A Brief Guide to Gated IS-IS Configuration

Revision:
12/15/92
Steve Heimlich
heimlich@watson.ibm.com

The operation of IS-IS within gated is controlled by the gated configuration file. This file contains any information needed to describe to gated the environment it is managing. The configuration file may also specify various tunable parameters for the IS-IS configuration. Once running, gated can be reconfigured by sending it a signal which will cause it to reread its configuration file.

In general, the style of the configuration file options specific to IS-IS follows that defined for the other protocols supported by gated. We begin with an annotated example of a gated configuration file which enables the ISIS protocol and specifies a simple environment. We continue with a brief discussion of the specific commands included in that configuration along with additional examples of each command. We conclude with a brief summary of the gated configuration syntax in an informal BNF notation.

Gated configuration example

The following example shows the contents of a gated configuration file that will enable gated to run the ISIS protocol on a machine with an ethernet interface. This configuration example shows how to specify circuits, IP addressing, and other information to be handled by ISIS. Each of the commands used in the sample configuration are discussed in detail in later sections. Note: commands discussed in the following sections but not included in this example may be obsolete; this document is not yet final.

```
isis dual                                   # integrated (ISO and IP)
{
   level 2;                                 # run in level 2 and level 1
   traceoptions all;                        # turn on all IS-IS trace options
   systemid <02608c2fe25f>;                 # set the system ID
   area <47000580deadbeef0000000001>;       # set the area address
   circuit <se0>                            # circuit device name
      metric 10 priority 10                 # level 1 metric and priority
      metric level 2 20 priority level 2 20;  # level 2 metric and priority
   set sysISHInterval 10;                   # non-default value for internal variable
};
```

Enabling IS-IS in gated

The command to enable IS-IS is similar to the existing commands for the OSPF and BGP protocols.

isis_stmt:

isis *isis_option isis_block*

isis_option:

[**yes** | **no** | **dual** | **iso** | **ip**]

The options have the following meaning:

yes -- run IS-IS in integrated mode (ISO and IP)
no -- do not run IS-IS
dual -- run IS-IS in integrated mode (ISO and IP); same as **yes**
iso -- run IS-IS for ISO only
ip -- run IS-IS for IP only (still requires underlying ES-IS support from the kernel)

In our initial example, we specified that the IS-IS protocol was to be enabled for both ISO and IP addressing with the "dual" option.

ISIS Block

The isis block consists of an initial specifier which determines the type of the system and a list of statements that determined the specific configuration of the circuits and networks which the IS-IS protocol will manage. Each statement is terminated with a semicolon, and the entire list is enclosed in curly brackets ({}). The statements may appear in any order. The allowable statements are described below. In general, each statement consists of a keyword indicating the type of statement, followed by any statement-specific information, and terminated by a semicolon.

Circuit Statement

The circuit statement is used to specify the circuits the system will manage. Normally these circuits correspond to unix network devices as listed by netstat -i, but simulated device names may also be specified. The circuit specifier has the syntax

circuit *string circuit_options* **;**

The string field of the circuit statement normally specifies the name of the interface which is used for input and output on the circuit. The string is delimited by angle brackets (<>). If the string is of the form "simN", where N is an integer, the circuit is assumed to be a simulated circuit managed by the network simulator troll.

The circuit attributes portion consists of a list of options that may appear in arbitrary order in the circuit statement. Circuit attributes may be used to specify the metrics and priorities associated with a circuit. Each of the options that may be used to describe attributes of a circuit are described below.

Metric Option

The metric circuit option allows specification of level 1 and level 2 metrics for each circuit. Currently only the default metric type is supported. The metric attribute is specified by the keyword

metric followed by an optional level specifier and an integral metric value. IS-IS metrics must be in the range 1..63.

metric_option:
    metric *level_option*
level_option:
    level [**1** | **2**]
    <empty>

The level specifier indicates whether the metric is to be used for level 1 or level 2 routing; it can be indicated by the keyword level and the integer values 1 or 2, or can be omitted to indicate level 1. For example, to configure a circuit with a level one metric of 10 and a level 2 metric of 20, we would add 2 metric options to the circuit statement. These options could be as below.

metric 10
metric level 2 20

The first statement does not specify a level and is used as a level one metric by default. If no metric is set for a circuit, the default value for the metric is 63.

Priority Option

The priority levels for a given circuit may be set in a manner similar to that used for metrics. The circuit priority option is done with the keyword priority:

priority_option:
    priority *level_option*

As with the metric option the level defaults to level 1. If no priority is specified the priority is set to a random value between 0 and 127. For example, to configure a circuit with a level one priority of 10 and a level 2 priority of 20, we would add 2 priority qualifiers to the circuit statement.

priority 10
priority level 2 20

The first statement does not specify a level and is used as a level one priority by default.

Ipreach Statement

The ipreach statement is used to specify IP networks that will be advertised as reachable by the IS-IS system. There are 4 parts to the options -- the keyword **ipreach**, the type of reachability (internal, external, or summary), the network (specified by IP address and network mask), and the metric associated with the network. All but the metric specification are mandatory. IP addresses and masks are specified in dot notation. The syntax is shown below.

ipreach *ipr_type ip_addr ip_addr metric_option* **;**

Some examples follow:
ipreach internal 128.104.1.0 255.255.255.0 **metric** 10 **;**
ipreach external 128.105.0.0 255.255.0.0 **metric** 10 **;**
ipreach summary 128.106.0.0 255.255.0.0 **metric** 10 **;**

IP reachable networks specified as internal are advertised as reachable in both level 1 and level 2 routing; external and summary networks are advertised at level 2 only. It is assumed that the actual routing to nets specified in ipreach statements are reachable from the ISIS host, and that routing for these nets is established by some means external to ISIS.

Prefix Addresses Statement

Level 2 prefix addresses are specified by the prefix statement. The prefix statement consists of the keyword prefix followed by a hexadecimal string specifying the ISO prefix address and a metric value associated with the prefix. For example, to specify routing for the ISO prefix 12.23.43.3, the prefix statement would be of the form

prefix 12.23.43.3 **;**

There may be any number of prefix addresses.

System ID Statement

The system identifier for the ISIS system is specified by a statement

systemid *string***;**

The string is a list of ASCII characters delimited by angle brackets (<>); only the first 6 characters of the string are used. If no system identifier is specified, the process id is used to create a random string. We usually follow the convention of using a right justified integer for the system ID statement, e.g.

systemid <   1> **;**

Area Address Statement

IS-IS area addresses are specified by a statement of the form
area <*hex-string*>**;**

Note: this is primarily useful for simulation; IS-IS automatically configures its area addresses based on the real circuits over which it is running. Addresses specified in this statement are maintained in addition to those configured automatically from the circuits.

Configuration Parameters

A variety of internal variables used to control and tune the operation of the IS-IS protocol can be modified through the configuration file. The general mechanism used for this is the set statement:

set <variable name> <*integer*> **;**

The following table lists the names and default values of the variables that can be changed:

Table 1: IS-IS Configuration Parameters

Variable Name	Description	Default Value
origL1LSPBufSize	Originating L1 LSP Buffer Size	1492 octets
origL2LSPBufSize	Originating L2 LSP Buffer Size	1492 octets
dataLinkBlocksize	Maximum Size of IIH or LSP	1492 octets
sysHoldingTimer	Value of IIH holding timer	20 s
sysISHInterval	Transmit frequency of ISH PDUs	5 s
sysIIHInterval	Transmit frequency of IIH PDUs	5 s
minLSPGenInterval	Minimum interval between successive LSP generations	30 s
maxLSPGenInterval	Max interval between successive LSP generations	900 s
minLSPXmitInterval	Minimum LSP Transmission Interval (p2p)	5 s
minBLSPXmitInterval	Minimum Broadcast LSP Transmission Interval	1 s
BLSPThrottle	Max number of LSPs sent per minBLSPXmitInterval	5 s
maximumAge	Initial (and maximum) lifetime of any LSP	1200 s
completeSNPInterval	Transmit frequency of CSNPs	10 s
partialSNPInterval	Transmit frequency of PSNPs	2 s
zeroAgeLifetime	Time an LSP with zero lifetime is held before purge	60 s
dumpDBinterval	Frequency with which LSP database is traced	20 s

For example, to set the ISIS Hello timer, use the statement
set sysIIHInterval 20**;**

ISIS Tracing

ISIS has its own internal tracing flags which are distinct from the flags maintained globally by gated. The mechanism used to set tracing is via the isis traceoptions statement, which has the form
traceoptions *options* **;**

The *options* specify the type of tracing desired. The following table indicates the trace options currently used and the type of tracing indicated by each:

Table 2: IS-IS Trace Options

Value	Information Traced
all	everything below
iih	IIHs sent and received
lanadj	lan adjacency updates
p2padj	point-to-point adjacency updates
lspdb	signatures in LSP database
lspcontent	contents of LSPs in database
lspinput	input processing of LSPs
flooding	flooding of all LSPs
buildlsp	generation of local LSPs
csnp	processing and construction of CSNPs
psnp	processing and construction of PSNPs
route	route changes
update	individual routes changed
paths	route paths as calculated by spf algorithm
spf	running of spf algorithm
events	interesting protocol events

Configuration Syntax Summary

This section is a concise summary of the IS-IS configuration syntax described in preceding sections. Keywords and punctuation are indicated in **bold** type, while syntactic productions are indicated in *italic*.

isis_stmt:
    isis *isis_option level_option isis_block*
isis_option:
    [**yes** | **no** | **dual** | **iso** | **ip**]

isis_block:
    { *level_option isis_stmt_list* **}**
    <empty>

isis_stmt_list:
isis_stmt
isis_stmt_list isis_stmt

isis_stmt:
circuit *string circuit_options* **;**
ipreach *ipr_type ip_addr ip_addr metric_option* **;**
prefix *string metric_opt* **;**
systemid *string* **;**
area *hex-string* **;**
snpa *integer* **;**
set *id integer* **;**
traceoptions *hex-string* **;**

circuit_options:
circuit_options circuit option
<empty>

circuit_option:
metric_option
priority_option

metric_option:
metric *level_option integer*

priority_option:
priority *level_option integer*

level_option:
level [**1** | **2**]
<empty>

ipr_type:
[**internal** | **external** | **summary**]