

第2部 端末の接続管理と上位サービス

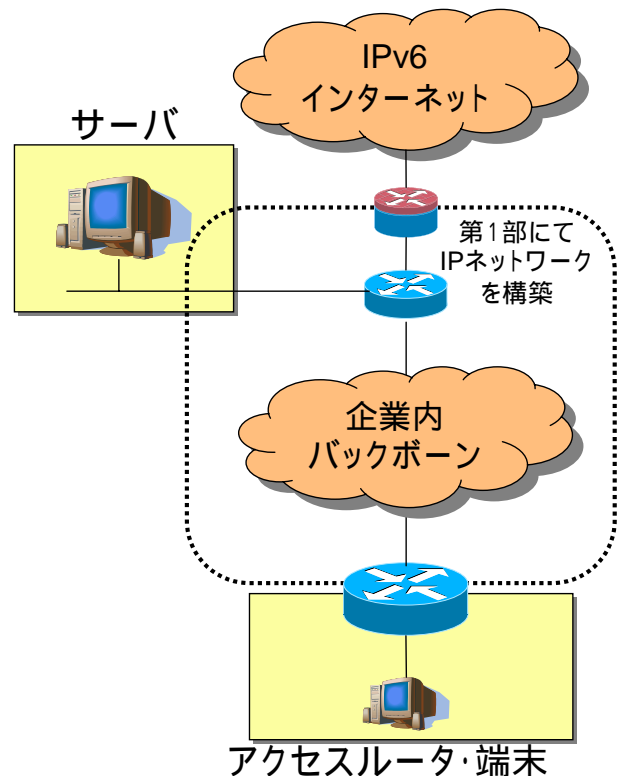
担当

加藤淳也 / NTT情報流通プラットフォーム研究所

1

第2部のアウトライン

1. DHCPv6による端末の着脱管理
2. 名前解決インフラ構築
3. Webインフラ構築
4. ファイルシェアリングサービス構築
5. 監視システム構築



2

1 . DHCPv6による端末の着脱管理

3

企業網でのDHCPv6の有用性

DHCPv6は管理者による端末状態管理に有用

- IPv6のアドレス自動設定方式
 - stateless (RA): アドレス付与状態を把握不可
「いつ」「誰が」「どのアドレスを」使ったか不明
 - statefull (DHCPv6): アドレス付与状態を把握可
端末の接続状況を管理者が把握可能

4

DHCPv4とDHCPv6の違い

DHCPv4	DHCPv6
- IPv4アドレス	- IPv6アドレス
- サブネットマスク	- サブネットマスク なし!
- デフォルトゲートウェイ	- デフォルトゲートウェイ なし!
- DNS情報	- DNS情報
- その他付加的情報 (NTP, SIP, WINSなど)	- その他付加的情報 (NTP, SIP, WINSなど)
- 端末識別はMACアドレス	- 端末識別はDUID

**DHCPv6はデフォルトゲートウェイ付与不可
Router Advertisement (RA)の併用が必要**

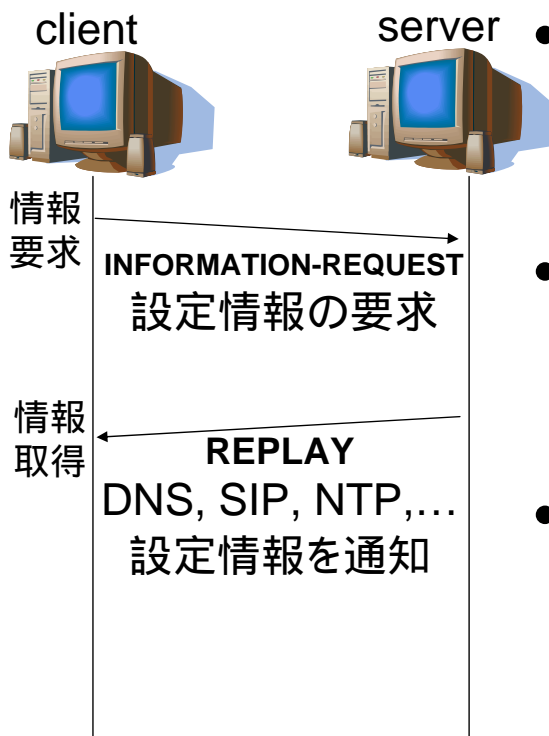
DHCPv6とRAの連携によるアドレス付与

- Router Advertisement (RA)
 - 本来の役目は「ルータの存在」を「広告」するもの
端末はRAの送信元をデフォルトゲートウェイに設定
 - アドレス情報(prefix information option)は**オプション**
アドレス情報なしのRAもありえる
- RAがもつ2つのフラグ : m/o flags

M anagedフラグ	O therconfigフラグ
• RAによりデフォルトゲートウェイのみ設定 • IPv6アドレスと付加的情報(DNS)はDHCPv6で取得	• RAでIPv6アドレスとデフォルトゲートウェイを構成 • 付加的情報(DNSなど)のみDHCPv6で取得する

実習の対象

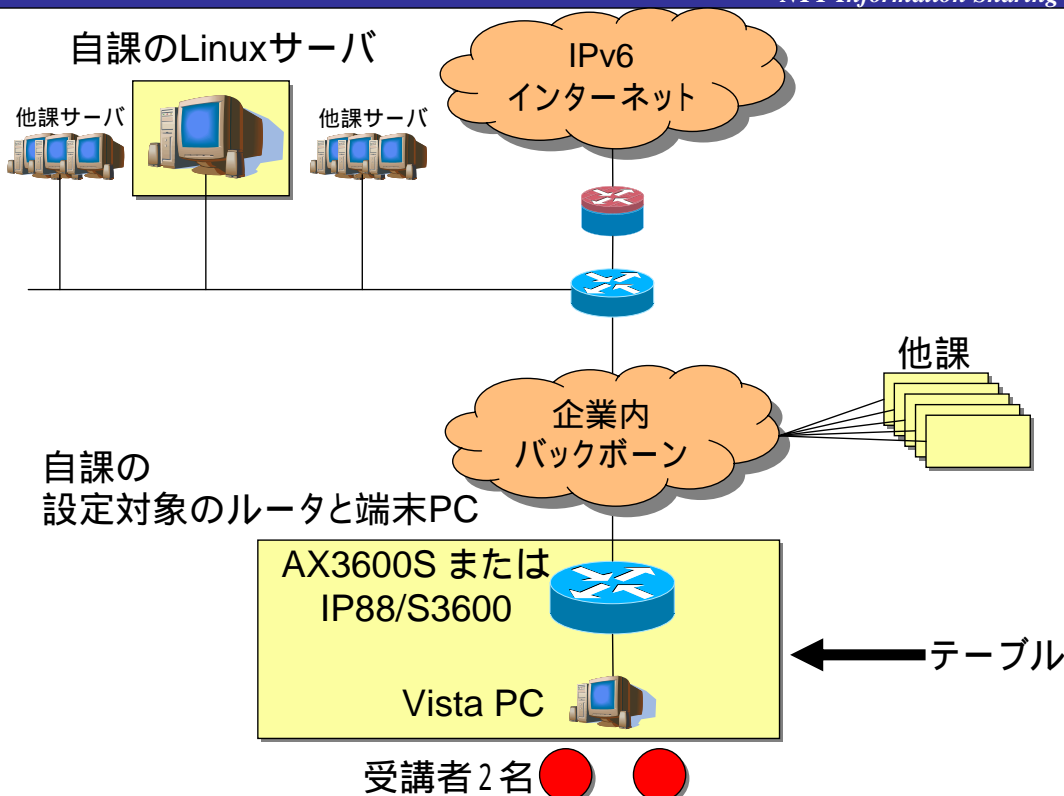
stateless-DHCPv6 (RFC3736)



- サーバがクライアントの状態を管理しない
- 端末の設定情報 (DNS, SIP, NTP) のみを渡す
- 1往復 (2メッセージ) だけで情報を取得

7

実習 DHCPv6サーバ&クライアント構築



8

(実習) stateless-DHCPv6サーバの構成

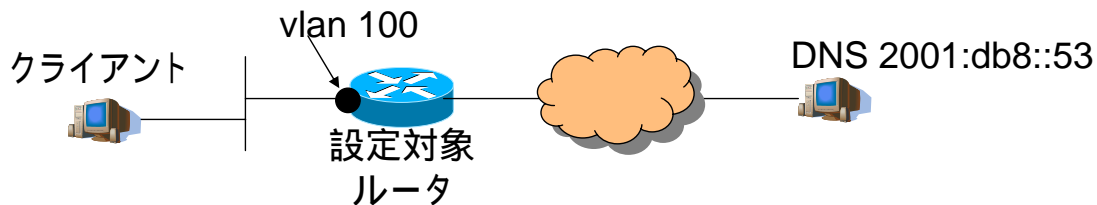
• AX3600S, IP88/S3600シリーズでの構成

アドレス配布プールにDNSサーバのIPv6アドレスのみを設定

```
(config)# ipv6 dhcp pool Group1
(config-dhcp)# dns-server 2001:db8::53
(config-dhcp)# domain-name mycompany.com
(config-dhcp)# exit
```

端末が接続するインターフェイスにプールを関連づけ

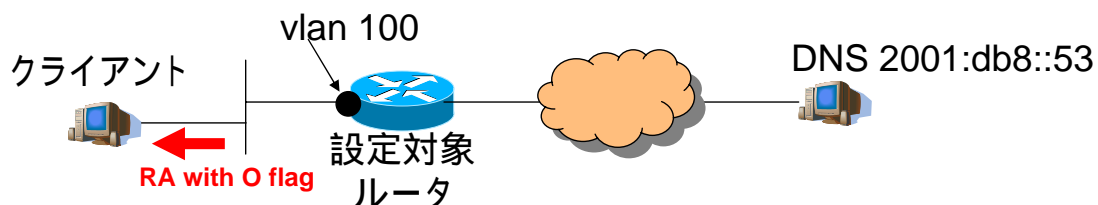
```
(config)# interface vlan 100
(config-if)# ipv6 dhcp server Group1
(config-if)# exit
```



(実習) ルータへのRAフラグ設定

• RAへOtherconfigフラグを立てる設定

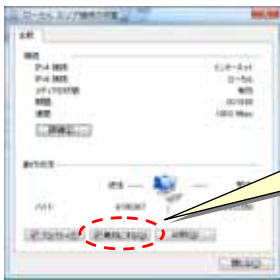
```
(config)# interface vlan 100
(config-if)# ipv6 nd other-config-flag
```



• (参考) RAへManagedフラグを立てる設定

```
(config)# interface vlan 100
(config-if)# ipv6 nd managed-config-flag
```

(実習) Windows Vista 端末を使った動作確認



アダプタの「無効化」「有効化」を行い、アドレス取得をリスタート

- コマンドプロンプト上で取得値を確認

```
C:¥>ipconfig /all | more
イーサネット アダプタ ローカル エリア接続:

接続固有の DNS サフィックス . . . . : mycompany.com
IPv6 アドレス . . . . . : 2001:db8::adca:a5b:eb0b:4ac6(優先)
一時 IPv6 アドレス. . . . . : 2001:db8::e59b:d5eb:9caa:e8b(優先)
リンクローカル IPv6 アドレス. . . . . : fe80::adca:a5b:eb0b:4ac6%9(優先)
IPv4 アドレス . . . . . : 192.168.0.2(優先)
サブネット マスク . . . . . : 255.255.255.0
デフォルト ゲートウェイ . . . . . : fe80::220:e0ff:fe60:4144%9
                                      192.168.0.1
DHCP サーバー . . . . . : 10.0.24.1
DNS サーバー. . . . . : 192.168.0.53
                                      2001:db8::53
```

(実習) ipconfig.exe コマンドのIPv6の扱い

- IPv6アドレス・ホストコンフィグ情報の開放

```
C:¥>ipconfig /release6
```

- IPv6アドレス・ホストコンフィグ情報の再取得

```
C:¥>ipconfig /renew6
```

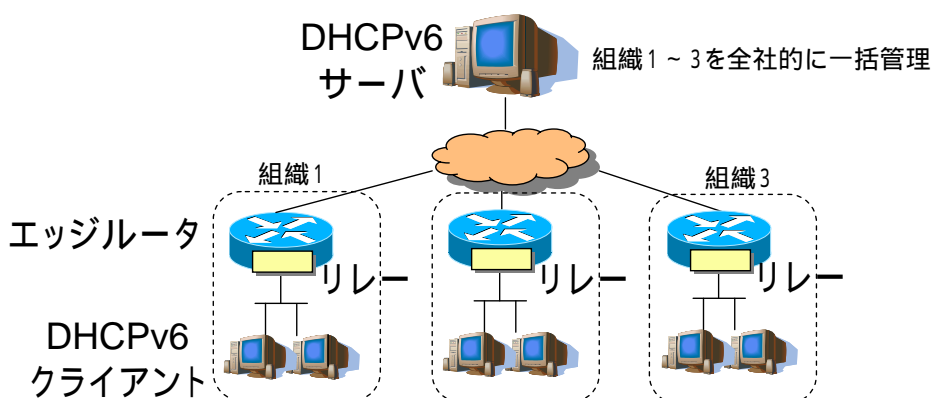
ルータ要請 (Router Solicitation) が再送信される
応答RAのフラグの状態でDHCPv6クライアントも再起動される



Windows Vistaでは特権が必要
コマンドプロンプトを管理者モードで実行

(デモ) DHCPv6リレーによるサーバの集約化

- 端末状態管理の集中化による管理稼働削減
 - サーバを一元的に配備し管理対象を絞り込む
 - DHCPv6リレーエージェントを使ってメッセージを集約する。リレーは状態を持たない(管理不要)



13

(デモ) DHCPv6サーバの構成法

- /etc/sysconfig/dhcp6s - DHCPサーバ設定ファイル

```
interface eth0 {
    option dns_servers 2001:db8::53 mydomain.com;

    link AAA {
        range 2001:db8::1000 to 2001:db8::2000/64;
    };
};
```

割り当てアドレスの範囲を定義

デフォルトゲートウェイの設定は存在しない

- /etc/sysconfig/dhcp6s - スタートアップ設定

```
# specify the interface for dhcp6s
DHCP6SIF=eth0
# Command line options here
DHCP6SARGS=
```

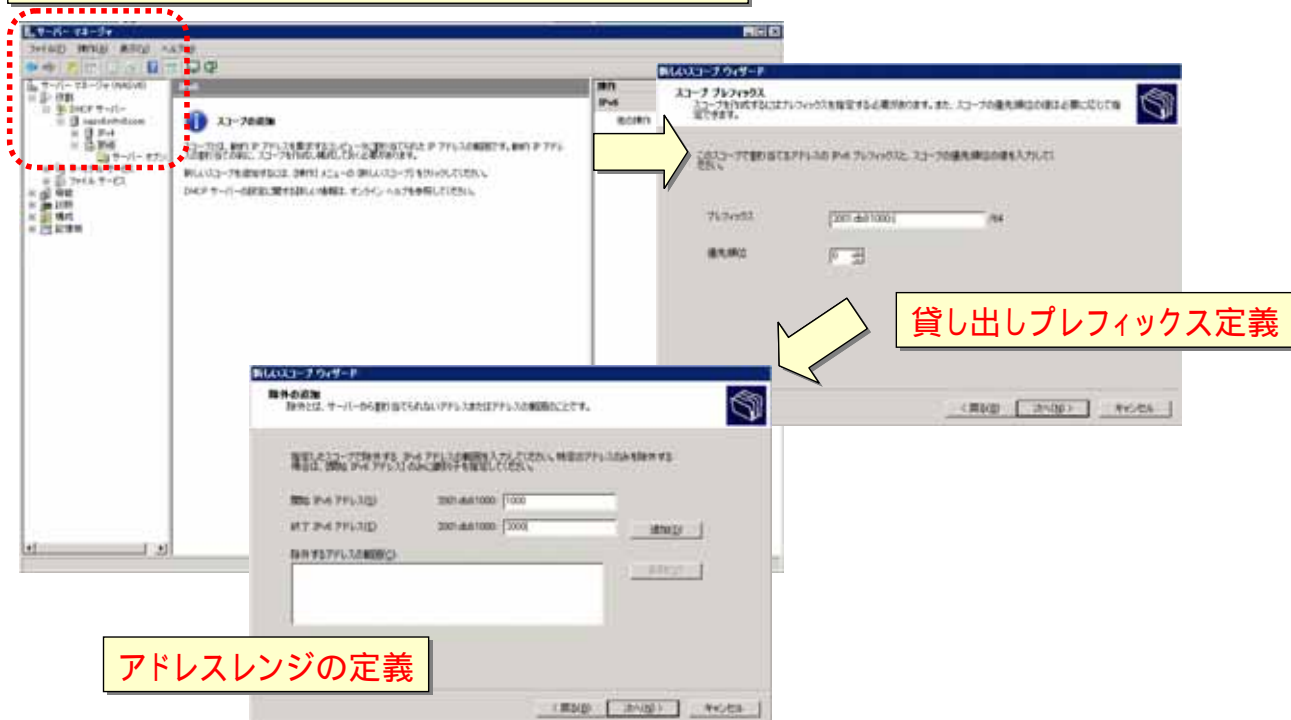
- netstat -an で待ち受けを確認できるアドレス

ff02::1:2	全DHCPリレーとサーバ	クライアントが割当て要求に使用するアドレス
ff05::1:3	全DHCPサーバ	リレーがサーバを探すアドレス

14

(参考) Windows Server 2008のDHCPv6サーバ設定

「サーバー マネージャ」を起動して設定を開始



(デモ) DHCPv6リレーエージェントの構築法

Ciscoルータでのリレーエージェントの設定

```
interface GigabitEthernet0/0.100
  ipv6 enable
  ipv6 address 2001:db8:1000::1/64
  ipv6 dhcp relay destination 2001:db8::547
```

CiscoルータでのRAのManagedフラグ有効化設定

```
interface GigabitEthernet0/0.100
  ipv6 nd managed-config-flag
```

CiscoルータでのRAのOtherConfigフラグ有効化設定

```
interface GigabitEthernet0/0.100
  ipv6 nd other-config-flag
```


(デモ) DHCPv6サーバのログ確認

- 「いつ」「誰に(DUID)」「どのアドレス」を貸し出したか確認する

```
Nov/06/2008 12:18:12
    get DHCP option option request, len 8
    requested option: domain search list
    requested option: DNS_SERVERS
Nov/06/2008 12:18:12
    client ID 00:01:00:01:0d:bb:81:ae:00:16:41:ac:d3:36
Nov/06/2008 12:18:12 request address is 2001:db8:1::1000/0
Nov/06/2008 12:18:12 lease address is 2001:db8:1::1000/0
Nov/06/2008 12:18:12 status code: success
```

DUID : 端末を一意に識別するID

端末へ貸し出したIPv6アドレス

17

DUID (DHCP Unique Identifier)

- DHCPv6でホストを一意に識別する値
 - non-Ethernet(MACが可変 または ない)リンクを想定
- DUIDのタイプと読み方
 - **Type 1: タイムスタンプ付きリンク層アドレス**
00:01:00:01:tt:tt:tt:tt:mm:mm:mm:mm:mm
タイプ メディア 時刻 MACアドレス
 1=Ethernet
 - Type 2: ベンダによる割当 ID
00:02:EE:EE:NN:NN:NN:NN:NN: (可変長)
 ベンダID ベンダが規定するID
 - Type 3: リンク層アドレス
00:03:00:01:mm:mm:mm:mm:mm (可変長)
 メディア MACアドレス
 1=Ethernet

18

(参考) m/o フラグのIETFでの標準化状況

- m/o フラグの詳細な使用方法は標準化中
 - 基本的な動作は決定している(実習にて紹介)
 - サーバ障害やメッセージ欠落などの異常系の動作を整理中
- 実装の状況
 - Windows Vista は最も実用的
 - Windows XPでは未実装
 - Linux/FreeBSDはユーザ依存(スクリプト起動まで)

2. 名前解決インフラ構築

名前解決インフラの構築

IPv6導入により名前解決機構が重要化

- **アドレスを覚えられない(管理者&利用者 負担増)**
- リテラル表記をサポートしていないアプリが存在
例: `http://[2001:db8:1234:::1]/`
 - 主要ブラウザ (google chrome除く)
 - エクスプローラー ×
 - Microsoft Office ファミリ ×
 - リポートで変化するリンクローカルアドレスのスコープ値
- 参考
 - Microsoft Office 製品 の IPv6 への対応状況および利用時の注意点について
<http://www.microsoft.com/japan/office/2007/compatible/IPv6/default.msp>

21

名前解決機構のIPv6対応状況

- 実装状況
 - Linux/FreeBSD系, Windows Server いずれも対応済み
 - 実習ではBIND 9.5.0 の設定を扱う
- ネームサーバのIPv6対応とは
 - (データ) 問い合わせされたホストのIPv6アドレスを返せる
 - (トランスポート) IPv6でDNS問い合わせが受信できる

既存のDNSがAAAAに対応済
データを追加するだけで対応可

↓ 段階的移行が可能

既存のDNSがIPv6トランスポート未対応なら要サーバ更改

dns=1.2.3.53 (IPv4)	kameのIPv6アドレスは?
IPv4トランスポートでIPv6アドレスを求める	
dns=2001:db8:::53 (IPv6)	kameのIPv6アドレスは?
IPv6トランスポートでIPv6アドレスを求める	

IPv4/IPv6トランスポートどちらでも同一内容を返答すること

22

(実習) BIND 9.5.0 の設定

設定項目

1. データベースにIPv6アドレス **データ** (AAAA) を登録する
 2. **トランスポート** がIPv6の問い合わせを受信可とする
1. BINDのデータベース記述ファイル

/var/named/data/mycompany.com.zones - ゾーン(ドメイン)ごとに記述するDB

www	IN	AAAA	2001:db8::80
cifs	IN	AAAA	2001:db8::80
nagios	IN	AAAA	2001:db8::80

2. トランスポートの記述

/etc/named.conf - ネームサーバ全体の動作を定義する設定ファイル

```
options {
    listen-on port 53 { 0.0.0.0; };
    listen-on-v6 port 53 { :: };
    allow-query {
        192.168.0.0/24;           # 自社内からのみ
        2001:db8::/32;          # 問い合わせを受信する
    };
};
```

(実習) ネームサーバのリスタートと動作確認

• サーバ(Linux)のリスタート

```
# /etc/initd/named restart
named を停止中:      [ OK ]
named を起動中:      [ OK ]
```

• 起動ログもあわせて確認する - /var/log/messages

```
Nov  4 16:58:05 localhost named[3462]: exiting
Nov  4 16:58:05 localhost named:
      /etc/named.conf:18: missing ';' before '}'
```

シンタックスも要チェック

• サーバ(Linux)上での動作確認

```
# netstat -an | grep 53
tcp 0 0 192.168.0.80:53 0.0.0.0:* LISTEN
tcp 0 0 2001:db8::80:53 :::* LISTEN
udp 0 0 192.168.0.80:53 0.0.0.0:*
udp 0 0 2001:db8::80:53 :::*
```

(実習) クライアントからの動作確認

- クライアント (Vista) 上での動作確認

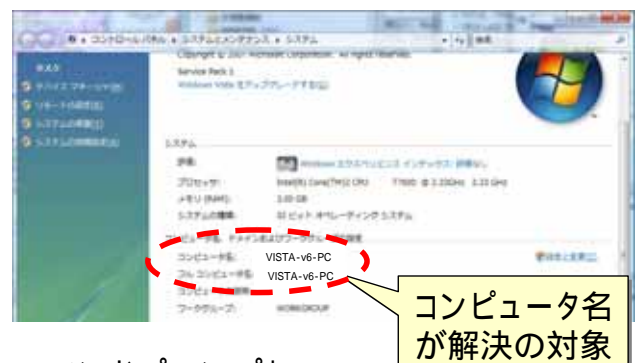
```
C:¥>nslookup www.mycompany.com. 2001:db8::80
                解決したいホスト名      サーバのアドレス
サーバー:      UnKnown
Address:       2001:db8::80
名前:         www.mycompany.com
Addresses:     2001:db8::80
```

- クライアント (Linux) 上での動作確認

```
# dig @::1 www.mycompay.com.
                サーバのアドレス      解決したいホスト名
; ANSWER SECTION:
www.mycompany.com. 86400 IN AAAA 2001:db8::80
```

(参考) Link Local Multicast Name Resolution (LLMNR)

- Windows Vistaからの名前解決機構
- 同一リンク内のホスト名を自動的に解決
 - NetBIOS over TCP/IPv4 に相当
- 設定不要のPnP名前解決機構



コンピュータ名が解決の対象

コマンドプロンプト

```
C:¥>ping VISTA-v6-PC

VISTA-v6-PC [fe80::abcd%9]に ping を送信していません fe80::abcd%9 から 32 バイトのデータ:

fe80::abcd%9 からの応答: 時間 <1ms
fe80::abcd%9 からの応答: 時間 <1ms
```

DNSとは独立!

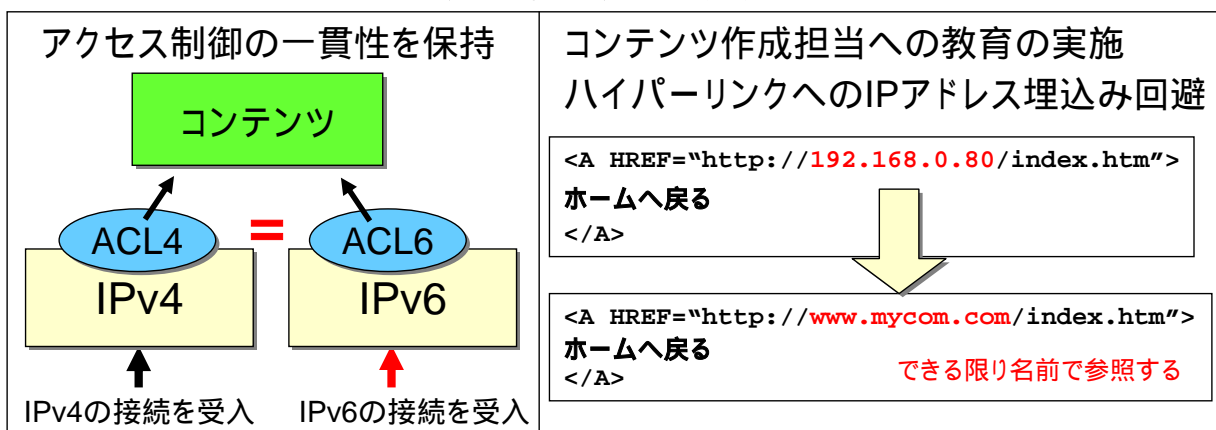
解決値はリンクローカルアドレスなので注意

3 . Webサービスインフラ構築

27

Webインフラストラクチャの構築

- Webサービスは企業活動の中核インフラ
- 実装のIPv6対応は良好
 - HTTPはIPv6に依存しない
 - Windows IIS, Apache とともに商用品質で稼動
- デュアルスタック運用時の注意点



28

(実習) Apache 2.2.x の設定

- 設定ファイルのポイント - /etc/httpd/conf/httpd.conf

- Listenディレクティブ

```
Listen 80 # ポート80番で待ち受け
# 下記の2つの設定と等価
# Listen [::]:80 # ポート番号と区別するため
# IPv6アドレスは[]でくる
# Listen 0.0.0.0:80
```

- Allow, Deny ディレクティブ

```
<Directory /var/www/html>
  Order Deny,Allow
  Deny from all
  Allow from 192.168.0.0/255.255.255.0
  Allow from 2001:db8::/32
</Directory>
```

29

(実習) Apache 2.2.x の動作確認

- サーバのリスタート

```
/etc/init.d/httpd restart
httpd を停止中: [ OK ]
httpd を起動中: [ OK ]
```

- サーバ(Linux)上での動作確認

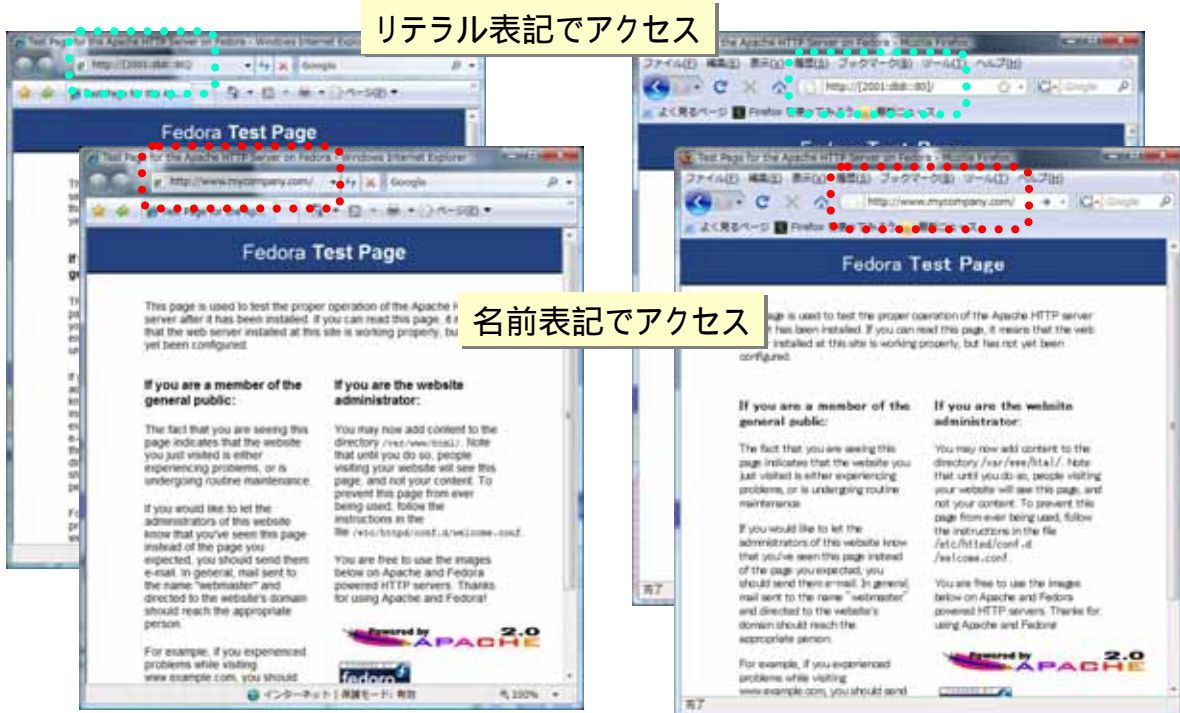
```
netstat -an | grep 80
tcp 0 0 192.168.0.80:80 0.0.0.0:* LISTEN
tcp 0 0 2001:db8::80:80 :::* LISTEN
```

同一のコンテンツに行き着く、2つのルート(IPv4, IPv6)に注意

30

(実習) Apache 2.2.x の動作確認

- クライアント (Vista) 上での動作確認



31

4. ファイルシェアリングサービス構築

ファイルシェアリングサービス

Common Internet File System

NTT Information Sharing Platform Laboratories

- CIFSはWebと並び企業活動の中心サービス
- サーバ実装の対応状況
 - Windows Server 2008 (公式サポート)
 - Samba 3.2以降でIPv6に対応 (2008年7月)
- Windows, Sambaともに安定
 - CIFSはIPv6/IPv4プロトコルに依存しない
- デュアルスタックの運用時の注意点
 - IPv6/IPv4のアクセス制限の一貫性を保持
 - 使用ポートの違いを意識

33

(実習) Samba 3.2.x の設定

NTT Information Sharing Platform Laboratories

設定項目

- 共有ファイルフォルダの新規作成
- アクセス制限の実施
 - 実習では認証サーバとしての機能は割愛
- 設定ファイルへの記述 - /etc/samba/smb.conf

```
[share]
comment = shared folder
browseable = yes
writable = yes
path = /data/shared
hosts deny = ALL
hosts allow = 192.168.0.0/24 2001:db8::/32
```

IPv6アドレスが使用可

34

(実習) Sambaのリスタートと使用ポートの確認

• サーバのリスタート

```
# /etc/init.d/smb restart
# /etc/init.d/nmb restart
SMB/NMB サービスを停止中:      [ OK ]
SMB/NMB サービスを起動中:      [ OK ]
```

• ファイル共有サービスのIPv6/IPv4でのポート番号違い

番号	サービス名	IPv4	IPv6
137	ネームサービス	使用	未使用
138	ブラウジング	使用	未使用
139	ファイル・プリンタ共有	使用	使用
445	ファイル・プリンタ共有	使用	使用

netstat コマンドで使用
ポートとプロトコルを確認

リンクローカル
マルチキャスト
(LLMNR)
ポート5355を使用

ルータでのフィルタリングの際はポート番号に注意

(実習) クライアント(Vista)からのアクセス

¥¥cifs¥share - エクスプローラにUNCパスを記述

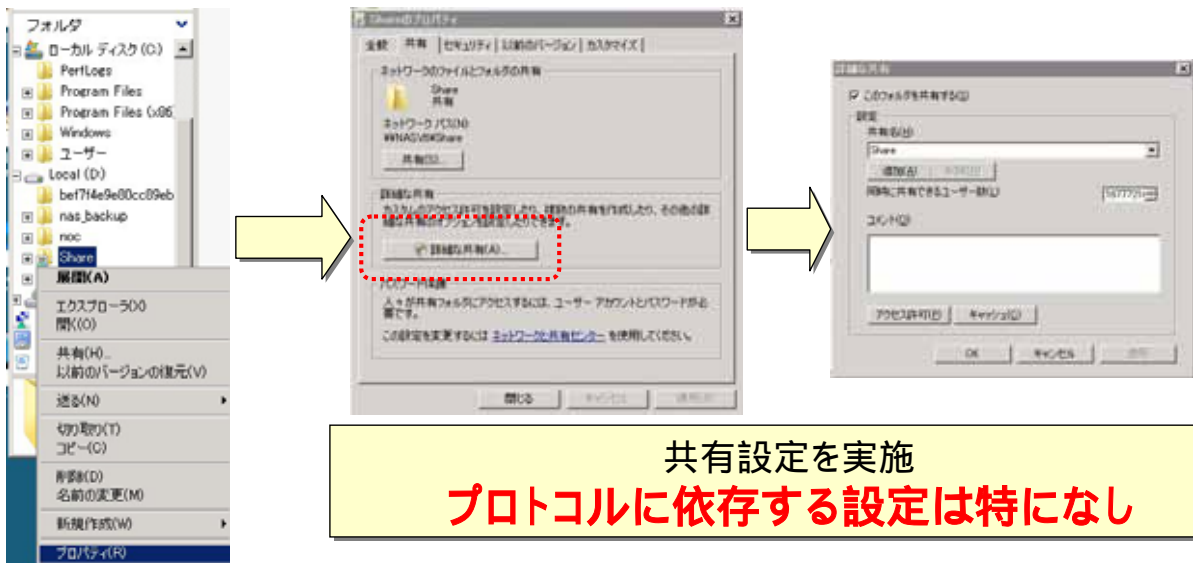


名前ベースアクセスを推奨
アドレス直書は廃止へ

UNC名の記述による使用可否

	¥¥cifs.mycompany.com¥share	名前ベース
	¥¥192.168.0.135¥share	IPv4直書
×	¥¥2001:db8::135¥share	IPv6直書
×	¥¥[2001:db8::135]¥share	IPv6直書

(参考) Windows Server 2008の共有フォルダ公開設定



公開対象のフォルダを右クリックしプロパティを開く

5. 監視システム構築

- Webベースの監視ツール
 - <http://www.nagios.org/>
 - 汎用のブラウザで監視対象を総覧できる
- L3的生死とL4以上のサービス監視が可能
 - 標準で監視可能なプロトコル・サービス
 - ping, smtp, telnet, ssh, ftp, http, pop, imap, dns, ...
 - マシンローカルな資源: CPU, メモリ, ディスク, ...
 - 独自サービスの監視へ拡張可能なプラグイン
 - Windows端末, Mac端末, プリンタ, UPS, RDBMS

拡張性が高く多様な企業環境に適用

39

nagios の監視機能

- 障害の発生・復旧の通知機能
 - 複数の通知先: E-MAIL, ページャ(携帯メール)
 - 監視対象ごと(担当者・監視部門)に設定可
- 監視対象のグループ化機能
 - 例: サーバ郡, ルータ郡をひとまとめに管理
- 監視対象に親子関係を定義可能
 - サーバ単体のトラブル or 上流ルータの障害か切り分けの手助け

40

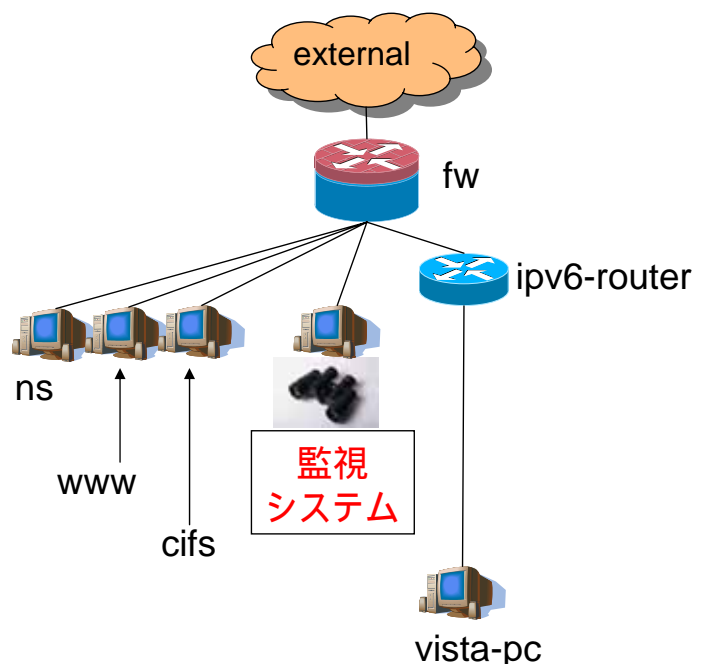
nagios の監視機能 (高度な機能)

- 他のツールとの連動による高い拡張性
 - RRDtools, MRTGと連動した流量監視
 - SNMPエージェント連動したトラップ検出
- リモートホスト上でのプラグイン実行機能
 - 監視対象のホストでプラグインを実行させ結果を収集する機能 (nrpe: Nagios Remote Plugin Executor)
 - システム情報の直接取得が可能
- push型のイベントトラップの処理機能
 - 監視対象の状態変化が発生した場合、nagiosへpush通知する機能 (NSCA: NetSaint Service Check Acceptor)
- 分散監視機能
 - 管理ドメインが分散している環境で、それぞれの内部にnagiosを分散配置。結果をNSCAにより親となるnagiosに集中させて監視する

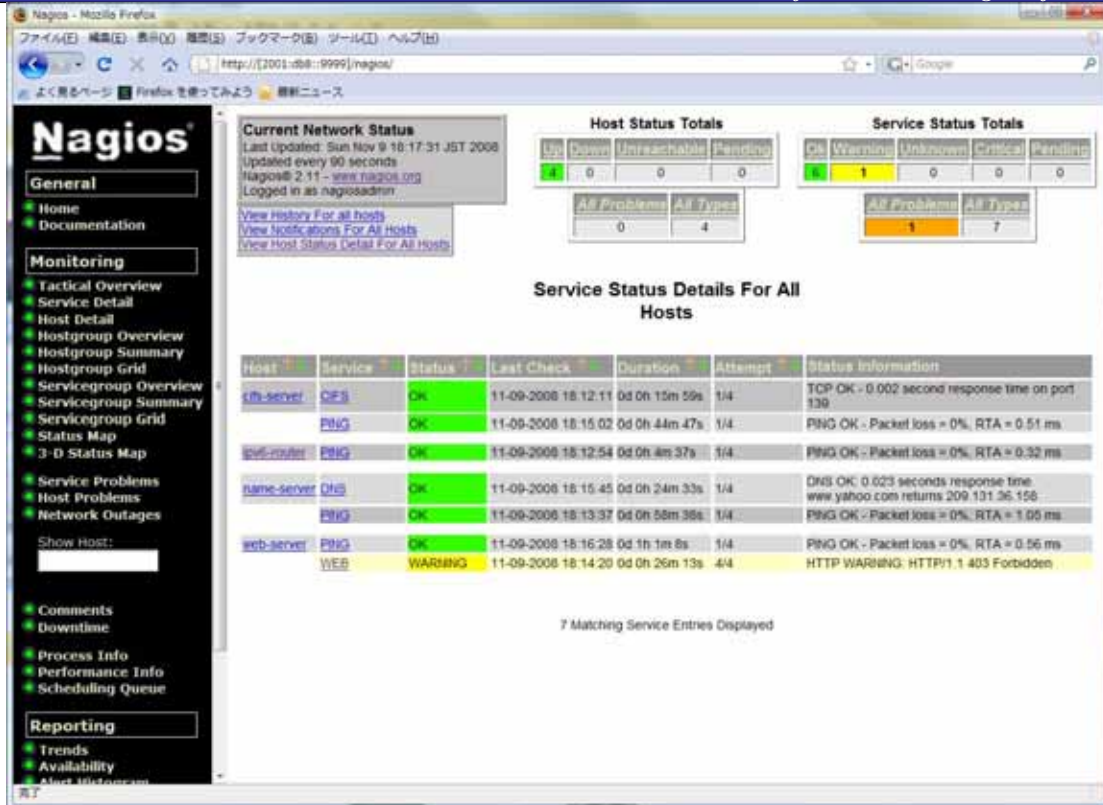
(デモ) mycompany.comのネットワークポロジ

監視対象機器

- ルータ郡
 - ipv6-router
- サービス郡
 - ns (ネームサーバ)
 - www (Webサーバ)
 - cifs (ファイルサーバ)



(デモ) nagiosのスクリーンショット



(実習) 監視対象機器の記述 /etc/nagios/servers.cfg の構成

監視対象の階層構造

- ホストグループ#1
 - ホスト#1
 - サービス#1 on ホスト#1
 - サービス#2 on ホスト#1
 - ...
 - ホスト#2
 - サービス#1 on ホスト#2
 - サービス#2 on ホスト#2
- ホストグループ#2
 - ...

(実習) 監視対象ホストの定義方法

```
define host{
    name generic-host ; サーバ共通テンプレ
    ...
}
```

継承

```
define host{
    use generic-host ; 全体テンプレを継承
    name linux-server ; Linuxサーバのテンプレ
    ...
}
```

継承

```
define host{
    use linux-server ; Linuxサーバのテンプレを継承
    host_name web-server
    alias www
    address 2001:db8::80
}
```

定義
済み

ホスト固有
の設定を
追記する

(実習) ホストグループの定義

複数のホストをひとまとめにした対象を定義

例1: ある組織のサーバ郡を監視

例2: メールサーバ郡をひとまとめに監視

ネームサーバ, Webサーバ, ファイルサーバが定義済みとする
name-server, web-server, cifs-server

```
define hostgroup {
    hostgroup_name Servers
    alias hanbai-ka-servers
    members name-server,web-server,cifs-server
}
```

(実習) ホスト上のサービスの定義

Webサーバの生死 (ping), Apacheプロセスの生死 (httpd) を監視

```
define service{
  use server-service ; サーバテンプレ(システム標準)を継承
  host_name web-server ; 定義済みのホストでサーバを指定
  service_description PING
  check_command check_ping!100.0,20!500.0,60%
}

define service{
  use server-service
  host_name web-server
  service_description WEB
  check_command check_http!/index.html!1!5
}
```

100ms遅延 または 20%ロスでWARNING
500ms遅延 または 60%ロスでCRITICAL
5秒遅延でWARNING
/index.html をGET 1秒遅延でWARNING

(実習) DNS, CIFSの監視

ネームサーバ上のBINDの監視

```
define service{
  use server-service
  host_name name-server
  service_description DNS
  check_command check_dns!www.mycompany.com
}
```

www.mycompany.com が
解決できなかったらCRITICAL

ファイルサーバ (TCPポート139反応) 監視

```
define service{
  use server-service
  host_name cifs-server
  service_description CIFS
  check_command check_tcp!139!2!5
}
```

TCPポート139へアクセスし 2秒遅延でWARNING
5秒遅延でCRITICAL

(実習) ホストリストの表示

「host detail」をクリック

定義したホストの一覧表示

Host	Status	Last Check	Duration	Status Information
lifs-server	UP	11-09-2008 17:55:18	0d 0h 40m 36s	PING OK - Packet loss = 0%, RTA = 0.58 ms
lifs-client	UP	11-09-2008 18:13:00	0d 0h 6m 36s	PING OK - Packet loss = 0%, RTA = 0.29 ms
lifs-server	UP	11-09-2008 17:48:08	0d 1h 0m 27s	PING OK - Packet loss = 0%, RTA = 0.39 ms
web-server	UP	11-09-2008 18:14:30	0d 1h 3m 7s	PING OK - Packet loss = 0%, RTA = 0.43 ms

4 Matching Host Entries Displayed

(実習) サービスリストの表示

service detail をクリック

定義済サービスの一覧を表示

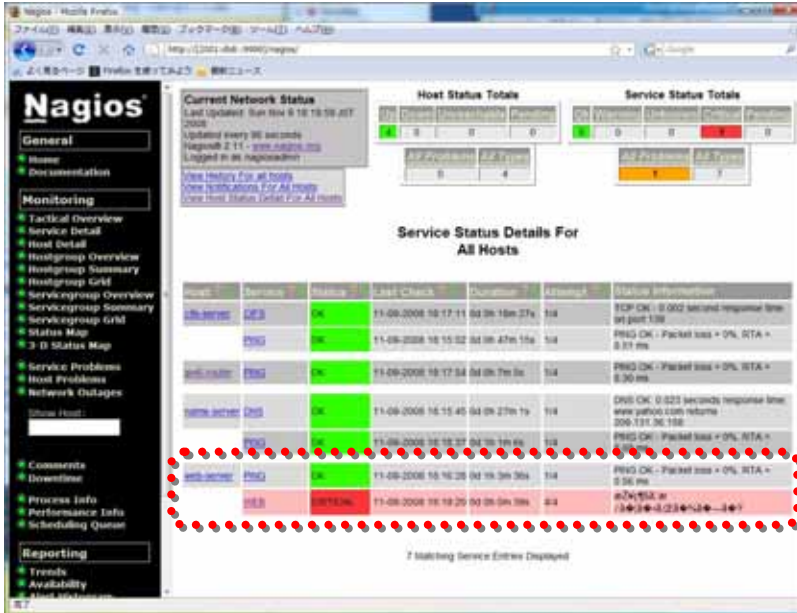
Host	Service	Status	Last Check	Duration	Attempt	Status Information
lifs-server	DEF	OK	11-09-2008 18:12:11	0d 0h 15m 56s	1/4	TCP OK - 0.002 second response time on port 139
lifs-client	PING	OK	11-09-2008 18:15:02	0d 0h 48m 47s	1/4	PING OK - Packet loss = 0%, RTA = 0.51 ms
lifs-server	DEF	OK	11-09-2008 18:12:54	0d 0h 4m 37s	1/4	PING OK - Packet loss = 0%, RTA = 0.32 ms
lifs-client	DEF	OK	11-09-2008 18:15:45	0d 0h 24m 33s	1/4	DNS OK: 0.023 seconds response time: www.yahoo.com returns 209.131.36.158
lifs-server	PING	OK	11-09-2008 18:13:37	0d 0h 58m 38s	1/4	PING OK - Packet loss = 0%, RTA = 1.05 ms
web-server	PING	OK	11-09-2008 18:16:28	0d 1h 1m 8s	1/4	PING OK - Packet loss = 0%, RTA = 0.56 ms
web-server	WEB	WARNING	11-09-2008 18:14:20	0d 0h 26m 13s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden

7 Matching Service Entries Displayed

(実習) サービスダウンの検出

Webサーバの停止

```
# /etc/init.d/httpd stop  
Stopping httpd: [ OK ]
```



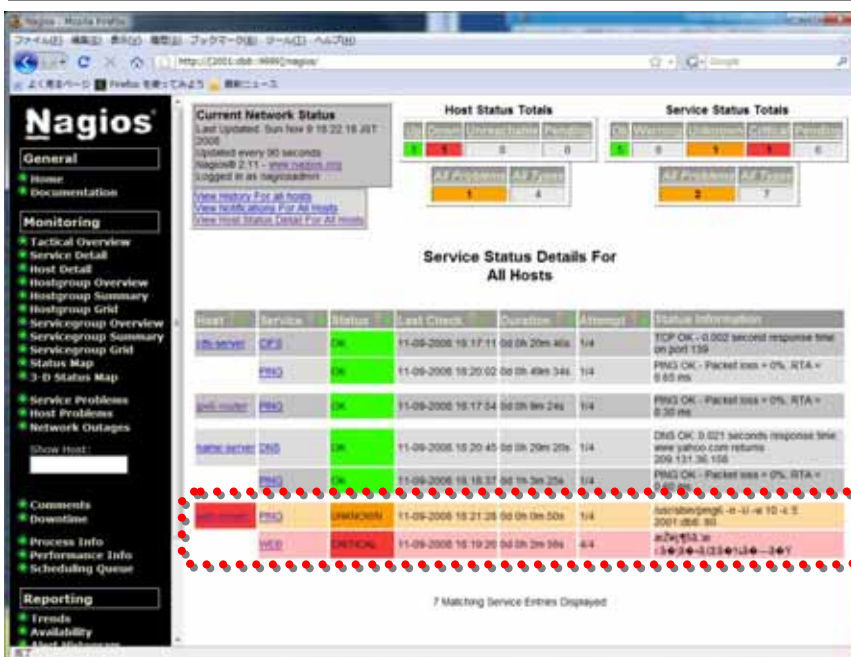
Webサーバ(サービス)の停止検出

ホストへのPING (IPv6接続)はOK

(実習) ホストダウンの検出

ホストダウンの発生 (IPv6アドレスの削除でダウンを模擬)

```
# ifconfig eth0 inet6 del 2001:db8::80/64
```



ホストのIPv6接続性ダウンとサービスダウンの両者を検出

障害検出時の通知方法

管理者へのメール送信

- Contactオブジェクトに
 - E-mail (通常の電子メール)
 - Pager (携帯電話向けメール)
- 複数のあて先を設定可

```
To: noc-admins@mycompany.com
From: nagios@mycompany.com

**** Nagios ****

Notification Type: PROBLEM
Host: web-server
State: DOWN
Address: 2001:db8::80
Info: CRITICAL - Plugin timed out after 10 seconds

Date/Time: Mon Nov 10 10:43:16 JST 2008
```

```
To: noc-admins@mycompany.com
From: nagios@mycompany.com

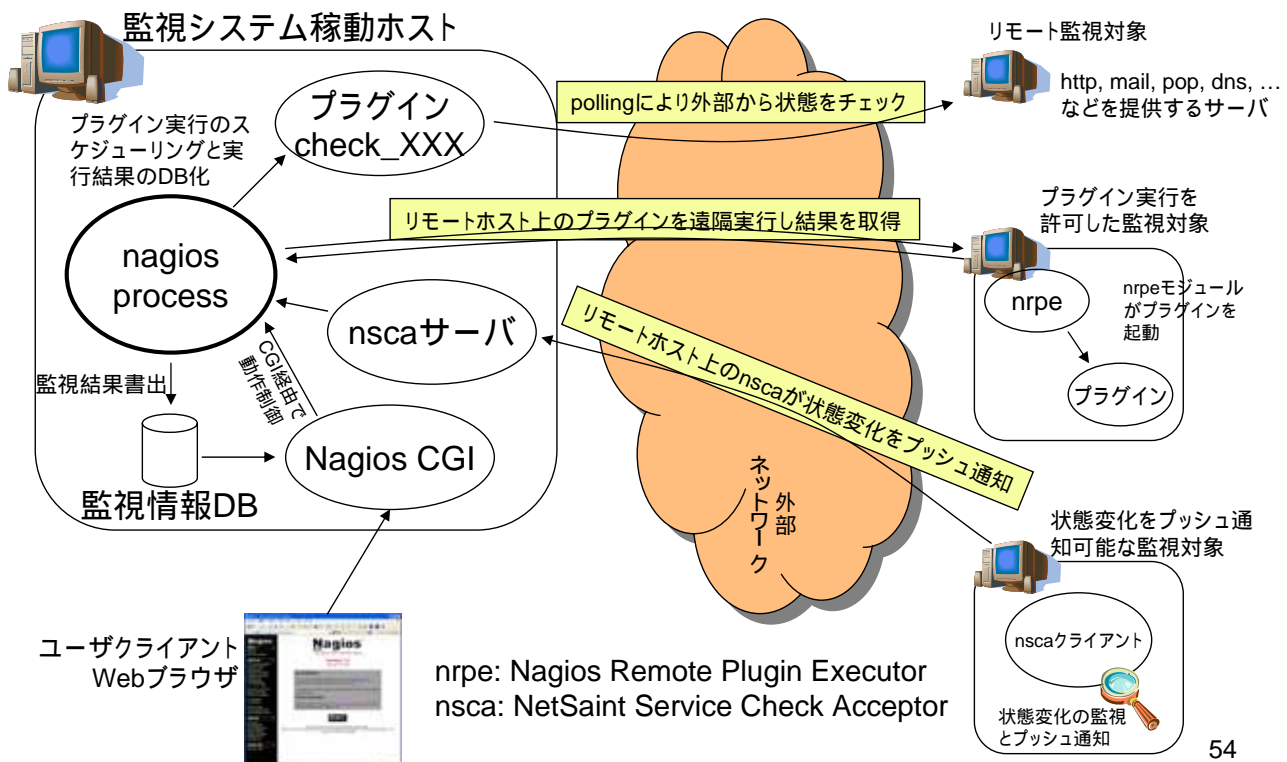
**** Nagios ****

Notification Type: RECOVERY
Host: web-server
State: UP
Address: 2001:db8::80
Info: PING OK - Packet loss = 0%, RTA = 0.83 ms

Date/Time: Mon Nov 10 11:03:39 JST 2008
```

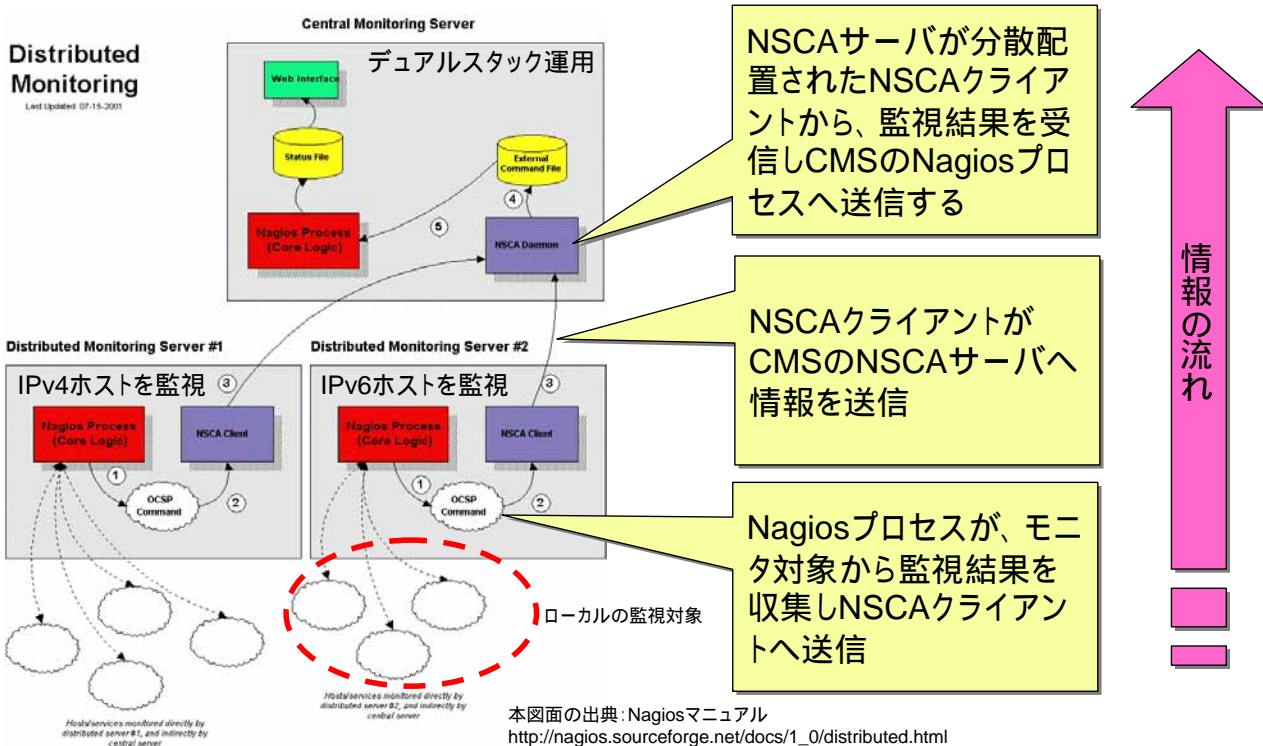
nagiosの監視モデル

より高度な監視へむけて



nscsを使った分散モデル

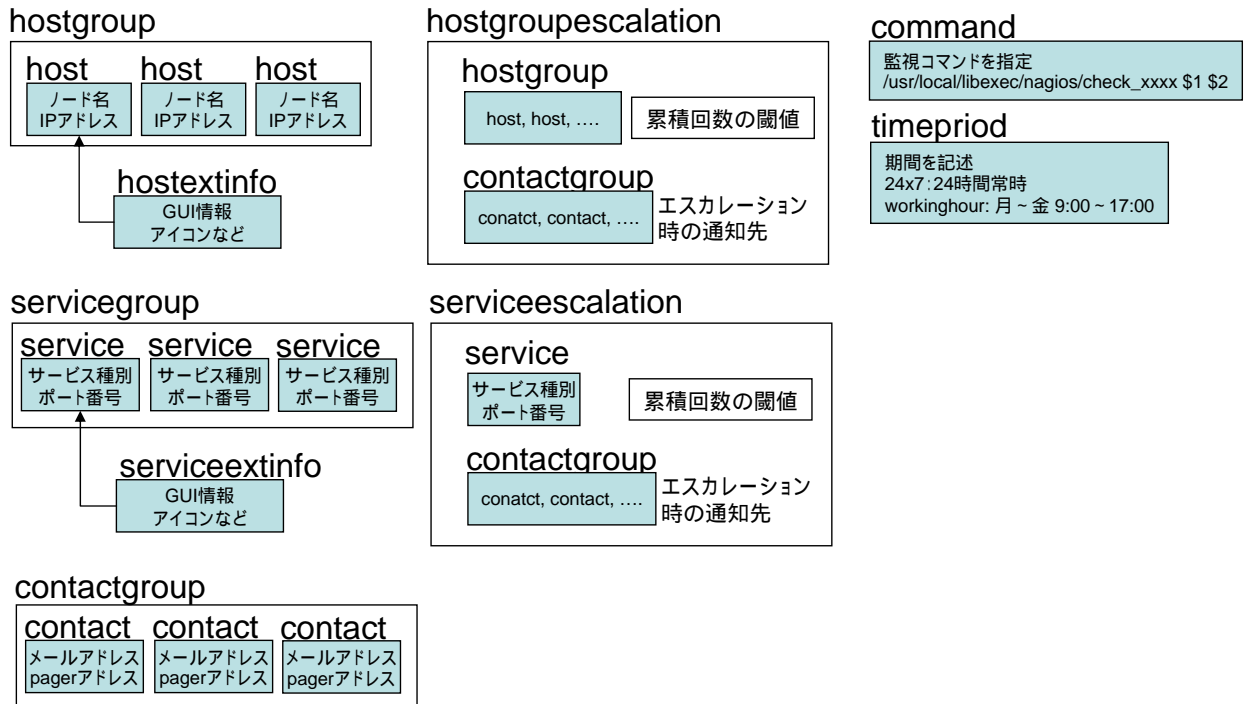
監視ドメインの異なるネットワークを統合的に監視



nagiosのオブジェクトの概念

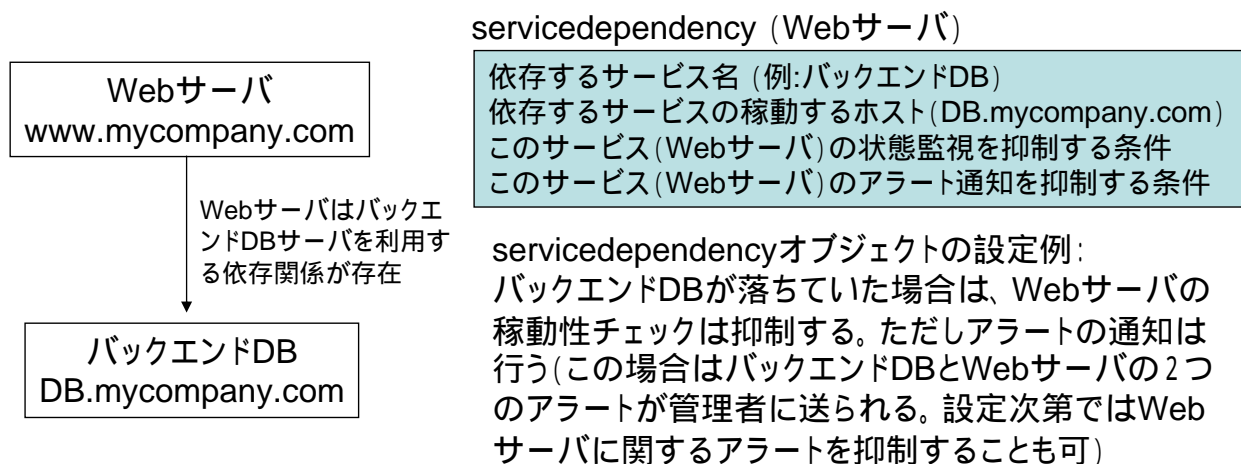
オブジェクト名	オブジェクトの内容
host	監視対象ノードを記述。通常1台のホストがこのエンティティに該当する
hostextinfo	GUIへのhost情報表示のための付加的な情報(アイコンなど)を記述
hostgroup	hostエンティティをグループ化したもの
hostgroupescalation	エスカレーション(ホストに関して障害が累積した)時の通知先を記述
hostdependency	ホストの依存関係を記述する
service	host上で提供されているサービス記述する。ひとつのhost上で複数のサービスが提供されていることもある
serviceextinfo	GUIへのservice情報表示のための付加的な情報(アイコンなど)を記述
serviceescalation	エスカレーション(サービスに関して障害が累積した)時の通知先を記述
servicedependency	サービスの依存関係を記述
contact	障害の検知時の管理者への連絡先を記述
contactgroup	contactgroupオブジェクトをグループ化したもの
command	監視データ収集のためのプラグイン(外部コマンド)のアクションを記述
timeperiod	期間を示すオブジェクト(例:24時間365日, 平日9:00 ~ 17:00)を記述

nagiosのオブジェクト関連図(1)



nagiosのオブジェクト関連図(2)

サービス・ホストの依存関係の定義



- 依存関係は設定が複雑だが切り分けには有用
- サービスと同様にホストの依存関係も記述可能