第 18 部

ネットワーク運用技術

第 1 章

gated を用いた経路制御

UNIX マシンを IP router として使用する場合、経路情報の交換が問題になる。4BSD では、経路情報交換を行う routed というプログラムが実装されており、RIP [123] を利用した経路制御を行うことが可能である。しかし、RIP 以外のプロトコルを使用している場合や複数の経路制御プロトコルが必要な場合、あるいは RIP のみを運用している場合でも経路のアナウンスを制御したい場合などは routed では十分ではない。

gated は Cornell 大学で開発されたプログラムで、RIP 以外の経路制御プロトコルをサポートしている他、どのインターフェースにどの経路をアナウンスするか、あるいはどのルータからのどの情報を信用するかというような細かな制御が可能になる。本稿は、gated の最新版である release 3 を利用する場合のメモである。

第 2 章

gated Release 3

1993 年 4 月末現在で、gated として正式にリリースされているバージョンは 2.1 である [1]。しかし、その後数多くの改良や新たなプロトコルのサポートなどが行われており、現在の最新版は R3_0Beta_2[2] となっている。gated に関する情報交換は、主にメーリングリストによって行われており、そのアドレスは次の通りである： [3]

gated-people@gated.cornell.edu [4]

gated R3 は次のような特徴がある：

- RIP および RIP-2 のサポート
- Hello のサポート
- EGP のサポート
- BGP version 2 および version 3 のサポート
- OSPF version 2 のサポート
- ISIS for IP のサポート
- SMUX インターフェースによる SNMP のサポート
- 経路情報の受理やアナウンスに関する制御（一部のプロトコルを除く）
- 動的なインターフェースのサポート

WIDE Internet の運用に於いては、RIP が主に用いられているが、Backbone 部分における経路制御では OSPF [27] への移行が始まっており、近日中に OSPF への移行が行われる。一方各組織と WIDE Backbone との間の経路情報交換には、当面 RIP がそのまま用いられることが予定されている。これらに関しては、別途述べる。

[1] 2.1 には 3 つのパッチが発表されている

[2] 1993 年 5 月 18 日現在

[3] α バージョンに対する議論やバグレポートなどは別の gated-alpha@gated.cornell.edu で行われている。

[4] 当然、参加申込みは -request に送る必要がある

また、WIDE 以外のネットワークとの経路情報交換は現在は RIP で行なわれているが、これは近い将来 BGP [126] に移行する予定で、そのための実験や調整が WIDE Project や JEPG/IP Routing WG で行なわれている。

SMUX [198] のサポートにより、SNMP [79] によるネットワークの管理をより具体的に行なうことができるようになった。例えば、OSPF mib 変数を参照することが可能になるし、またこれまで UNIX 上の SNMP daemon で各経路の metric を正しく表示しているものは少なかったが、gated では、ipRouteProto や ipRouteMetric1,ipRouteAge などの各変数にアクセスできるようになった。

動的なインターフェースのサポートは、ISDN などの常時運用されていないリンクが存在する場合に有用な機能である。しかし専用線を利用している場合でも、各インターフェースの定義をしておくことにより、当該インターフェースが活動状態でなくても gated は問題なく実行を始める[5]という点で大きく改善されているということができる。

[5]従来のバージョンでは、活動状態にないインターフェースに関する記述— accept interface や propagate interface など — がある場合、gated はエラーとして処理を停止してしまっていた。

第 3 章

gated のインストレーション

gated 自身のインストレーションは、Config ファイルにシステムに関係する種々のオプションや設定を記述するだけで比較的容易に行うことができるが、前準備としていくつかのソフトウェアのインストレーションが必要になる。

3.1 ISODE-SNMP のインストレーション

SMUX インターフェースを利用した SNMP のサポートを組み込むためには、gated に先だって ISODE-SNMP のインストレーションが必要になる。ISODE は沢山のライブラリやモジュールからなる巨大なパッケージであるが、SNMP に必要な機能のみを取り出したアーカイブを利用するのがてっとリ早い [1]。

SNMP パッケージの READ-ME に従って必要な設定を行う。make に先立ち、./make once-only を実行する必要があるが、完全な ISODE パッケージではなく SNMP 部分を利用している場合には、エラーになるが、気にせずに先に進む。./make の実行が終り次第、SNMP の作成に取りかかる前に ./make inst-partial を実行しておく。そして、SNMP の作成し（./make all-snmp）インストールする（./make inst-snmp）。

実行に先立ち、*ETCDIR*/snmpd.rc にコミュニティーやアクセス範囲、管理者やサイト情報を更新しておく。また、gated との関係を定義するために、*ETCDIR*/snmpd.peers に次の一行を加え、root のみが読み出しできるようにしておく：

```
"gated"    1.3.6.1.4.1.51      "password"
```

そして、/etc/services に次のエントリが含まれていることを確認する：

```
snmp        161/udp
snmp-trap   162/udp
smux        199/tcp
```

gated がこの SMUX インターフェースを利用して SNMP の問い合わせを処理するためには、gated より先に snmpd を起動する必要がある。そのため、rc.local には次のような指定をする：

[1] gated.cornell.edu:pub/gated/isode-snmp-7.0.tar.Z
あるいは sh.wide.ad.jp:routing/isode-snmp-7.0.tar.Z を利用することができる。この場合でも、isode-snmp-7.0.tar.Z を除いて、約 24MB の作業領域が必要である。

```
### Start SNMP daemon(s)
if [ -f /usr/etc/snmpd ]; then
    snmpd > /dev/null 2>&1 && echo -n ' snmpd'
    if [ -f /usr/etc/smux.unixd ]; then
        smux.unixd > /dev/null 2>&1 && echo -n ' smux.unixd'
    fi
fi
### Start routing daemon
if [ -f /usr/etc/in.gated -a -f /etc/gated.conf ]; then
    in.gated && echo -n ' gated'
else
    in.routed && echo -n ' routed'
fi
```

3.2　UDP のチェックサム

gated は Release 3 から、UDP [199] のチェックサムを使用する設定になっていない場合、RIP が動作しないようになっている。そのため、kernel 中の UDP のチェックサムを ON にしておかなければならない。

SunOS4.0 や NewsOS の場合

Kernel 中の _udpcksum を adb などを使用して 1 にする。このとき、稼働中の kernel だけではなく、/vmunix や後日 kernel の再構成を行った場合に設定を忘れないように udp_usrreq.o に含まれる同変数も同時に変更する。

SunOS4.1 の場合

Kernel と /vmunix の _udp_cksum を 1 にする。また、sys/netinet/in_proto.c に含まれる

```
int udp_cksum = 0; /* turn on ...
```

を

```
int udp_cksum = 1; /* turn on ...
```

に変更しておく。

3.3　Multicast

gated Release 3 では、IP マルチキャストがサポートされている場合、OSPF を運用す

ることができるし、RIP では従来の RIP に併せて RIP2[2][200] の運用も可能になる。この両方の場合とも、マルチキャストはルータを超えて伝搬する必要はないので、マルチキャストの経路制御の機能は不要である。従って、gated の為に mrouted を走らせる必要はない。

マルチキャストは MBONE や vat の実験のために開発されたコードが入手可能であり、以下の OS ではソースコードが無くてもマルチキャストを実装することができる[3]：

アーキテクチャ	OS	Interface
sun3	SunOS 4.0	ie, ec
sun4	SunOS 4.1.1	ie, le
sun4c	SunOS 4.1.x	ie, le

それぞれのパッチは sh.wide.ad.jp:vmtp-ip から入手することができる。SparcStation などの sun4c アーキテクチャの場合、ipmultisunos41x.tar.Z を入手し、展開する。そして .diff を用いて、該当するファイルにパッチを当てる。このとき、ソースコードを持っていない場合には、一部の .c ファイルのパッチは無視してよい。そして、.o ファイルを sys/sun4c/OBJ にあるそれぞれのオリジナルファイルと交換する。そして、config ファイルに次の 2 行を加え、kernel を再構成すればよい：

```
options MULTICAST
options MROUTING
```

Sun 用のパッケージには、マルチキャスト対応版の ifconfig や netstat, ping のバイナリも含まれている。これらは /usr/etc/mifconfig のようにインストールしておくと便利である[4]。

3.4 Reject & Blackhole インターフェース

ルータでパケットの経路制御を行う場合、当該パケットを廃棄したり、対応する ICMP unreachable を返したりする指定が必要な場合がある。国内のインターネットでは、海外の経路を default route で表現している。ここで、国内のあるネットワークへの到達性が失われ、経路情報が欠落した場合には、この default route が適用されて、最寄りの default route に依存しないルータまで運ばれ、そこで ICMP が返されることになる。特に WIDE Internet の場合には、NASA Ames で ICMP になるので、国際リンクの帯域を無駄にしてしまうことになる。このような場合、default route を適用して欲しくない

[2] RIP の空いているフィールドに Tag や Nexthop Gateway などを含め、経路情報の伝達にマルチキャストを利用するなど、RIP を改良したプロトコル。残念ながら、metric max は 16 と変わっていない。

[3] また、NewsOS 4.2 でもマルチキャストを可能にするモジュールが作成されているが、OSPF との動作の確認はまだ行われていない。

[4] vat などの MBONE に関係する実験を行う場合、このモジュールに含まれている mrouted では正常な動作が得られない場合があるので、mrouted は最新版を使用されることをお勧めする。

経路に対しては、経路情報が届いていない場合、ICMP unreachable を発生するように制御することが必要である。

このような機構は 4.4BSD では実装されているが、それ以前のものに関しては同様の動作を行う Reject 仮想インターフェース（ICMP unreachable を発生する）、Blackhole 仮想インターフェース（パケットを単に廃棄する）が作成されている[5]。このモジュールは `sh.wide.ad.jp:routing/reject.tar.Z` として入手可能である。

`gated` Release 3 では、`interfaces` に指定することによってこれらのインターフェースがサポートされている：

```
interfaces {
     interface 127.0.0.2 reject;
     interface 127.0.0.3 blackhole;
};
```

3.5　Point-to-point インターフェース

WIDE Internet では、WIDE Backbone と各組織を結ぶ point-to-point インターフェースは、原則としてそのリンクに別個のアドレスを割り振らない、いわゆる unnumbered で運用している。この場合、point-to-point リンクの両端には、それぞれのルータの代表的な（Ethernet インターフェースに使われている）IP アドレスを割り振る。`gated` は point-to-point インターフェースの識別は remote 側アドレスで行っているので、同一ルータ間に複数のリンクを設定しない限り問題にはならない。

ところで、point-to-point インターフェースに与える netmask であるが、従来は、local 側アドレスの属すネットワークの netmask を指定していた。例えば、`133.4.3.2` — `133.2.1.1` というリンクに関して、`133.4.3.2` 側のルータでは、`133.4` の netmask `255.255.255.224`（`0xffffffe0`）を与えていた。

Release 3 の `gated` では、point-to-point リンクの netmask は natural mask（subnet を使用しないときの mask, Class B では `255.255.0.0`）あるいは host mask（netmask `255.255.255.255`）を使用するので注意が必要である。これは、`gated` の作者が point-to-point リンクはネットワークではなくトンネルであると考えていることによるものである。

WIDE Backbone のルータでは、host mask で運用を行っているが、multicast のサポートの都合で SunOS4.0.3 で運用しているルータは natural mask で設定されている。

`gated` release 3 を使用する場合、あるネットワークの subnet が point-to-point リンクに割り当てられている場合には注意が必要である。そのネットワークの別な subnet が Ethernet などの point-to-point インターフェースに割り振られている場合には問題ない。しかし、そのような Ethernet インターフェースが存在しない場合、経路情報の取扱い上問題が発生する。

[5] 以前は unreach インターフェースと呼ばれていた。

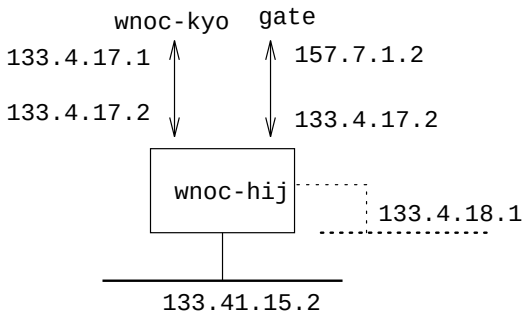

図 3.1: 広島 NOC の構成

図 3.1 は、WIDE 広島 NOC の構成であるが、WIDE Backbone のアドレス 133.4 は、京都 NOC への point-to-point リンクのみにその subnet が割り当てられている。放射線影響研究所へのリンクは unnumbered であり、Ethernet 側は広島大学の subnet である。このような場合、gated は 133.4.0.0 の subnet 経路を送出するが、ネットワーク全体に対する経路を送出してくれない。

この問題は、図 3.1 の点線のように Broadcast 型の仮想インターフェースを設定し、133.4 の異なった subnet アドレスを割り当てることによって回避することができる。この仮想インターフェースはアドレス割り当ての対象となるだけで、実際のパケットは単に廃棄するようなコードで差し支えない。

3.6 gated のインストール

gated の最新版を入手し、unpack する。実際に make する作業領域を含めると、tar.Z ファイルを除いて約 13 MB のスペースが必要になる。

src ディレクトリに移動し、最初に make に必要なディレクトリを作成する。ディレクトリ名はマシンの OS のバージョンや CPU アーキテクチャを含んだものになる。従って、NFS などで共有して、異なった OS や異なった CPU アーキテクチャ用のバイナリも並行して作成可能である。必要なディレクトリは、

```
obj.‘util/archtype‘
```

である。Sun の場合、obj.SunOS-4.1.2-sun4 などとなる。

次にどのようなプロトコルをサポートする必要があるのか、カーネルとのインターフェースのとり方、などを指定した Config ファイルを obj ディレクトリに作成する。

Config ファイルの例は configs ディレクトリにいくつか含まれている。実際に完全にフィットしたものはなくても、configs/README にパラメータの指定法が書かれているので、それを参考に作成することも可能である。Sun の場合、gcc を使用したものが含まれているが、gcc なしでも問題はない。この場合の Config を付録 5.1 に示す。

Config を作成したら、src ディレクトリで

```
% make config
```

を実行する。これによって、必要なシンボリックリンクを設定し、指定されたプロトコルをサポートするように、gated.conf のパーザや Makefile が生成される。

次いで、src ディレクトリあるいは obj ディレクトリで

```
% make
```

によって、gated および附属のユーティリティを作成することができる。

gated はカーネルライブラリの関係で static link されている。そのため実行モジュールは比較的大きくなり、RIP, OSPF, BGP, SNMP をサポートするように指定した場合で約 1.8MB である。また、カーネルライブラリが static link されているため、OS の revision が異なる場合には別途 make した方が安全である。

作成された実行ファイルは obj ディレクトリにある。これを適当なパス（例:/usr/etc/in.gated）にインストールする。また、boot 時に起動されるように /etc/rc.local などを編集する必要があるが、SNMP を使用する場合には snmpd の後に起動するが、このあたりは 3.1 を参照されたい。

3.7 syslog

gated は gated.conf の fatal でないエラーやその他の情報を syslog に送る。そのため、BSD 系のマシンならば、/etc/syslog に

```
daemon.err;daemon.info      /var/log/syslog
```

という記述を含めることによって、これらの情報が syslog に出力される。この場合、/var/log/syslog を適当に aging する設定にしておかないと、ファイルシステムが溢れる危険性があるので注意が必要である。

3.8 gated の運転

gated を動かすためには、正確な gated.conf ファイルを作成することが必要である。文法の誤りがあった場合、正しい動作は期待できない。従って、gated.conf を書き換えた場合、必ず -c オプションを指定して gated.conf の文法チェックを行なうようにすることを勧める。最近の gated は fatal でないエラーはそれを無視してそのまま走るように設計されているので、しばしば管理者の意図通り動作していない場合があり、チェックは重要である。

gated が正常に立ち上がった場合、/etc/in.gated.pid、/etc/in.gated.version にそのプロセス番号とバージョン情報をそれぞれ記録する[6]。

[6] これらのファイル名は Config の指定や gated の実行モジュールのファイル名によって変わるので注意が必要である。

OSPF を ON にしていない場合、gated.conf の修正を反映させるためには、gated に HUP シグナルを送ることによって、再構成が可能である：

```
# kill -HUP `/etc/in.gated.pid`
```

しかし、OSPF を実行中は再構成の機能は働かず、gated プロセスが終了してしまうので注意が必要である。この場合には、gated を kill して再起動する必要がある。

pid ファイルは lock されており、複数の gated が同時に走らないようになっている。もしプロトコルのテストなどのため、カーネル中の経路表を変更せず、プロトコルにだけ参加してみる場合には、-n オプションを付けるとともに、実行モジュールを別な名前で起動できるように link を張っておく必要がある。なお、gated が終了した場合には pid ファイルは消去される。

gated プログラムは経路制御上重要であるから、もし何らかの原因で異常終了した場合、すぐに再起動することが必要になる。そこで、version ファイルを利用して、例えば次のような script を 5 分程度おきに起動すると便利である：

```
#!/bin/sh
#
if [ -f /etc/in.gated.version ]; then
     if kill -WINCH `sed -n 's/pid \(.*\),.*$/\1/p'` 2>/dev/null
     then
          :
     else
          /usr/etc/in.gated
     fi
else
     /usr/etc/in.gated
fi
```

OSPF のデバッグには、内部の経路表の状態やアナウンスの状況を知る必要がある。ある程度は、SNMP で問い合わせを行なうことができるが、より細かい情報が必要な場合には、内部の dump は gated.conf のデバッグに参考になる（4.13 参照）。この場合、gated プロセスに INT シグナルを送る。gated は SIGINT を受けとった場合、速やかに fork し、子プロセスに内部状態をユーザに理解できる形式でダンプを出力させる。ダンプは、Config ファイルの指定にもよるが、/var/tmp/gated_dump が多く用いられている様である。

ダンプは、gated_dump に追加される。日本のインターネット全ての経路を得ている場合には、300kB を越えることがあるので、事前に消去しておいたほうがよい。gated.conf の文法チェックのため、-c オプションを使用した場合にも、fatal な文法エラーのない限り、その状態を gated_dump に追加するので、不要になったら消去するのを忘れないようにしたい。

第 4 章

gated.conf の書き方

4.1 概論

gated は複数の経路制御プロトコルをサポートし、プロトコル相互の経路情報の交換やフィルタリングを可能にしている、比較的大規模なプログラムである。

gated は各種経路制御プロトコルから学習した経路を内部の経路表に登録する。そして、 preference によって優先順位を付け、同一経路に対しては最も preference の高い[1] 経路を採用する。同一プロトコルから複数の経路を学習した場合には、そのプロトコルにおけるコスト（cost あるいは metric）の最も小さなものを採用する。

gated は採用した経路を kernel 中の経路表に登録し、その経路情報が実際の経路制御に使用されるようにするとともに、gated.conf の指定に従って経路制御プロトコルによってアナウンスを行う。従って、採用していない経路はアナウンスされない点に注意が必要である。

経路の入力は、経路制御プロトコルに依存するが、distance-vector 型のプロトコルの場合、経路を送出したゲートウェイ、経路が到着したインターフェース、および対象経路によって、その経路を考慮の対象にするか、捨てるかを指定することができる。これらは import 節に指定する。OSPF などの link-state 型のプロトコルは、すべてのゲートウェイが同一のデータベースから計算された経路表に基づいて経路を計算することになっているので、このようなスクリーニングは設定できない。

あるプロトコルに経路をアナウンスする場合、採用された経路がどのプロトコルから学習されたものであるかということを指定する。これらは送出するプロトコル毎に export 節で指定する。この場合も OSPF に対しては、経路が採用された場合には自動的にアナウンスがなされ、それをスクリーニングすることはできない。

以下、gated.conf の書き方を概説する。また、WIDE Backbone で使用されている gated.conf のいくつかを参考のため添付するが、後述のように WIDE Backbone は経路制御プロトコルを OSPF にするとともに、外部との経路情報交換に BGP を使うべく種々の実験が行われているので、現在の設定とは異なる。また、実際の運用に関して、WIDE Backbone との経路情報交換に関しては、別に述べるので、それに従った設定をお願いしたい。

[1] gated.conf に指定する数字の最も小さなものを、ここでは preference が高いと表現する。

4.2 コメント

gated.conf では、# 以降はコメントとして扱い、処理の対象とはしない。また、C におけるコメントの /* ··· */ もサポートされているので、使い分けると便利である。コメントは gated.conf のどこにでも書くことができる。

4.3 ファイルの include

別なファイルの内容を引用するためには、%include "*filename*" を使用する。特にマニュアルには記されていないが、現在のバージョンでは、% 記号は行の先頭になければならないという制約がある。Include するファイルの default ディレクトリは%directory "*directory*" で指定することができる。これらの指示はファイルの任意の場所で可能である。

WIDE Backbone では、BGP への移行が完了するまでは、RIP などで与えられた経路に tag を付け、経路の出典を表示し、これによってアナウンスする経路の metric の設定を行っていた時期があった。この設定は、新規のネットワークの接続によって変更する必要があり、一貫性を失わないように変更を行わなければならない。そのため、どの経路はどのネットワークに属しているかという情報を集中的に生成し、各ルータでこのファイルを適宜 include することによって参照を行っている。

4.4 トレースに関する指定

gated は実行中にさまざまな情報を出力することができる。これらをうまく使うと、経路制御の運用に関するデバッグに便利である。従来はトレースファイルは単調に増加してしまうものであったが、Release 3 からはトレースファイルがある程度成長した場合、別なファイルにトレースを出力する機能がある。これの指定は、例えば次のように行う：

```
tracefile "/var/tmp/gated_trace" replace size 100k files 10;
```

この設定では、トレースファイルが 100kB に達すると、そのファイル名を ".0" が付加されたものに変更し、新しいファイルにトレースを行う。従って、上の設定の場合、約 1MB の領域が確保されていればよい、ということになる。

4.5 経路の取り扱い

gated の基本的な動作は、各経路制御プロトコルから得られた経路を、内部の経路表に記入し、選択された経路を各経路制御プロトコルによってアナウンスするということである。一方、実際のパケットの転送に使われる UNIX カーネル内部の経路表に対して、選択された経路の書き込みを行ない、パケットの経路を制御する。

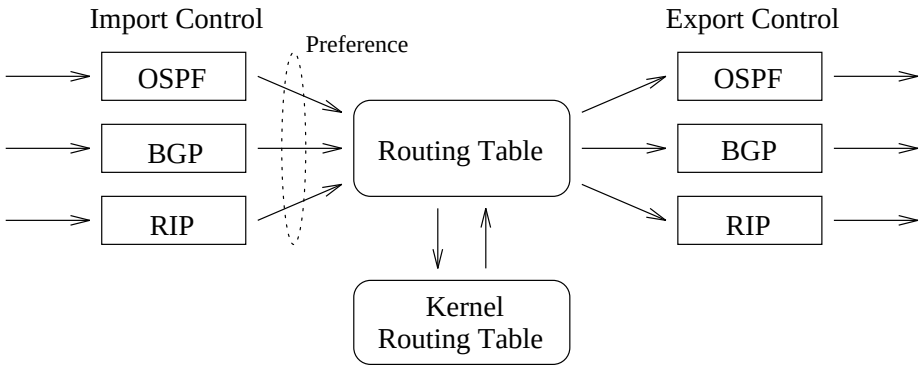

図 4.1: gated における経路の取り扱い

gated では各種の経路制御プロトコルがサポートされている。ある経路が複数の経路制御プロトコルから学習されたという状況も起こり得る。この場合 gated は、その経路制御プロトコル毎に割り当てられた *preference* によってどの経路を採用するかを決定する。Preference は 0 ～ 255 の整数であり、数字の小さなものほど優先される。gated の default で設定されている各種経路制御プロトコルの prefenence は表 4.1 に示す通りである。

表 4.1: gated の preference

プロトコル	Preference	プロトコル	Preference
Direct	0	RIP	100
OSPF	10	OSPF ASE	150
Redirect	30	BGP	170
SNMP	50	EGP	200
Static	60	Kernel	254
HELLO	90		

言うまでもないが、同一経路制御プロトコルで複数の経路が存在する場合には、そのプロトコルで定義された metric あるいは cost によって選択される。例えば、RIP metric 5 の経路、RIP metric 8 の経路、OSPF の経路が存在する場合、preference によって OSPF 経路が採用される。しかし、もし OSPF 経路が存在しなければ、RIP metric 5 の経路が採用されることになる。

採用された経路は、実際のパケットの転送に使用するため、カーネル中の経路表に書き込まれる。また、この経路は条件にしたがって各種経路制御プロトコルで広告される。採用されていない経路は広告されることはないという点に注意が必要である。

Preference を変更するときには注意が必要である。また、他のルータでは、プロトコル毎の preference （あるいは distance）の順位が異なっている場合があるので、複数の

プロトコルを同時に異なったベンダのルータで走らせる時には細心の注意が必要である。さもなければ、簡単に routing loop が発生する。

4.6 gated.conf

gated の挙動は gated.conf によって指定される。gated.conf は、gated を make するときのコンフィギュレーションに依存するが、/etc/gated.conf あるいは /etc/in.gated.conf などのようなパスにおくことが多い。

gated.conf には、trace に関する指定の他に、ファイルの include の指定やディレクトリの指定などは、どこにあっても構わないが、それ以外の指定は順序が次のように決まっており、その順番を守らなければならない：

1. オプション — 全体に
2. インターフェース — インターフェースの指定
3. 定義 — オプションや AS 番号などの指定
4. プロトコル — 各種経路制御プロトコルのパラメータの指定
5. 経路 — 静的な経路の設定
6. 制御 — 経路制御に関する経路の受理や公開に関する指定

gated.conf には、オプション noresolv によって DNS lookup を禁止することができる。gated.conf で指定するアドレスは通常一意に定まっている必要があるため、ゲートウェイを symbolic な名前で記述することは避けるべきで、アドレスによって指定することが好ましい。一方インターフェースの指定は、slip0 などのインターフェース名で行なう方法とそのインターフェースのアドレス（pointopoint インターフェースでは remote 側のアドレス）で行なう方法がある。この場合も、インターフェース名で指定した場合には、DNS lookup を行ない、fail した場合にインターフェースリストを検索する。従って、gated が立ち上がる時には DNS が稼働している必要があるが、経路が確立していない時点では lookup がハングアップすることがある。そのため、noresolv を指定することでこれを回避することができるが、インタフェースの指定もすべてアドレスによって行なうことをお勧めする。

gated.conf は、% で始まる指示や # で始まるコメントを除き、一つの記述は複数の行に跨っていても差し支えない。また、tab や空白、空行は無視されるので、indentation などに工夫をして読み易い記述をおこなうことが肝要である。一般に一つの記述は、セミコロン ; で終了する。

4.7　インターフェースの指定

gated.conf では、インターフェースを指定する場合、インターフェース名（le0 など）あるいはそのインタフェースに割り当てられているアドレスで指定する。前述のようにDNS 問題があるため、アドレスを直接指定することが好ましい。Point-to-point インターフェースの場合、アドレスは local 側アドレスではなく、remote 側のアドレスを指定する。このことによって、unnumbered point-to-point リンクの場合でも、同一ゲートウェイに複数のリンクが存在しない限り問題なく区別をすることができる。

gated.conf は自動的にインターフェースを検出し、各プロトコルの指定や入出力の指定に従って経路制御プロトコルを実行する。gated.conf でこれらの指定に引用されたインターフェースは、gated が起動されたときに存在し、活性化されていない場合にはエラーになる。ISDN などを利用しない場合でも、回線や対抗ルータがダウンしている場合などはインターフェースが活性化されていない。これらのインターフェースに対する記述が gated.conf に含まれていればエラーになる。

そこで、これを防ぐため、各インターフェースに関する定義をしておくことが望ましい。インターフェースの定義は define を用いて、アドレス、boradcast 型の場合はブロードキャストアドレス、pointopoint 型ならばリモートアドレス、さらに netmask や multicast が可能かどうかを指定する。

また、経路制御に RIP を使用している場合、相手が RIP パケットを送出しなくなった場合、gated はインターフェースがダウンしていると判断し、RIP パケットの受信が再開されるまで RIP パケットの送出を停止する。この機能は、時おり双方の gated がお互いをダウンであると判定してしまう可能性がある。これを防ぐのは、インターフェースオプション passive を指定する。一般に WIDE 参加組織の WIDE Backbone に対するルータにおいては、WIDE Backbone 側のインターフェースを passive に指定しておいた方が良い。

インターフェースオプション reject と blackhole については 3.4 に述べた通りである。

4.8　定義

gated が EGP や BGP に参加する場合、autonomoussytem に AS 番号を指定することが必要である。また、BGP や OSPF に参加している場合、経路情報の発行の際に使用する routerid は、gated は自動的にルータの持っている適当なアドレスを使用するが、陽に指定する場合には、routerid で指定する。BGP も OSPF も使用しない場合、特に定義部を記述する必要はない。

4.9 プロトコルの定義

ここでは各プロトコル別に、どのインターフェースに対して経路制御プロトコルを実行するのか、隣接ルータやバージョンの指定などを行なう。無論、各プロトコルが起動されるためには、gated を make する際に指定しておかなければならないが、RIP や SNMP を除いて各プロトコルは陽に on を指定しなければならない。RIP（と SNMP）のみを使用し、特に RIP の挙動に指定をしない場合には、プロトコル定義部は空でも差し支えない。

4.9.1 RIP

RIP は default では ON になっている。従って、RIP を送る gateway を制限したい場合や、特定のインターフェースに対して RIP を受けとらない、あるいは送らないなどの制御を行なう場合以外は rip 節は不要である。

特定のインターフェースに対して RIP の受信あるいは送信を禁止する場合には、import あるいは export で全ての経路を無視するような指定をしても差し支えないが、そのインターフェースに対して noripin, noripout を指定した方が簡単である。

4.9.2 OSPF

OSPF は default では OFF である。従って OSPF で経路制御を行なうためには、ospf 節にコンフィギュレーションを指定することが必要である。WIDE Backbone は OSPF で運用されているが、WIDE 参加組織が OSPF に参加する場合には事前に調整が必要であるので、経路制御担当と相談して欲しい[2]。

OSPF は、各エリアごとに経路情報交換を行なうインターフェースを指定する。一つのインターフェースが複数のエリアに属すことはできない。インターフェースの指定では、そのインターフェースに対するコストを指定することができる。コストは複数の経路が存在する場合、コストの合計が小さい方の経路が選択される。Cisco のマニュアルでは、

$$\frac{10^8}{bandwidth}$$

を指定するとよい、としている[3]。また、point-to-point インターフェース以外には、designate router になる優先度を決める priority を指定する必要もある。

エリアの設定であるが、WIDE Backbone 部分を backbone エリアとし、各組織は独立したエリアを構成する、という運用を行なっている。各組織が point-to-point ネットワークで接続されている場合での運用はまだ開始していないため[4]、詳細は未定であるが、

[2]WIDE Backbone 側のルータは、Backbone 以外のインタフェースは OSPF を disable にしているし、パスワードの設定も必要なので内緒で参加することはできない。

[3]このコストは、FDDI を 1, Ethernet を 10 としており、1.5Mbps で 67, 768kbps で 130, 384kbps で 260, 192kbps で 521, 64kbps で 1563, 9.6kbps で 10417 となる

[4]1993 年 6 月初旬現在では、Ethernet で接続されている組織のみが OSPF internal で制御され、その他は RIP から変換された OSPF ASE として取り扱われている。

WIDE Backbone 側のゲートウェイが backbone エリアと各参加組織のエリアの両方の経路制御に参加する形態になると思われる。そして、特に必要でないかぎり、stub エリアとして定義されることになると予想されている[5]。

OSPF では、パスワードによる authentication が多くのルータで実装されている。gated では、エリア毎に authentication を行なうかどうかを指定し、インターフェース単位でパスワードを指定することができる。パスワードは 8 文字以下の文字列を使用する。エリアの指定の一例は次のようになる：

```
area 133.27.0.0 {
     authtype simple;
     interface 133.27.48.5 cost 10 {
          enable;
          authkey "abcdefgh";
          priority 10;
     };
};
```

gated の OSPF では、internal な経路（type 2）と external な経路（type 5）は区別して取り扱われる。前者は ospf と記され、後者の経路は ospfase として指定される。本稿では、後者を OSPF ASE と表現する。Proteon や Cisco では、OSPF ASE の type によって、E1, E2 のように表示している。

4.9.3 BGP

BGP は AS 間の経路情報交換を行なうプロトコルであり、WIDE 参加組織と WIDE Backbone の間で BGP を使用することはほとんどない。一部の参加組織で、WIDE とは別の AS を持っている場合で、管理上のいくつかの状況によっては BGP を使用する必要がある場合がある。

BGP を使用する場合には、/etc/services に次の一行が必要である：

```
bgp       179/tcp
```

gated の BGP は、group 別に相手側 BGP speaker を指定する。group は一般に使用されるものとしては、次のようなものがある：

external
: 隣接する AS との間での経路情報交換を行なう。複数の隣接 AS が存在する場合、複数の external group を定義する。

[5] stub エリアに対しては、default 経路と OSPF Internal 経路のみが公開され、OSPF ASE 経路はアナウンスされない。

igp
: IGP（現在は OSPF のみがサポートされている）を介して BGP の Internal な交換を行なう場合に使用する。BGP と OSPF の関係は BGP OSPF Interaction[201] に基づいており、AS Path が OSPF tag で表現できない場合（隣接 AS 以遠からの経路）にのみ IBGP で Path attribute を交換する。igp に指定された BGP speaker からの情報は、OSPF tag から BGP Path attribute を生成するときにのみ参照され、IBGP からの経路がアナウンスされることはない。

internal
: IBGP ですべての経路を交換する必要がある場合に使用される。当初はネットワークを共有している範囲でしか使用できなかったが、peer 毎に gateway を指定することによって、この gateway 経由であっても差し支えない。この場合、lcladdr に指定する local address は、gateway に向かう方向のインターフェースのアドレスを使用し、対抗する BGP speaker にもこのアドレスを指定する必要がある。現在の cisco のソフトウェア[6]が IBGP Speaker である場合には igp ではなく internal group を使用する必要がある。

各グループの指定には、peeras として相手の AS 番号を指定する。当然、igp, internal の場合は、通常 local AS 番号が指定される。

BGP の運用の際、IBGP と IGP の同期に関する問題がある [202]。gated では、group igp の場合には AS Path が短い場合には完全に OSPF tag で AS Path が転送され、AS Path が長い場合でも IBGP での経路は直接アナウンスされず、OSPF 経路とマッチングをしてからアナウンスされるため、同期の問題は解決されているということができる。

一方、group internal は、本来 IGP なしで到達できるゲートウェイ間で使用されることが当初の制約であり、gateway の指定は後から追加されたものである。この gateway を指定しなければならない場合、IGP との同期問題は解決されていないと考えられる。従って、transit AS として機能する場合には、このような使用法は極力避けるべきである。

4.9.4 Redirect

gated は、経路情報交換に参加している場合、ICMP redirect を無視するように動作する。これは、受信した ICMP redirect による "D" フラグ付きの経路をすぐさま経路表から取り除くことで実現している。通常はこの動作でよいわけで、特に redirect 節を指定する必要はない。

4.9.5 SNMP

gated は、isode_snmp を指定して make した場合には、SNMP による内部変数へのアクセスを許しており、特に指定する必要はない。

[6] 少なくともバージョン 9.1.5 はそうである

4.10　Static 経路

各プロトコルの構成を指定した後で、必要な場合には static 経路を設定する。経路は、natural mask であれば 202.13.183.0 のように単に IP アドレスを指定するだけでよいが、ホストを指定する場合には、目的アドレス部を

```
host 202.13.183.1
```

あるいは

```
202.13.183.1 mask 255.255.255.255
```

のように指定する。また、subnet 経路の場合は必ず mask が必要である。

Preference を経路制御プロトコルの preference より大きな値にしておけば、その経路制御プロトコルで同じ経路が得られた場合、それが優先される。また、組織の全ての subnet 経路を preference 255 で reject インターフェースに向けておくことによって、ダウンしている subnet へのパケットが default route によって Backbone に送られるのを防ぐことができる。

ここで指定された static 経路は gated が shutdown した際には削除されてしまうが、gated 終了後も残すには retain を指定する。

gated では、point-to-point ネットワークはトンネルとして取り扱う。そのため、point-to-point ネットワークに subnet アドレスを割り振った場合、両端のアドレスに対する host 経路は direct として生成されるが、その subnet の経路情報は生成されない。Subnet 経路を生成するためには、次のように static 経路を定義し、RIP などでアナウンスする際に指定する：

```
static {
    133.4.2.0 mask 255.255.255.224 gateway 133.4.2.2;
};
```

Static 経路を RIP でアナウンスする場合、RIP の default metric は 16 になっていることに注意せよ。従って、rip プロトコルに対して defaultmetric を指定するか、export するときに metric を上書きする必要がある。そうしないと有効な経路はアナウンスされない。

4.11　経路の受理

OSPF 以外のプロトコルでは、経路情報を条件によって採用したり無視したりすることができる。これは import によって指定する[7]。

import はプロトコル別に指定する。また、プロトコルによっては、隣接 AS 番号や AS Path、インターフェース、隣接ゲートウェイ、tag などを指定できる場合がある。また、

[7]Version 2 では accept によって指定していた。文法も変更になっているので注意が必要である。

gated では、それぞれのプロトコルの import の指定はネットワーク単位で制御することができる。

プロトコルが ON になっており、import が特に指定されていない場合は、そのプロトコルでは全ての経路を受信するように指示したことと等価になる。従って、routed に相当する gated.conf は、null ファイルでよいことになる。

```
import proto rip;
```

は全ての RIP 経路を受信することになるし、restrict を用いて

```
import proto rip restrict;
```

は全ての RIP 経路を無視する設定になる。また、

```
import proto rip gateway 133.4.1.1;
```

は、133.4.1.1 というアドレスを持つゲートウェイからの RIP は全て採用するということを指定する。

これらの import を ; で終了させないで *import_list* を指定することによって、ネットワーク単位の制御が可能である。*import_list* は次のいずれかの形式の要素をセミコロン ; で終端したものの列である：

- 「すべて」を意味する all
- ネットワーク番号。133.4 あるいは 133.4.0.0 のように指定する。この場合には、natural mask が伴っているものと判断される。
- キーワード host を用いて、対象をホストに限定する：

  ```
  host 133.4.1.1
  ```

- ネットワーク番号とマスク：

  ```
  133.4.0.0 mask 255.255.0.0
  ```

  この場合、マスクは、当該ネットワークマスクを持つ経路という意味ではなく、経路の destination とここで指定されたマスク 255.255.0.0 との bit 毎の論理積が 133.4.0.0 に等しい経路全てが表現される、と考える。従って、

  ```
  133.0.0.0 mask 255.0.255.255
  ```

  は、133.xx という Class B の経路全部、ただし subnet 経路や host 経路は除く、という指定になる。

- ネットワーク番号とマスク幅：

```
133.4.0.0 mask-length 16
```

は先頭 8bit を有効とするもので、BGP4 用語でいうところの prefix に相当するが、意味は、

```
133.4.0.0 mask 255.255.0.0
```

と同じである。

- これらの 5 つの後にキーワード `restrict` を付したもの。`restrict` によって、条件にマッチした経路は受理されない、ということを指定する。

例えば、

```
import proto rip gateway 133.4.1.1 {
     133.27.0.0 mask 255.255.255.255;
};
```

は、ゲートウェイ `133.4.1.1` からは経路 `133.27.0.0` のみを受信しその他の経路は無視することを意味する。Import リストの最後には、全てを無視する意味の

```
all restrict;
```

があることが仮定されており、

```
import proto rip gateway 133.4.1.1 {
};
```

は、ゲートウェイ `133.4.1.1` からの RIP は全て無視するという記述になることに注意が必要である。

4.12　経路のアナウンス

経路制御を行なう際、もっとも気をつけて設定しなければならない部分が、学習した経路をどのようにアナウンスするか、ということである。特に、internal 専用のネットワークがある場合や、複数のネットワークに接続し、かつその間の transit なトラフィックの取り扱いを行なわない場合、慎重な設定が必要である。また、インターネットの発展につれ、効率的な経路制御も要望されるようになってきており、不必要な host 経路のアナウンスなどは行なわないことが要請されている。これらの経路のアナウンスの操作のほとんどが `export` の指定で制御することができる。

`export` はアナウンスするプロトコル毎に指定を行なう。ドキュメントには明記されていないが、`export` を全く指定しなかった場合、そのプロトコルで普通に学習した経路はそのままアナウンスされるようである。また、ルータのインターフェースのアドレスもアナウンスされる。従って、RIP のみに参加しているゲートウェイで、特に制御が必要

でない場合、export を全く指定しなくても差し支えない。しかし、不用意にあるいは実験的な経路が外部に広告されるのを防ぐため、組織の WIDE Internet 向き gateway では、登録された network 経路のみをアナウンスするように指定することが望ましい。

export であるプロトコルでアナウンスする経路を制御する場合、export_list として経路が得られたプロトコルに着目して制御を記述する。つまり、

```
export proto DST_PROTO {
    proto SRC_PROTO_1 ;
    proto SRC_PROTO_2 {
        announce_list
    };
};
```

という記述は、

1. *SRC_PROTO_1* で得られた経路全てと
2. *SRC_PROTO_2* で得られた経路のうち、*announce_list* にマッチするもの

が *DST_PROTO* によってアナウンスされる。

ここで、「得られた経路」とは、単に経路が得られたということだけではなく、その経路が gated によって採用されているということが必要である。例えば、133.4.0.0 に対する経路が RIP (pref = 100) と OSPF (pref = 10) の両方で得られた場合、OSPF 経路が採用される。従って、RIP で得られた経路のみをアナウンスする設定では、133.4.0.0 はアナウンスされないことになる。

実際にある経路がどのプロトコルから得られているかということを判断するのはある程度の経験が必要であるが、

- Ethernet などの non point-to-point ネットワークのネットワークアドレスは direct として取り扱われる。
- Subnet 化されているネットワークの境界（異なったネットワークに対するインターフェースを持っている場合)、subnet ネットワークのネットワーク経路は direct として取り扱われる。
- Point-to-point リンクの remote 側のアドレスも同様に direct で学習されたとして処理される。
- Point-to-point リンクに subnet アドレスを振った場合、これに対する subnet 経路は static に宣言する必要があり、static からの経路として取り扱われる。

ということを念頭においていれば大きな間違いはない。また、運用中の gated プロセスに INT シグナルを送ったときに得られる gated_dump ファイルをみると、各経路がどの

プロトコルで得られているか、どのプロトコルで得られた経路が採用され、どのプロトコルでアナウンスされているかを知ることができる。これについては、4.13 に述べる。

export_list には、ソースプロトコル名と必要な制約を記し、さらに必要な場合には announce_list として経路一つ一つの指定も可能である。また、restrict を付加することによって、そのプロトコルからの経路をアナウンスしないように指定することもできる。

指定できる制約はソースプロトコルによって異なっているが、RIP の場合、

- インターフェースを指定（経路がそのインターフェースの方向にあるか）
- ゲートウェイを指定（経路がそのゲートウェイを next hop とするかどうか）

の 2 種類が可能である。

ソースプロトコルの default は、BGP や EGP を利用するゲートウェイで default 経路の生成を行なった場合の default 経路を意味し、通常の RIP で学習した default 経路はソースプロトコルは RIP として取り扱われる。

OSPF ASE 経路に対しては、それぞれに付加されている tag を指定してアナウンスをすることができる他、BGP の場合は AS Path の指定も可能である。

export の指定における *announce_list* は、import の *import_list* と同じ指定が可能であり、マスクによって経路の範囲を指示したり、restrict によってアナウンス不可を表明することができる。*announce_list* の最後には、*import_list* と同様に

```
all restrict;
```

が仮定されており、指定にマッチしない経路のアナウンスは *import_list* と異なる点は、アナウンスするプロトコルとは異なったプロトコルから得られた経路をアナウンスする場合、ネットワークの範囲毎に metric を指定できる点である。

Metric を指定しない場合には、*export_list* に指示した metric が使用され、ここでも指定されていない場合には、アナウンスするプロトコルの default metric が使用される。従って、static 経路などをアナウンスする際には、metric を指定しないと、RIP では metric 16 が仮定され、実質的にアナウンスは行なわれない点に留意されたい。

Point-to-point リンクは gated では tunnel として取り扱われ、remote 側のアドレスが direct で得られたものとしてアナウンスされる。この host 経路は、多くの場合冗長であり、インターネット全域にアナウンスされてしまう可能性があるので、注意を要する。例えば、次のような export の指定は、このような host 経路のアナウンスを防ぐことができる[8]：

```
export proto rip {
    proto rip;
    proto direct {
```

[8] Remote 側アドレスが Class B アドレスであり、0.0.0.255 との論理積が 0 の場合には、各々のネットワークアドレスをアナウンスする記述が必要になる。

```
            0.0.0.0 mask 0.0.0.255;
        };
    };
```

4.13 gated_dump について

```
 133.123         mask 255.255
                 entries 1       announce 1
                 TSI:
                         KERNEL gateway 133.123.1.1
                         OSPF LSDB seq 800003cd
                         RIP 133.4.3        2
                         RIP 133.156.1.1    2

           RIP   Preference: 100
                 NextHop: 133.123.1.1            Interface: 133.123.1.1(ptp4)
                 State: <Int Active Gateway>
                 Age: 00:00:00   Max Age: 00:03:00       Metric: 2       Tag: 0
                 Announcement bits(4): 1-KRT 2-OSPF 3-RIP.0.0.0.0 4-RIP.0.0.0.0
                 AS Path: IGP (Id 1)
```

図 4.2: gated 中の経路表の例

gated_dump にはさまざまな gated の内部状態が出力されるが、経路制御のデバッグに一番役立つのは、gated 内部の経路表とその経路の学習・アナウンスの状況である。図 4.2 は gated_dump の出力のうち、一ネットワーク分の経路表である[9]。

これを見るとさまざまなことがわかる。まず、経路 133.123 は 133.123.1.1 からの RIP で学習されており、インターフェースは ptp4、metric が 2 であることがわかる。この経路の状態は、Active つまり、活動状態を示している。もし経路が holddown 状態ならば、State に HoldDown が表示される。

Age: は 30 秒毎に更新されるので、それほど正確ではないが、経路がきちんと更新されているかどうかを調べるときには重要である。図 4.2 の Age: は 00:00:00 となっており、経路情報が 30 秒毎にきちんと到着してることを確認することができる。一般に RIP の場合、ほとんど常に Age: は 00:00:00 になる。

この RIP で得られた経路は、TSI: という部分によってどのように利用されているのかがわかる。KERNEL は、この経路が実際に kernel の経路表に書き込まれたことを示しており、OSPF LSDB は OSPF の link state database に出力されていることを示している。さらに、RIP によって、133.4.3 と 133.156.1.1 にそれぞれ metric 2 でアナウンスされていることがわかる。

経路が kernel の経路表に使われるだけで他にアナウンスされていない場合には、TSI: は KERNEL だけの表示になる。また subnet の境界のゲートウェイでは、network 経路は

[9] これは説明のためのもので、実際のものとは異なる

プロトコル Direct によって得られるが、この場合は、kernel の経路表には使用されないので、TSI: には KERNEL の表示は含まれないことになる。

OSPF ASE 経路の場合は 4.3 のようになる。この経路は、OSPF ASE と BGP の両方で得られているが、Preference: の数字を比較して小さい方（最初の経路）が採用されていることを示している。entries 2 は二つのプロトコルで経路が得られていることを表す。

この OSPF ASE 経路は、3489663425 という tag を持っている。この tag は、16 進表示では 0xd00009c1 であり、RFC 1403 に基づいた tag 生成された結果の数字である。0x9c1 = 2497 であるから、AS 2497 から得られた経路であるということがわかる。それを解釈したのが Path: で、(2500) 2497 IGP は隣接 AS 2497 から origin が IGP である経路であることが示されている。

```
192.244.176     mask 255.255.255
                entries 2       announce 1
                TSI:
                        KERNEL gateway 133.4.11.1
                        BGP 133.4.11.29 (External AS 2528) metric 1000

        OSPF_ASE        Preference: 150
                NextHop: 133.4.11.1             Interface: 133.4.11.14(le0)
                State: <NoAge Int Active Gateway>
                Local AS:  2500
                Age: 04:40:00   Metric: 9281    Tag: 3489663425
                Task: OSPF
                Announcement bits(2): 1-KRT 4-BGP_2528.133.4.11.29
                AS Path: (2500)  2497 IGP (Id 4)
                Cost: 9281      Area: 0.0.0.0   Type: ASE       AdvRouter: 192.244.177.2
                Type: 1 Tag: 0 Path: (2500) 2497 IGP

        BGP     Preference: 170         Source: 133.4.3.16
                NextHop: 133.4.11.1             Interface: 133.4.11.14(le0)
                State: <NoAge Int Gateway>
                Local AS:  2500 Peer AS:  2500
                Age: 00:16:00   Metric: -1      Tag: 0
                Task: BGP_2500.133.4.3.16
                AS Path: (2500)  2497 IGP (Id 4)
```

図 4.3: OSPF ASE 経路の例

また、gated_dump の経路表の後には、各経路制御プロトコル毎の種々の情報が表示されている。たとえば OSPF では、Ethernet などのマルチキャスト型のネットワークでは、designated router および backup designated router を動的に決定するが、これらの情報は、図 4.4 のようになる。これによると、このルータ自身が designated router となっており、133.4.11.1, 133.4.11.8 という二つのルータがこの OSPF に参加していることがわかる。

```
Task OSPF.133.4.11.14:
        Interface: le0
        Area: 0.0.0.0           Cost: 10
        State: DR               Type: Broadcast
        Priority: 20
        Designated Router: 133.4.11.14
        Backup Designated Router: 133.4.11.1
        Authentication: 48.69.6d.69.74.73.75.21
        Timers:
                Hello: 00:00:10  Poll: 00:00:00  Dead: 00:00:30  Retrans: 00:00:05
        Neighbors:
                RouterID: 133.4.1.1             Address: 133.4.11.1
                State: Full     Mode: Master    Priority: 5
                DR: 133.4.11.14 BDR: 133.4.11.1
                Last Hello: 18:15:32    Last Exchange: 09:00:00

                RouterID: 133.4.11.8            Address: 133.4.11.8
                State: Full     Mode: Master    Priority: 22
                DR: 133.4.11.14 BDR: 133.4.11.1
                Last Hello: 18:15:30    Last Exchange: 09:00:00
```

図 4.4: OSPF に関する情報

第 5 章

付録

5.1 Sun で gcc 無しの場合の Config

SunOS4.1.x の場合、以下の Config を使用することによって、gcc 無しでも gated を作成することができる。プロトコルは必要最小限にすべきである。また、ISODE_SNMP を利用しない場合には、プロトコル記述から isode_snmp を削除し、ライブラリからも -lisnmp -lisode を削除する。

このコンフィギュレーションでは、/etc/gated.conf に gated.conf があることが仮定される。実際の gated を in.gated としてインストールした場合、PID ファイルおよび version ファイルはそれぞれ/etc/in.gated.pid, /etc/in.gated.version となる。

```
#
# Config file for a Sun 4 running SunOS 4.1.x
#

bindir   /usr/etc
signal_h /usr/include/sys/signal.h

cc       cc
cflags   -g

ldflags  -Bstatic -lkvm -lisnmp -lisode

lex      lex
lflags   -v

mkdep    mkdep -flag -MM

yacc     yacc
yflags   -d

options  INCLUDE_UNISTD HAVE_DIRENT GID_T=gid_t
options  POSIX_SIGNALS HAVE_SYS_SIGLIST HAVE_WAITPID
options  KRT_RTREAD_KMEM KRT_IFREAD_IOCTL KRT_RT_IOCTL
options  KRT_LLADDR_SUNOS4 KRT_SYMBOLS_NLIST KVM_TYPE_SUNOS4
```

```
options  KSYM_IPFORWARDING=""_ip_forwarding""
options  KSYM_UDPCKSUM=""_udp_cksum""

path_config     /etc/gated.conf
path_dump       /var/tmp/gated_dump
path_dumpdir    /var/tmp

protocols       icmp rip bgp ospf isode_snmp
```

SunOS 4.0.3 の場合は、上記の SunOS 4.1.x 用の Config の options を次のように変更する。ただしこの場合、krt_lladdr_kmem.c にパッチを当てる必要がある。実は、SunOS 4.1.x の場合と同様に KRT_LLADDR_SUNOS4 を選択すればパッチは必要ないが、WIDE 版 ISDN driver などを実装している場合、NIT インターフェースを持たないためのメッセージが定期的に表示される。KRT_LLADDR_KMEM を使用することによって、これを避けることができる。

```
options  HAVE_DIRENT GID_T=int PID_T=int
options  BSD_SIGNALS HAVE_SYS_SIGLIST
options  KRT_RTREAD_KMEM KRT_IFREAD_IOCTL KRT_RT_IOCTL
options  KRT_LLADDR_KMEM KRT_SYMBOLS_NLIST KVM_TYPE_SUNOS4
options  KSYM_IPFORWARDING=""_ipforwarding""
options  KSYM_UDPCKSUM=""_udpcksum""

*** krt_lladdr_kmem.c.ORG       Tue May 18 04:58:02 1993
--- krt_lladdr_kmem.c   Fri May 21 15:48:33 1993
***************
*** 107,113 ****
--- 107,117 ----
                /* This is the one we want */

                return sockbuild_ll(LL_8022,
+ #ifdef         sun
+                                    (byte *) &arpcom.ac_enaddr,
+ #else
                                     (byte *) arpcom.ac_enaddr,
+ #endif         /* sun */
  #ifdef         KRT_RT_SOCK
                                     (size_t) ifp->if_addrlen
  #else /* KRT_RT_SOCK */
```

5.2　運用

WIDE Internet における経路制御は、backbone の部分ではほぼ OSPF への移行が終了しており、一部を除いて OSPF で経路の交換が行なわれている。また、BGP への移行も一部で始まっているが、何分国内では経験がなかっただけにいろいろな cut and try が必要になる。

1993 年 5 月当初の WIDE Backbone の経路制御の状況は、同月に開催された WIDE 研究会に加藤によって報告されているが、このときにメモを参考のため添付する。おおまかな方針に関してはあまり変更はないが、細部の変更はいろいろと生じている。

例えば、他のネットワークプロジェクトから RIP で得た経路を OSPF ASE に変換する際、どのネットワークからの情報であるかということを tag に記しておき、OSPF ASE から再び別のネットワークプロジェクトに RIP でアナウンスするかどうか、する場合の metric はどうするかを決定していた。しかし、tag を自由に設定できないルータがあるため、現在、tag はそのままネットワークを表現する値が設定されているが、RIP にアナウンスする際には tag を参照しないで、経路を一々指定する方式にしている。

そのため、ある組織が WIDE Backbone と RIP で経路情報を交換していた場合に、これを OSPF に変更するのは、その組織へのリンクを収容する router だけの調節ですむようになっている。従って、OSPF での経路制御を希望する場合には、経路制御担当と調整の上、移行することは可能である。

1993 年 6 月現在、この移行の実験はまだスタートしていないので、どのような方針で設定を行なうかは未定である。しかし、おそらく、WIDE Backbone 側のルータを backbone エリアとその組織のエリア両方の経路制御に参加することになると考えられる。また、特に必要でない場合には、ASE 経路の flooding を行なわない “stub” エリアとして設定することになると考えられている。

また、Proteon では、OSPF ASE type 1 経路の cost と RIP metric は直接比較されるので、$cost >= 15$ の経路は RIP ではアナウンスされないことになる。従って、WIDE Backbone 側からは、`133.4.0.0` の経路と default 経路のみを RIP で送るようにし、組織側では default 経路を RIP で override 可能な状態で static に定義する、という設定が必要になる。この場合、送出される RIP 経路は数個であるため、300 を越える経路情報を 30 秒毎に送っているオーバヘッドは大幅に改善される。

今後の経路制御の種々の変更は、WIDE Internet 全体の管理者の集合のメーリングリスト

```
inet@wide.ad.jp
```

にアナウンスされる。

経路制御プロトコルの移行について

加藤　朗
kato@wide.ad.jp

Abstract

国内のネットワークは、その発達につれてトポロジが複雑になり、また経路に関するポリシの実現や自動的な経路のバックアップといった要求も重要になってきている。この状況を単一の RIP で運用するのはもはや限界であり、JEPG/IP は、ネットワーク間の経路情報の交換は BGP-3 に移行することを要請している。

WIDE Internet では、これに基づいて移行の計画を行ってきたが、その第一段としてバックボーンにおける経路制御プロトコルの変更を行っている。本稿では、この移行の計画および状況、各参加組織に対する影響などについて報告する。

1. はじめに

JEPG/IP では、国内のネットワークプロジェクト相互間の経路情報交換には、従来から用いられてきた RIP[123] に替えて BGP-3[126] を用いるように要請している。この主な理由は次のようなものである：

- RIP では metric=16 が ∞ であり、日本のインターネットの直径をすでに越えてしまっている。このため、経路情報が到達しない部分が存在している。
- 経由するネットワークの選択を行いたいという要求があり、これによって RIP metric を人工的に加工する必要さえ生じている。
- ネットワーク毎に異なった経路制御を行うことができない。
- 接続されている IP ネットワーク数が増大するにつれ、経路情報の伝達のオーバヘッドが増加している。また、CIDR[50] 方式の基づく IP ネットワーク番号割当を受けた組織の接続により、このオーバヘッドの増大は著しいものが予想されている。
- CIDR への対応に備えて、CIDR 対応のプロトコルに移行することが要請される時期がまもなく訪れると予想されている。
- BGP では AS PATH で経路選択を行なうので、RIP のみでは実装困難な自動的な backup route の選択も可能になる。

国内のインターネットの相互接続の現状は本稿末尾に示す通りであるが[1]、北海道から九州までをカバーし、しかも国策に依存していない WIDE Internet は、研究ネットワーク相互の接続を行う、いわゆる transit としての期待が高い[2]。

ところで、BGP を transit AS で運用する場合、AS 内の IGP による経路の伝搬と、 BGP speaker 同士の BGP による経路情報の伝搬（IBGP と略記する）には、一般には時間差が生じ

[1] 最新版は nic.ad.jp:inet/connection.ps[.mono] に置くことにしている。
[2] 正確な意味は不明であるが、「学者の運用するネットワーク」という評価を受けることもある。

る。このため、IBGP で得た経路を外部の AS にアナウンスした場合、IGP による経路が確立していないことがある。そうした場合、内部の router で unreachable になったり、古い IGP による経路のため、正しい border gateway にパケットが運ばれないという問題が生じることがある。これを防ぐため、運用上次のいずれかを実施することが要請されている [202]：

- 経路に tag を付けることができる IGP を使用し、IBGP における経路とのマッチングを可能にする。
- Border gateway 間で transit するパケットを encapslation し、内部の router の経路表の確立には依存しないようにする。
- 上のいずれにもよらない場合、IBGP で経路が得られた場合、静定時間を経過してから、経路を外部にアナウンスする。

上の手法のうち、最も好ましいのは一番最初の tag 付き IGP を使う方法である。この tag 付き IGP には、OSPF[27] や RIP-2[200][203]、dual-ISIS などが知られており、現在のネットワークの状態ソフトウェアの実装状況などを考慮すると、OSPF が好ましいと考えられている。事実 BGP-3 と OSPF の相互の情報交換には、ガイドラインが定められており [201]、IAB や IESG も OSPF を標準として位置付けることを考えている [204][205]。

1993 年 4 月 30 日に開催された JEPG/IP での合意は次のようなものであった：

- 国内のネットワーク相互間の経路情報の交換は早期に BGP-3 に移行する。
- 当面、国際的な経路情報の交換は現状通りとする。

そして、JPNIC 経由で割り当てをうけた国内使用分 32 個の AS 番号を表のように割り振った。

AS 番号の割当て

AS 名（暫定）	AS 番号
IIJ-AS	2497
JOIN-AS	2498
SINET-AS	2499
WIDE-AS	2500
TISN-AS	2501
TRAIN-AS	2502
unassigned	2503-2528

2. WIDE Internet

WIDE Internet では、現在 Sun + `gated`, Proteon P4200, Cisco IGS/AGS+ がバックボーンの接続に使用されている。`gated` は昨年以来 Release 3 が α バージョンとしてリリースされてきており、数多くの bug fix や機能の追加がなされてきた。

`gated` の relase 2 から release 3 への変更は、次のようなものがある：

- `gated.conf` の書式が変更になった。従って、`gated.config` の書換えが必要であるが、それほど大がかりな作業ではない。

- OSPF や ISIS のサポートが追加された。
- サポートされている BGP のバージョンが 2 から 3 になった。
- RIP-2 もサポートされるようになった。
- Dynamic interface のサポート。この機能により、インターフェースを定義しておけば、`gated` を起動する時にそのインターフェースが active になっている必要がなくなった。
- SMUX SNMP のサポート。これによって、`ipRouteMetric1` などを正しく SNMP で知ることができるようになった。

α リリースの段階では、まだ動作が不安定であったが、1993 年 4 月末にβ リリースに移行した。やや不安定な部分が残っているものの、WIDE Backbone 上での RIP & OSPF の使用経験から、そろそろ実用に耐える範囲に近づいていると判断することができる。

WIDE Backbone では、OSPF への移行に関して、次のような順序で実施する方針にした：

1. RIP とは独立に OSPF を運用する。この場合、OSPF は kernel 中の経路表の更新は行なわないようにして、単に経路情報の交換だけを行なうようにする。OSPF がサポートされていた Proteon を交えて動作を確認していった。
2. RIP と OSPF を同一の `gated` プロセスで運用する。この場合、経路情報は、OSPF, RIP, OSPF-ASE の順で採用されるので、OSPF に関係していない経路は RIP で搬送される。
3. RIP による経路の搬送を停止することによって、OSPF-ASE を含む OSPF での経路制御が実現される。

もちろん `gated` だけではなく、cisco や Proteon も対象する必要がある。これらのルータは、OSPF のみで運用する場合には大きな障害はないと思われるが、現実には RIP との経路の変換は不可欠であり、このあたりの実装の方針のずれで必ずしも満足いくような状況ではない。

WIDE Internet の場合、実際にネットワークを動かしながら、down を最小限にして移行を行なわなくてはならないし、また外部との経路情報交換は相変わらず RIP である。当然 OSPF では RIP metric は保存されないので、今まで RIP metric の人工的な増加で表現されてきた routing policy が実現できなくなる。そのため、次のような対応をとった：

- 各ネットワークと調整し、現在の routing policy を逸脱せず、かつ routing loop が発生しない範囲で、WIDE から送出する RIP metric の変更を承認して頂いた。
- 各ネットワークからの経路は、入口のゲートウェイで OSPF-ASE に変換する際、経路がどのネットワークのものであるかを示す tag を付ける。
- 経路を OSPF-ASE から RIP に変換する際には、tag を参考にして必要な metric を設定するようにした。

特に、TISN/GENOME とは東京及び京都の両方の地点で相互接続されているため、特に慎重な調整が必要であった。京都以西と TISN/GENOME の通信は京都経由で行なわれ、それ以外は東京経由になることが、両者のリンクの効率的な利用を考えると好ましい。従って、OSPF 経路を RIP でアナウンスを行なう際にも、それが京都以西かどうかを判別する必要がある。

現在、経路情報のほとんどは RIP から得られており、従ってネットワーク毎に tag を定義し、OSPF-ASE に公開する際にはその tag を付けることにした。そして、東京側では、東京大学の交換セグメント上に送出する metric を 8 とし、京都側では、京都以西を連想させる tag がつい

た経路については metric 3、それ以外は metric 6 にすることによって、この条件は一応満足されている。東京側での metric を一定にしたのは、cisco の問題と、TISN/GENOME 内部の事情で 8 より小さくした場合に問題が生じることによる。

TISN/GENOME 以外は WIDE とは一点で接続されているので特に問題は発生しないと思われる。そこで、現状では、metric 3 を原則として RIP で経路の送出を行なっている。ただし、一部 SINET との利用の区別を行なうため、異なった metric でアナウンスしている場合がある。

gated を利用する上で、Ethernet のような subnet mask が有効な媒体を接続している場合、そのインターフェース上でも OSPF を動かし、必要な場合には Summary を送出するように設定する必要がある。現在、次の経路が OSPF （OSPF ASE ではない）で搬送されている：

```
133.5, 133.27, 133.41
```

3. 問題点

この OSPF への移行を考える場合の、ルータの実装に関係する問題点として次のようなものがある：

Proteon P4200 と P4100+ では `V12.1Z [8`[3] を使用しているが、次のような点が問題になる：

- RIP 経路を OSPF に変換するとき、経路の指定ができない。
- 任意の tag を付けることができない。
- OSPF 経路を RIP に変換する時、metric を設定できない。また、経路の指定もできない。

これらは、外部との経路情報の交換が BGP に移行し、また backbone 部分が完璧に OSPF に移行した場合、multi-home でない組織を収容するルータとしてはそれほど不自由ではないが、現在のような過渡期には問題である。

Cisco 9.1.3 では、次のような点が問題である：

- OSPF 経路を RIP にアナウンスする時、internal / external-type 1 / external-type 2 という単位でしか、metric を設定できない。
- 大きな問題ではないが、routerid を自由に設定できない。
- 古いルータの場合、ハードウェアの交換が必要になる場合がある。

これらも移行期のみに問題となるものである。

gated R3_0Beta_1 で OSPF を運用する場合の問題点は、

- Multicast のサポートが必要である。Sun などの場合、ドライバを入手することができるが、ハードウェアによっては OS のバージョンが限られている[4]。

[3] バグを指摘し、修正を行なったバージョンで、ベンダから入手できるかどうかは定かではない。

[4] Sun3 の場合は SunOS4.0, Sun4 の場合には SunOS4.1.1, Sun4c の場合には SunOS4.1.x に対するドライバが入手可能である。これらは、`sh.wide.ad.jp:vmtp-ip/` にある。また `gated` は、`sh.wide.ad.jp:routing/` から入手可能である。

いずれにせよ、どの製品を使用する場合でもまだ若干の不安定さは残存しており、常に最新のバージョンをインストールし、デバッグし、バグレポートを送ることができる種々の根性が必要である。

gated の現在のバージョンでは、OSPF を使用している場合、config の変更に関しては再起動が必要になる。そのため、経路が確立するまでの時間（OSPF ルータ間では 30sec 程度）は unreachable になり、TCP コネクションに影響を与える可能性がある。また、RIP へ変換したものを受信している場合、最大２分程度が必要な場合もある。なるべくネットワークの利用に影響を与えないように留意したいが、やむを得ない場合もあるので、利用者各位のご協力とご理解を仰ぎたい。

一方、経路が unavailable になった場合、RIP では当該経路がアナウンスされなくなるまでに 180sec の holddown time の経過が一般に必要である。この間はループが発生したり、metric の大きな代替経路が存在していても切替えが行なわれないなどの問題点がある。OSPF の場合には、情報は速やかに伝搬するので数秒でルータの経路表が更新されるという特徴がある。

現在の WIDE Backbone の東京－京都間のように、常時は藤沢経由であるが、藤沢に障害が発生した場合、奈良経由を選択できる場合、OSPF の特徴が有効であると予想されている。OSPF が発生するトラフィックや設定された経路の動特性の解析は今後の研究が必要である。

4. WIDE 参加組織

現在の OSPF の運用は backbone の一部に限られている。Full OSPF の運用を行なっている Backbone を構成しているルータは６台[5]で、近日中に札幌のルータが加わる予定である。

WIDE Backbone と各参加組織との間の経路制御は当面 RIP のままとしたい。それは次のような理由による：

- 外部への経路の送出は tag の情報を利用している。OSPF (type 1, 2, 3) の経路は tag が付されていないため、RIP へ変換する際の metric を個別に記述しなければならない[6]。
- ソフトウェアが十分に安定していない。また、頻繁にパッチやバージョンアップがアナウンスされている。従って、gated のメーリングリスト[7] を購読している必要がある。
- 運用方法が十分に確立しているとはいえない。

しかしながら、近年の接続ネットワーク数の増加により、RIP による経路の搬送のオーバヘッドは 64kbps のリンクでは無視できないものになりつつある。そこで、WIDE Backbone と各組織の間の経路の搬送については、次のいずれかを選択できるようにしたい：

- RIP によって現状通り全ての経路情報を送出する。これは、ネットワークの管理に必要である、あるいはマルチホームであるなど特別な理由がない限りお勧めしない。
- RIP によって default 経路を送る。この default は従来の「海外」を意味するものではなく、「組織外」を意味するものとなる。従って、海外リンクがダウンしている場合でも、WIDE Backbone 側のルータで default を生成し、送出する。

ルータの種別にも依存するが、WIDE 関係の経路のみ、という設定も不可能ではない。実際の移行については、準備ができ次第、各組織の管理者と相談の上実施したい。

---

[5] 17:00 JST 13 May 93 現在

[6] 共同研究遂行上の理由で、日本 IBM（192.156.220）は OSPF を利用しており、そのための設定を各ルータで行なっている。

[7] gated-people@gated.cornell.edu 参加申し込みは-request にメールを送ること。

なお、各参加組織からの経路情報は当面現状通り RIP とする。WIDE と他のネットワークとの経路情報の交換が BGP に移行した時点で、希望があれば WIDE Backbone との経路情報交換を OSPF に移行できるように検討・準備したい。

5. BGP

ネットワーク間の経路情報交換の BGP への移行に関しては、1993 年 5 月下旬より JOIN および IIJ とテストを開始する予定である。これは cisco を利用したものであるが、今後各地域ネットワークの準備の状況によって `gated` でのテストも行ない、6 月中旬には移行を行ないたいと考えている。

国内の IP ネットワークの数は現在 270 程度であり、経路制御が OSPF に移行した場合、1000 程度は問題なく処理できると予想されている。従って、CIDR に基づいた経路の集成（route aggregation）の導入は緊急課題ではないが、いずれ国際部分を BGP に移行する時には、BGP-4 を要求される可能性もあるので、準備は進めておきたいと考えている。なお、NSFNET では 1993 年夏から経路の集成を開始する予定である。

BGP は本質的に Connection Oriented なプロトコルであり、`gated` の config の設定に関する再起動では、経路が安定するまでに必要な時間がさらに長くなることも考えられる。早朝あるいは夜間に行なうことが望ましい[8] が 24 時間体制のオペレータがいない状況では簡単ではない。変更を行なう時間帯や曜日を設定し、緊急度の高くないものに関しては、そのタイミングで変更を行なうようにしたい。

謝辞

JEPG/IP およびその routing WG の諸氏、OSPF への移行に対して作業あるいは作業のバックアップを戴いた関係諸氏、また短い時間ではあったが、若干生じた移行期のネットワークの接続性の低下を容認して戴いたユーザ諸氏に感謝する。

[8] NSFNET の経路データベースの更新は週 2 回、いずれも現地時間の未明に行なわれる。