
        }
```


Nagiosの設定：service定義2

【/usr/local/nagios/etc/hosts.cfg - service定義】

```
define service{
        use                     generic-service         ; Name of service template to use
        host_name               mail0.xy.jp
        service_description     POP
        check_command           check_pop
        contact_groups  admins
        }

define service{
        use                     generic-service         ; Name of service template to use
        host_name               localhost, www.xy.jp, mail0.xy.jp, dns0.xy.jp, fw-intra
        service_description     SSH
        check_command           check_ssh
        contact_groups  admins
        }

define service{
        use                     generic-service         ; Name of service template to use
        host_name               localhost, www.xy.jp
        service_description     HTTP
        check_command           check_http
        contact_groups  admins
        }
```

Nagiosの設定：hostextinfo定義

【/usr/local/nagios/etc/hosts.cfg - hostextinfo定義抜粋】

```
define hostextinfo{
    host_name       localhost
    icon_image      linux40.jpg
    icon_image_alt  linux40.gif
    vrml_image      linux40.png
    statusmap_image linux40.gd2
    2d_coords       300,50
}

define hostextinfo{
    host_name       mail0.xy.jp
    icon_image      freebsd40.jpg
    icon_image_alt  freebsd40.gif
    vrml_image      freebsd40.png
    statusmap_image freebsd40.gd2
    2d_coords       50,50
}

define hostextinfo{
    host_name       www.xy.jp
    icon_image      sun40.jpg
    icon_image_alt  sun40.gif
    vrml_image      sun40.png
    statusmap_image sun40.gd2
    2d_coords       50,150
}
```

```
define hostextinfo{
    host_name       fs0.intra.xy.jp
    icon_image      win40.jpg
    icon_image_alt  win40.gif
    vrml_image      win40.png
    statusmap_image win40.gd2
    2d_coords       150,450
}

define hostextinfo{
    host_name       isp-upstream
    icon_image      router40.jpg
    icon_image_alt  router40.gif
    vrml_image      router40.png
    statusmap_image router40.gd2
    2d_coords       450,300
}


define hostextinfo{
    host_name       dmz-sw0
    icon_image      switch40.jpg
    icon_image_alt  switch40.gif
    vrml_image      switch40.png
    statusmap_image switch40.gd2
    2d_coords       150,100
}
```

2006/12/7

追加資料２：
Hobbit WML監視画面追加設定

2006/12/7

Hobbit WML監視画面追加設定

- HobbitにWML画面を作成させるためには以下の手順にてWML用BBDISPLAYサーバを追加する
  - ① $HOBBITHOME/etc配下のプロセス起動設定：hobbitlauch.cfgに次ページの設定を追加
  - ② $HOBBITHOME/www配下にwmlディレクトリを作成
  - ③ hobbitを再起動する。
  - ④ apache httpd.confにindex.wmlをデフォルトファイルにする右のリストの設定を追加
  - ⑤ apache再起動
  - ⑥ hobbit-alerts.cfgにWML URL通知メールスクリプトを追加（本文参照）

[httpd.confへの追加設定]

```
<IfModule mod_dir.c>
    <IfModule mod_php3.c>
        <IfModule mod_php4.c>
            DirectoryIndex index.php index.php3 index.html
            index.wml
        </IfModule>
        <IfModule !mod_php4.c>
            DirectoryIndex index.php3 index.html index.wml
        </IfModule>
    </IfModule>
    <IfModule !mod_php3.c>
        <IfModule mod_php4.c>
            DirectoryIndex index.php index.html index.wml
        </IfModule>
        <IfModule !mod_php4.c>
            DirectoryIndex index.html index.wml
        </IfModule>
    </IfModule>
</IfModule>
```


hobbitlaunch.cfgへの追加設定

[hobbitlaunch.cfgへの追加設定]

```
# "bbdisplay" runs the bbgen tool to generate the Hobbit webpages from the status information that
# has been received. Big Brother updated the webpages once every 5 minutes. The default here is to
# run it every minute for faster updates, but you can change it if you have a highly loaded server
# and dont need updates that often.

[bbdisplay]
        ENVFILE /usr/home/hobbit/server/etc/hobbitserver.cfg
        NEEDS hobbitd
        GROUP generators
        CMD bbgen $BBGENOPTS --report
        LOGFILE $BBSERVERLOGS/bb-display.log
        INTERVAL 1m

[bbdisplayWAP]
        ENVFILE /usr/home/hobbit/server/etc/hobbitserver.cfg
        NEEDS hobbitd
        GROUP generators
        CMD bbgen $BBGENOPTS --report --wml
        LOGFILE $BBSERVERLOGS/bb-display.log
        INTERVAL 1m

# bbcombotest is an extension script for the Hobbit display server. It generates
```

この設定を追加する

追加資料３：
Big BrotherからHobbit/bbgenで拡張された監視サービスタグ 一覧

Big BrotherからHobbit/bbgenで拡張された監視サービスタグ－1

機能名	タグ用途	使用方法
外部ファイル挿入	外部設定ファイル挿入	include <filename>
	外部BBDISPLAY設定ファイル挿入	dispinclude <filename>
	外部BBNET設定ファイル挿入	netinclude <filename>
BBDISPLAY bb-hosts拡張	多階層サブページ指定	subparent <parentpage> <newpage> [<Page-title>]
	タイトルテキスト挿入	title <title_text>
	表示ホスト名変更	A.B.C.D HOST # NAME:<hostname>
	BBCLIENTエイリアス名指定	A.B.C.D HOST # CLIENT:<hostname>
	ホストコメント表示	A.B.C.D HOST # COMMENT:<host_comment>
	ホスト説明設定	A.B.C.D HOST # DESCR:<HostType>:<Description>
	bb.html非表示設定	A.B.C.D HOST # nodisp
	障害サマリ画面非表示設定	A.B.C.D HOST # nobb2
	複数ホスト定義時優先指定	A.B.C.D HOST # perfer
	IPレンジでのIP死活監視	dialup <hostname> <startIP> <count>
	LARRDグラフ表示設定	A.B.C.D HOST # LARRD:[*,][![<larrdgraph>….]]
NK機能拡張	NK監視指定	A.B.C.D HOST # NK:testname[,<testname>]
	NK対象時間指定	A.B.C.D HOST # NKTIME=<day>:<starttime>:<endtime>[,<day>:<starttime>:<endtime>]
WAP機能拡張	WMLページ設定	A.B.C.D HOST # WML:[+|-]<testname>[,[+|-]<testname>]
状態広報オプション	状態広報規制設定(RED)	A.B.C.D HOST # NOPROPRED:[+|-]<testname>[,[+|-]<testname>]
	状態広報規制設定(YELLOW)	A.B.C.D HOST # NOPROPYELLOW:[+|-]<testname>[,[+|-]<testname>]
	状態広報規制設定(PURPLE)	A.B.C.D HOST # NOPROPPURPLE:[+|-]<testname>[,[+|-]<testname>]
	状態広報規制設定(ACK)	A.B.C.D HOST # NOPROPACK:[+|-]<testname>[,[+|-]<testname>]
稼働率レポート指定	稼働率レポート対象時間指定	A.B.C.D HOST # REPORTTIME=<day>:<starttime>:<endtime>[,<day>:<starttime>:<endtime>]
	稼働率レポートしきい値指定	A.B.C.D HOST # WARNPCT:<percentage>

Big Brotherから hobbit/bbgenで拡張された監視サービスタグー2

機能名	タグ用途	使用方法
BBTEST-NET bb-hosts拡張サービス監視	監視対象ネットワーク指定	A.B.C.D HOST # NET:<location>
	SSL認証対象外指定	A.B.C.D HOST # nosslcert
	SSL認証有効期限設定	A.B.C.D HOST # ssldays=<WARNDAYS>:<ALARMDAYS>
	非監視時間指定	A.B.C.D HOST # DOWNTIME=<day>:<starttime>:<endtime>[,<day>:<starttime>:<endtime>]
	SLAレポート対象時間指定	A.B.C.D HOST # SLA=<day>:<starttime>:<endtime>[,<day>:<starttime>:<endtime>]
	監視テスト依存設定	A.B.C.D HOST # depends=(<test A>:<host1>/<test1>,<host2>/<test2>),(<test B>:<host3>/<test3>),[...]
	障害判定回数設定	A.B.C.D HOST # badTEST[-<weekdays>-<starttime>-<endtime>]:<x>:<y>:<z>
	dns監視拡張指定1	A.B.C.D HOST # dns[=<hostname>]
	dns監視拡張指定2	A.B.C.D HOST # dns=<TYPE>:<lookup>[,<TYPE>:<lookup>...]
	dns監視拡張指定(dig使用)	A.B.C.D HOST # dig
	ntp監視	A.B.C.D HOST # ntp
	ldap監視指定1	A.B.C.D HOST # ldap
	ldap監視指定2	A.B.C.D HOST # ldap://<hostname>/dn[?<attrs>[?<scope>[?<filter>[?<exts>]]]]
	ldap ssl監視指定1	A.B.C.D HOST # ldaps
	ldap ssl監視指定2	A.B.C.D HOST # ldaps://<hostname>/dn[?<attrs>[?<scope>[?<filter>[?<exts>]]]]
	ldap login ID指定	A.B.C.D HOST # ldaplogin=<username>:<password>
	ldap障害ステータス変更指定	A.B.C.D HOST # ldapyellowfail
	RPCサービス監視指定	A.B.C.D HOST # rpc[=rpcservice1,rpcservice2,...]
BBTEST-NET bb-hosts拡張IP死活監視	IP死活監視指定	A.B.C.D HOST # conn
	IP死活監視拡張指定	A.B.C.D HOST # conn={best,|worst,}IP1[,IP2...]
	IP死活監視非実施指定	A.B.C.D HOST # badconn[-weekdays-starttime-endtime]:x:y:z
	IPルーティング監視指定1	A.B.C.D HOST # route:router1,router2,....
	IPルーティング監視指定2	A.B.C.D HOST # route_LOCATION:router1,router2,...
	traceroute確認指定	A.B.C.D HOST # trace
	traceroute非確認指定	A.B.C.D HOST # notrace

Big Brotherから hobbit/bbgenで拡張された監視サービスタグー3

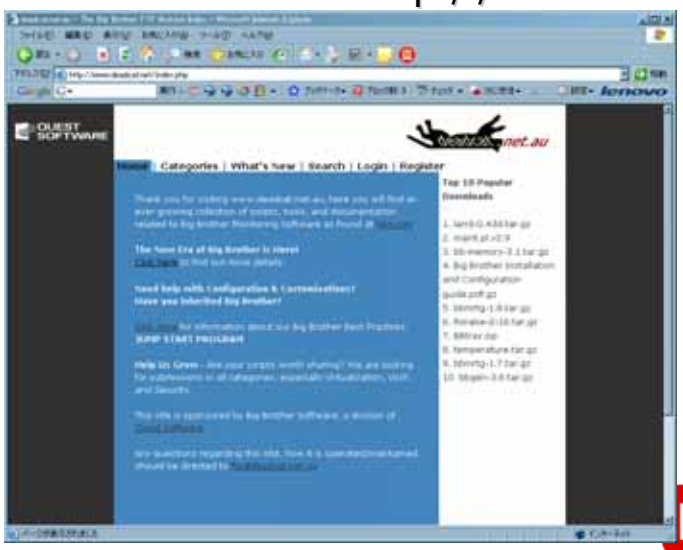
機能名	タグ用途	使用方法
BBTEST-NET HTTP拡張監視	BASIC認証	A.B.C.D HOST # http://USERNAME:PASSWORD@www.sample.com/
	ssl client認証	A.B.C.D HOST # http://CERT:FILENAME@www.sample.com/
	http ver指定	A.B.C.D HOST # http10://www.sample.com/ : use HTTP 1.0
		A.B.C.D HOST # http11://www.sample.com/ : use HTTP 1.1
	ssl ver指定	A.B.C.D HOST # https2://www.sample.com/ : use only SSLv2
		A.B.C.D HOST # https3://www.sample.com/ : use only SSLv3
		A.B.C.D HOST # httpsm://www.sample.com/ : use only 128-bit ciphers
		A.B.C.D HOST # httpsm://www.sample.com/ : use only 128-bit ciphers
	IPアドレス指定	A.B.C.D HOST # http://www.sample.com=1.2.3.4/index.html
		A.B.C.D HOST # http://www.sample.com:3128=1.2.3.4/index.html
	proxy経由	A.B.C.D HOST # http://webproxy.sample.com:3128/http://bb4.com/
		A.B.C.D HOST # http://fred:Wilma1@webproxy.sample.com:3128/http://bb4.com/
BBTEST-NET コンテンツチェック	コンテンツチェック	A.B.C.D HOST # cont[=COLUMN];URL;[expected_data_regexp|A.B.C.D HOST #digesttype:digest]
		A.B.C.D HOST # content=URL
	CGIチェック	A.B.C.D HOST # post[=COLUMN];URL;form-data;[expected_data_regexp|A.B.C.D HOST #digesttype:digest]
	コンテンツ不稼働チェック	A.B.C.D HOST # nocont[=COLUMN];URL;forbidden_data_regexp
	CGI不稼働チェック	A.B.C.D HOST # nopost[=COLUMN];URL;form-data;expected_data_regexp
	content-typeチェック	A.B.C.D HOST # type[=COLUMN];URL;expected_content_type

追加資料4:
Hobbit/Big Brotherによるトラフィック監視
<< MRTGとの連携:bbmrtg.pl >>

Hobbit/Big Brother機能拡張スクリプト

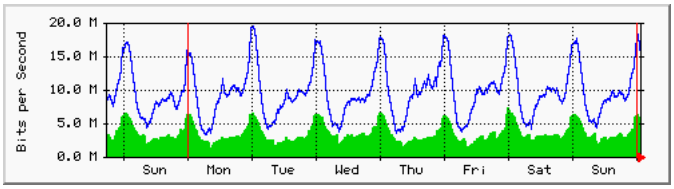
- 拡張インタフェースが公開されており、多彩な拡張監視モジュールが存在する
  - オープンソースの利点を生かし、BB基本ソフトをそのまま置換する機能拡張版ソフトも存在する
    - http://www.deadcat.net/
  - モジュールごと拡張版への置換
  - 外部拡張スクリプトによる機能追加
- 実現されるもの
  - サービス稼働監視拡張:
    - radius, ntp, ldap, smb, mqueue, ・・・
    - RDBS (ORACLE, PostgreSQL, MySQL, ・・・)
    - 他システム監視: RAS, UPS, RAID, Printer, ・・・
  - 他ソフトとの連係: 例えばMRTG、Cacti(RRDtools Frontend)
  - モジュールへの入れ替えによる高速化
  - BBTray : Big Brother監視ツール on Windows

BB - Extension Archive
http://www.deadcat.net



トラフィックリソース監視
MRTG(Multi Router Traffic Grapher)

- MRTG：Multi Router Traffic Grapher
  - http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
- 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
- ネットワーク・トラフィック監視をする上での定番ツール



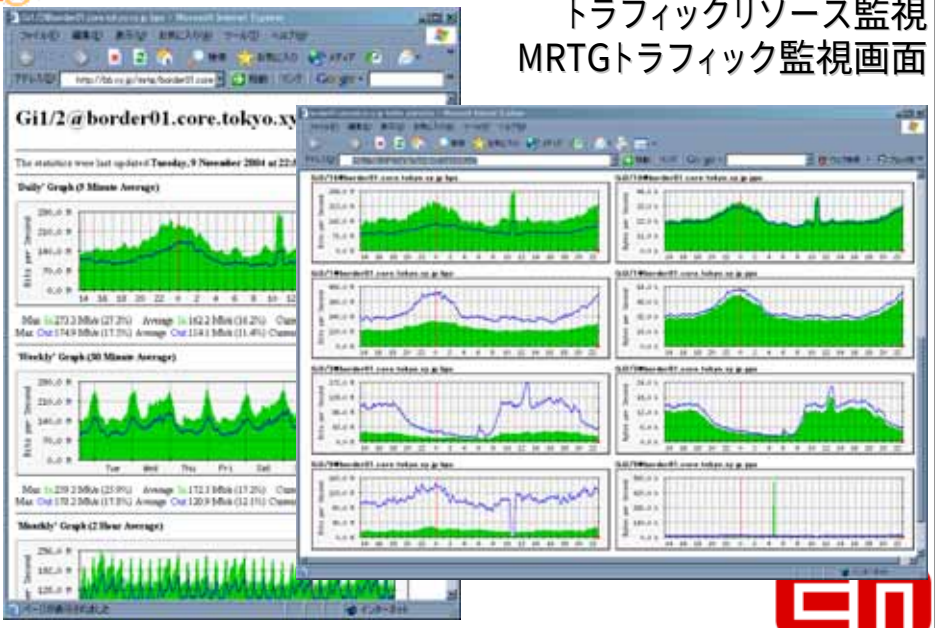
トラフィックリソース監視 MRTGの特徴1

- 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

トラフィックリソース監視 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
- RRDtool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

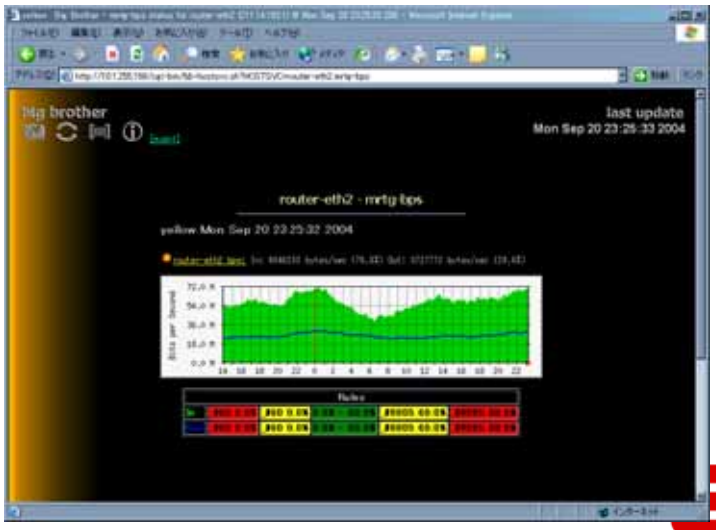
トラフィックリソース監視
MRTGトラフィック監視画面



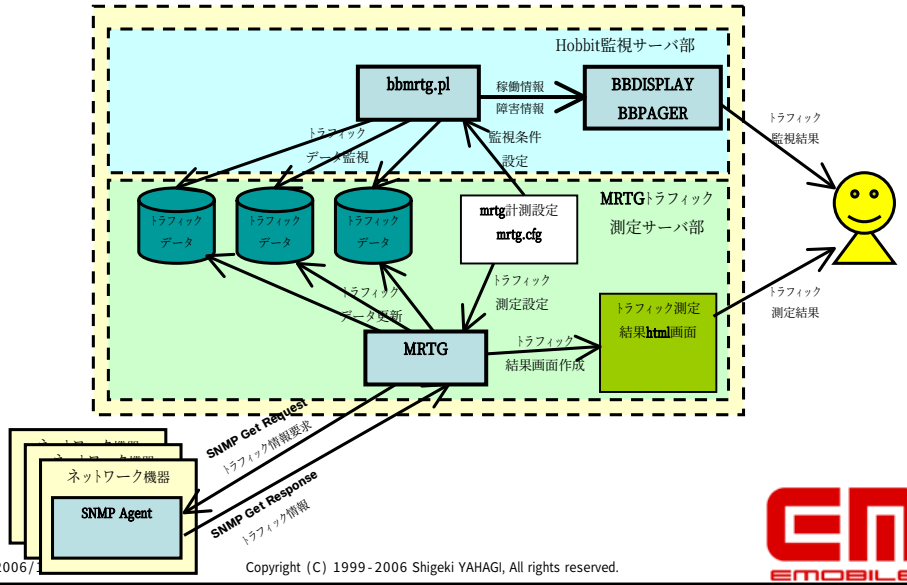
トラフィックリソース監視
BBとMRTGの連携：bbmrtg.pl

- MRTGと連携するHobbit/BigBrother機能拡張
  - bbmrtg.pl ver 1.8：
    http://www.deadcat.net/viewfile.php?fileid=926
- MRTGで計測しているトラフィックデータの監視を行う
- MRTGの計測設定ファイルの各測定項目にしきい値を設定
- 監視はMRTGの計測ファイル単位で行う。MRTG設定ファイルに定義されている計測項目を監視する

トラフィックリソース監視
BBとMRTGの連携：bbmrtg.pl



bbmrtg.plプロセス関係図



トラフィックリソース監視 bbmrtg.pl 指定タグ - 1

① 監視ホスト名称タグ（必須項目）
- 'bb*host[<測定項目>]: <監視ホスト名>‘
- <測定項目>が属するホスト名を<監視ホスト名称>に設定する。<監視ホスト名>はbb-hostsに登録されている名前と一致している必要がある。
- 設定例：bb*host[fw-intra-bps]: fw-intra

② 監視テスト項目タグ（必須項目）
- 'bb*svc[<測定項目>]: <テスト項目名>‘
- <測定項目>で行われているトラフィック監視のテスト名称を設定する。<監視ホスト名>のエントリーの中で<テスト項目名>として表示される。
- 設定例：bb*svc[fw-intra-bps]: mrtg-bps

トラフィックリソース監視 bbmrtg.pl 指定タグ - 2

- 監視しきい値タグ（必須項目）
  - 'bb*yellow[<測定項目>]: <上限閾値>‘
  - 'bb*red[<測定項目>]: <上限閾値>‘
  - <測定項目>のトラフィック監視の注意・警報レベルの閾値を設定する。bb*yellowが注意レベル、bb*redが警報レベルの指定になる。値の設定方法は測定値とパーセント値の二つが設定でき、パーセント指定の場合には<測定項目>で設定されるMaxByte[<測定項目>]:での値を100%とした比率で計算される。
- 監視計測単位タグ（オプション項目）
  - bb*unit[<測定項目>]: <表示単位>
  - <測定項目>で行われているトラフィック監視の測定単位を設定する。
  - 設定例：bb*unit[fw-intra-cpu]: "CPU utilization“
- データ種別設定タグ（オプション項目）
  - bb*in[<測定項目>]: <第1系列データ種別>
  - bb*out[<測定項目>]: <第2系列データ種別>
  - 2種類のデータ系列の種別を設定する。
  - 設定例：bb*in[fw-intra-cpu]: "in 1min“
bb*out[fw-intra-cpu]: "in 5min"

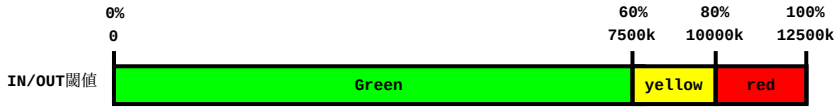
トラフィックリソース監視 bbmrtg.pl 閾値設定 - 1

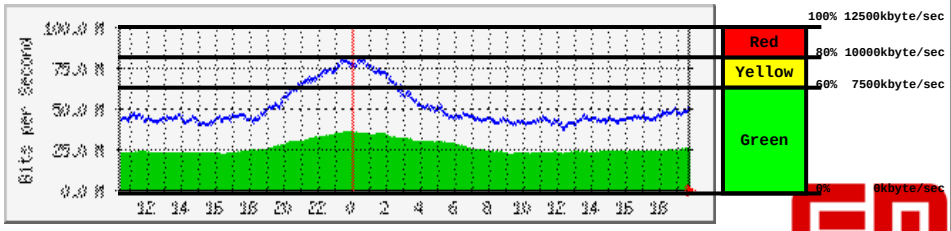
(1) 上限値指定 （測定値指定）

```
MaxByte[fw-intra-bps]: 12500000
bb*yellow[fw-intra-bps]: 7500000
bb*red[fw-intra-bps]: 10000000
```

(1') 上限値指定 （パーセント指定）

```
MaxByte[fw-intra-bps]: 12500000
bb*yellow[fw-intra-bps]: 60%
bb*red[fw-intra-bps]: 80%
```





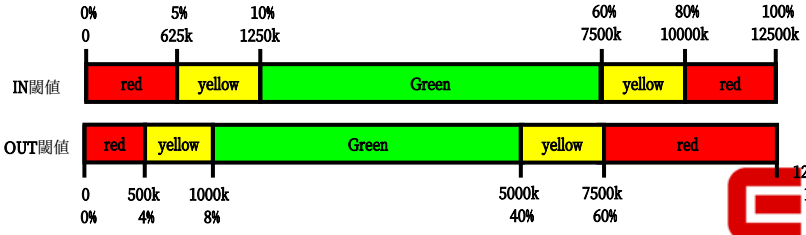
トラフィックリソース監視 bbmrtg.pl 閾値設定 - 2

(2) 上限下限値指定（IN/OUT同値指定、測定値指定）

```
MaxByte[fw-intra-bps]: 12500000
bb*yellow[fw-intra-bps]: 1250000:7500000
bb*red[fw-intra-bps]: 625000:10000000
```

(3) 上限下限値指定（IN/OUT個別指定、測定値指定）

```
MaxByte[fw-intra-bps]: 12500000
bb*yellow[fw-intra-bps]: 1250000:7500000:1000000:5000000
bb*red[fw-intra-bps]: 625000:10000000:500000:7500000
```



fw-intra MRTG設定 bbmrtg.pl版

```
Target[fw-intra-bps]: #Fa2:myReadCom@router
MaxBytes[fw-intra-bps]: 12500000
Title[fw-intra-bps]: fw-intra bps
PageTop[fw-intra-bps]: <H1>fw-intra bps</H1>
Options[fw-intra-bps]: growright, bits
bb*host[fw-intra-bps]: fw-intra
bb*svc[fw-intra-bps]: mrtg-bps
bb*yellow[fw-intra-bps]: 60%
bb*red[fw-intra-bps]: 80%
bb*unit[fw-intra-bps]: bytes/sec

Target[fw-intra-pps]: ifInUcastPkts#Fa2&ifOutUcastPkts#Fa2:myReadCom@router
MaxBytes[fw-intra-pps]: 30000
Title[fw-intra-pps]: fw-intra pps
PageTop[fw-intra-pps]: <H1>fw-intra pps</H1>
YLegend[fw-intra-pps]: packet/sec
ShortLegend[fw-intra-pps]: pps
Options[fw-intra-pps]: growright
bb*host[fw-intra-pps]: fw-intra
bb*svc[fw-intra-pps]: mrtg-pps
bb*yellow[fw-intra-pps]: 10000
bb*red[fw-intra-pps]: 20000

Target[fw-intra-discards]: ifInDiscards#Fa2&ifOutDiscards#Fa2:myReadCom@router
MaxBytes[fw-intra-discards]: 10000
Title[fw-intra-discards]: fw-intra discards
PageTop[fw-intra-discards]: <H1>fw-intra discards</H1>
YLegend[fw-intra-discards]: Discards
ShortLegend[fw-intra-discards]: Discards
Options[fw-intra-discards]: growright
bb*host[fw-intra-discards]: fw-intra
bb*svc[fw-intra-discards]: mrtg-discards
bb*yellow[fw-intra-discards]: 1
bb*red[fw-intra-discards]: 5
```


Hobbit bbmrtg追加設定

- HobbitにWAP画面を作成させるためには以下の手順にてWAP用BBDISPLAYサーバを追加する
  - $HOBBITHOME/etc配下のプロセス起動設定：hobbitlauch.cfgに以下のbbmrtg.plエントリを追加
  - hobbitを再起動する。

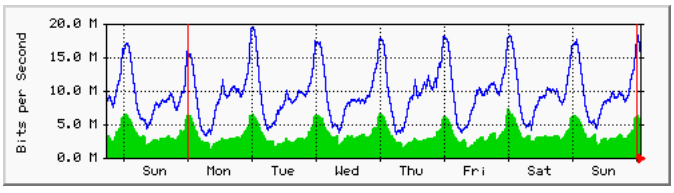
[hobbitlaunch.cfgへの追加設定]

```
[bbmrtg]
        ENVFILE /usr/home/hobbit/server/etc/hobbitserver.cfg
        CMD /usr/home/hobbit/server/ext/bbmrtg.pl
        LOGFILE $BBSERVERLOGS/bbmrtg.log
        INTERVAL 5m
```

追加資料5:
MRTGによるトラフィック計測

MRTGとは

- MRTG : Multi Router Traffic Grapher
- http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html
- 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



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

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
- RRDtool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

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.xy.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

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
```

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

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

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)

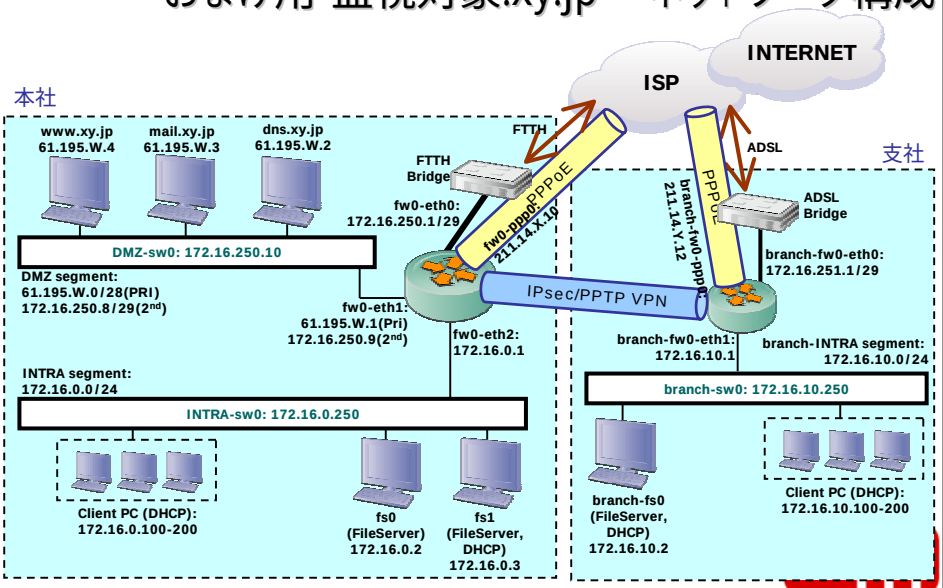
トラフィック監視
使用するSNMP OID/MIB Symbols

- **[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**

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

おまけ用 監視対象:xy.jp – ネットワーク構成



MRTG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

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**

MRTG 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
###
```

MRTG
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
###
```


MRTG
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
###
```

MRTG
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
###
```


MRTGトラフィック計測における 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

crontab – hobbit.xy.jp

```
####
# crontab mrtg@hobbit.xy.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@hobbit.xy.jp end
##
```


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@hobbit.xy.jp (Cron Daemon)
To: alert@xy.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
```

参考資料:文献/URL

参考:文献1

- “Big Brotherによるネットワーク監視”
  - http://www.thinkit.co.jp/tech/#20
  - 矢萩茂樹@イー・モバイル
    - 第1回 ネットワーク監視とは何をするのか
    - 第2回 インストールと基本設定
    - 第3回 監視サーバ側での詳細設定
    - 第4回 BBサーバのメンテナンス
    - 第5回 クライアントプローブのインストール
    - 第6回 機能拡張 - BBtrayとmaint.pl
    - 第7回 機能拡張 - LARRD
    - 第8回 トラフィック監視(前編)
    - 第9回 トラフィック監視(後編)
    - 第10回 究極のBB機能拡張bbgen
    - 第11回 bbgenの拡張機能
    - 第12回 bbgenの機能拡張2 ～複数監視画面定義

参考：文献2

- ■特集 Hobbit／Nagios／Hinemosで実現 サーバ＆ネットワークモニタリング大全
  - 2章：サーバ／ネットワーク監視の決定版Hobbit登場
  - 矢萩茂樹／越川康則
  - Software Design 2006/12月号
- ■特集 事前対策はこれで完璧！ 安心ネットワークのススメ
  - 3章：ネットワーク監視システムHobbit入門
  - 矢萩茂樹／越川康則
  - Software Design 2006/5月号
- ”丸ごとわかる「ネットワーク監視」の秘訣”
  - 矢萩茂樹@イーアクセス / 河本卓司
  - NetworkWorld 2004/10月号

参考：文献3

- “Big Brotherで快適ネットワークシステム管理”
  - Software Design 2003/9-2004/11
  - 矢萩茂樹@イーアクセス / 越川康則
    - 2003/10月号: 第 1回 Big Brother概説
    - 2003/11月号: 第 2回 Big Brotherを設定する
    - 2003/12月号: 第 3回 Big Brotherサーバの詳しい設定
    - 2004/01月号: 第 4回 bbclientと拡張スクリプト
    - 2004/02月号: 第 5回 BBの外部拡張 ①
    - 2004/03月号: 第 6回 BBでできるセキュリティ監視（BBの外部拡張 ②）
    - 2004/04月号: 第 7回 最新バージョン1.9eの導入/バージョンアップ
    - 2004/05月号: 第 8回 独自エクステンションを導入する
    - 2004/06月号: 第 9回 BBのメンテナンス
    - 2004/07月号: 第10回 BBシステムのチューニング①
    - 2004/08月号: 第11回 複数BBサーバによる分散処理（BBシステムのチューニング②）
    - 2004/09月号: 第12回 BBによるDBの監視
    - 2004/10月号: 第13回 BBによるSNMP監視①
    - 2004/11月号: 第14回 BBによるSNMP監視②
    - 2004/12月号: 第15回 BBによるSNMP監視③ そして 次のBB：bbgen
  - Software Design 2005/12 おまけCDROMに全記事PDF収録

参考：文献4

- ネットワーク現場の教科書
  - Network World特別編集号
  - IDGムックシリーズ

ツールURL集1

- **AWARE**
  - **http://www.elegant-software.com/software/aware/**
- **Big Brother**
  - **http://bb4.org/**
  - **Extensions Archive: http://www.deadcat.net/**
- **Big Sister**
  - **http://bigsister.sourceforge.net/**
- **Demarc PureSecure**
  - **http://demarc.com/**
- **Expect**
  - **http://expect.nist.gov/**
- **fping**
  - **http://www.fping.com/**
- **Ganglia**
  - **http://ganglia.info/?page_id=47**
- **Graphviz**
  - **http://www.graphviz.org/**
- **hping**
  - **http://www.hping.org/**

ツールURL集2

- Hobbit Network Monitor
  - http://hobbitmon.sourceforge.net/
- IPTraf
  - http://cebu.mozcom.com/riker/iptraf/index.html
- Lire
  - http://www.logreport.org/
- LogSentry (Senty Tools)
  - http://sourceforge.net/projects/sentrytools
- MRTG
  - http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/
- mon
  - http://www.kernel.org/software/mon
- monit
  - http://www.tildeslash.com/monit/
- moodss
  - http://moodss.sourceforge.net/
- Nagios (NetSaint)
  - http://www.nagios.org/
- Nagios Plugins and Add Ons Exchange
  - http://www.nagiosexchange.org/
- Nagios日本語サポートページ
  - http://nagios.x-trans.jp/naija/

ツールURL集3

- NeTraMet
  - http://www2.auckland.ac.nz/net/NeTraMet/
- 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
  - 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

- OS-SIM
  - http://www.ossim.net/
- RRDtool
  - http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/
  - Frontend - Cacti
    - http://www.cacti.net/
  - 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/
- sing
  - http://sourceforge.net/projects/sing/
- snort
  - http://www.snort.org/
- ssh
  - http://www.ssh.com/
- SWATCH
  - http://swatch.sourceforge.net/

ツールURL集6

- syslog-ng
  - http://www.balabit.com/products/syslog_ng/
- php-syslog-ng
  - http://www.vermeer.org/projects/php-syslog-ng
- SysOrb
  - http://www.evalesco.com
- visualroute
  - http://www.visualroute.com
- Zabbix
  - http://zabbix.sourceforge.net/