ISEAGE

Internet-Scale Event and Attack Generation Environment

ISEAGE

ISEFLOW configuration file format

updated August 30, 2012

Version 2.0

The ISEFlow configuration file is used to define the operation of the boards running ISEFlow. The ISEFlow configuration file needs to be compiled using mk_flow to create the binary configuration file for each board. The file format is shown below:

Overall file format consists of three sections.

1. The version number
2. A global directive
3. One or more board directives.

Overall syntax of the configuration file:

```
VER=X.Y

Globals= {
      # Global Directives
/Globals

Board=board_num
      #Board Directives
/Board
```

Each section is described below:

Version Section

The version section is used to define the version number of the config file format. The version numbering system is:

Version 1.0 is the first version using the new config file format. Version 1.0 produces config files that are identical to the config files produced by mk_map. This is currently the only version supported by the iseflow software.

Version 2.0 will support the cloud device

Version 3.0 will support different link types

The format of the VER directive is:

```
VER=Major.Minor  //version number of the conf file format
```

Global Section

The global section is used to define a set of global types at are used across each board. These include the definitions of the backplanes and definitions of globally defined links.

The format of the globals directive is:

```
Globals= {
      BPnet=Net_num,IP,Mask
            Board=board_num, router_num, Interface_num, IP
      Parms = {
            name=network_name
      /parms

      #define default link for type = IP
      DLINK=IP
      Parms = {
            name=link_name
      /parms

      GLINK=NUM,type
      Parms = {
            name=link_name
            loss=(loss in %)
      /parms
/globals
```

The global types are defined below:

BPnet is used to define the backplane networks used to connect multiple routers together.

```
BPnet=Net_num,IP,Mask
      Board=board_num, router_num, Interface_num, IP
Parms = {
      name=network_name
/parms
```

Where:

1. Net_num is a unique identifier for the backplane (starts at 0)
2. IP,Mask are the address of the backplane network and netmask
3. Board= is used to specify the board number, router, IP address, and interface of each router connected to the backplane
4. The supported parms are:

    a. Name= (name of the backplane)

In version 2.0 This is not needed if two routers are connected across boards. Note in the example network router XX on board 2 is directed connected to router YY on board 3, while a backplane network is used when a router wants to connect to either the board interfaced to keyhole or to the board running the default root router and the ISEAGE tap.

DLINK is used to define the link characteristics for the default links. The syntax is shown below and information on the link characteristics is provided below.

```
DLINK={IP, CIMORE} [,Module]
Parms = {
      # Optional set of parms based on the needs of the router
      name=generic link
      # 1 percent loss rate
      loss=1.0
/parms
```

Where:

1. IP indicates the default IP network
2. CIMORE specifies the default CIMORE network
3. The optional modules is used to indicate what dynamic link characteristic module will be included at run time (This is a planned feature to support link errors, delay, etc.)
4. The optional params are used to either pass information to the link and/or to the module or to be used by other tools within ISEAGE, examples are link data rate, link capacity, etc.
5. The supported parms are:
    a. Name= (name of the link)
    b. Loss=% of loss (not supported)

GLINK is used to define the link characteristics for the globally defined links. More information on the link characteristics is provided below.

```
GLINK=Link_num,{IP, CIMORE} [,Module]
Parms = {
      #Optional set of parms based on the needs of the router
      name=IP link with loss
      # 1 percent loss rate
      loss=1.0
/parms
```

Where:

1. link_num is a unique identifier within the ISEAGE configuration (starts at 1).

2. IP indicates the default IP network
3. CIMORE specifies the default CIMORE network
4. The optional modules is used to indicate what dynamic link characteristic module will be included at run time (This is a planned feature to support link errors, delay, etc.)
5. The optional params are used to either pass information to the link and/or to the module or to be used by other tools within ISEAGE, examples are link data rate, link capacity, etc.
6. The supported parms are:
    a. Name= (name of the link)
    b. Loss=% of loss (not supported)

Board Section

The board section is used to define the characteristics of each board. These include the definitions of the device on the boards (routers, clouds, etc) and how they are connected.

The format of the board directive is:

```
Board=board_num // Indicates the start of a new board (1 to N)

#connections section must be first
Connections={
     Router_num,Int => [O, R#,int BP# N# C#,Int], link_type, [module]
/connections


Device=[router, cloud, link, network]
     #directives for each device
/[router, cloud, link, network]
```

The **Board=Board_num** directive is used to identify when a new board starts. As of now there you need to cut and paste if you want to reuse sections. Future work should add an include directive to the configuration file to allow for the inclusion of individual board configuration files.

The **Connections** directive is used to describe the interconnection of the devices. The format of each **Connections** type is shown below:

```
Connections={

     #router connected to an outside interface
     RRouter_num,Int => O
     #router to router connection
     RRouter_num,Int => router_num,Int,link_type, [link_num]
     #router to backplane connection
     RRouter_num,Int => Bplackplane_num,Int,link_type, [link_num]
/connections
```

Where:

1. Router_num is the number of the router and Int is the interface number on the router.
2. backplane_num is the number of the backplane.
3. Type is the type of connection (IP, CIMORE, etc.)
4. The optional link number is used to specify a particular link type. If there is no link_num specified then link_num 0 (the default link) is assumed. Note a link_num can be used more than once.

The **Device** directive is used to indicate which device is being specified. The router device is used to define routers within a board and the cloud defines a group of routers with a single path through them. The link device defines a link that is used to connect two routers. It should be noted that two routers do not need to have link to connect them and that all of the routers within a cloud have not link devices. In version 1.0 only the router device is supported. The link device is parsed, but only a simple IP type is supported. The format of each **Device** type is shown below:

Router

```
Device=router, router_num
      If_out=num,IP,mask[,MAC]
      If=Num, IP, Mask
      If=Num, IP, Mask
      R_table=dest_ip, mask, next_ip, next_interface
      R_table=dest_ip, mask, next_ip, next_interface
      R_table=dest_ip, mask, next_ip, next_interface
      Parms = {
            #Optional set of parms based on the needs of the router
      /parms
/router
```

Where:

1. Router_num is a unique identifier within the board (starts at 0)
2. The if and if_out directive is used to specify interfaces on the router
3. num is the interface number
4. IP is the IP address of the interface
5. Mask is the network mask of the network that is connected to the interface
6. MAC is the MAC address associated with the IP address. Note "::" can be used as a MAC address to indicate the address should be auto generated.
7. The optional params are used to either pass information to the router or to be used by other tools within ISEAGE, like the GPS coordinates of the router.
8. The supported parms are:
    a. Name= (name of the link)
    b. Type = [root, edge]
        i. The type parm is used to specify special routers.
        ii. The root router is the default router. The route table will be built dynamically and will contain paths to all know routers. All traffic it cannot route either be tossed or sent out an outside interface. A board containing this router must be listed after all boards in the configuration file. Currently this can be the only router on the board.

iii. The edge router is like a root router except all un-routable traffic is sent to the root router. This is use for routers like the one connected to keyhole. If using both an edge router and a root router the edge router is listed before the root router and the edge router will need to have a default route specified.

Cloud

```
Device=cloud, cloud_num
If0=IP,Mask[, out,MAC], in]
If1=IP,Mask[, out,MAC], in]
Router=router_num, IP0, IP1
Router=router_num, IP0, IP1
Parms = {
Optional set of parms based on the needs of the cloud
/parms
/cloud
```

Where:

1. Cloud_num is a unique identifier within the board (starts at 1)
2. The if0 directive is used to specify one interface to the cloud
3. The if1 directive is used to specify the second interface to the cloud
4. IP is the IP address of the interface
5. Mask is the network mask of the network that is connected to the interface
6. The optional value "out" is used to indicate the interface is connected to the physical outside interface of the board. MAC is the MAC address associated with the IP address. Note "::" can be used as a MAC address to indicate the address should be auto generated.
7. The optional value "in" is used to indicate the interface is connected to the backplane.
8. The router directive is used to specify the routers inside the cloud. Where router_num is the number of the router. Router_num 0 is the router connected to if0 and the last router in the list is connected to if1. These router addresses are used for handling ICMP messages (ping) or for handling traceroute through the cloud.
9. The optional params are used to either pass information to the cloud or to be used by other tools within ISEAGE.

Link

```
Device=link, link_num, {IP, CIMORE} [,Module]

Parms = {
     #Optional set of parms based on the needs of the router
/parms
/link
```

Where:

1. link_num is a unique identifier within the board (starts at 0). Link_num is the default link definition.
2. IP indicates this an IP link
3. CIMORE indicates this link supports CIMORE
4. The optional module allows for special routines to be included to process the link
5. The optional params are used to either pass information to the link and/or to the module or to be used by other tools within ISEAGE, examples are link data rate, link capacity, etc.

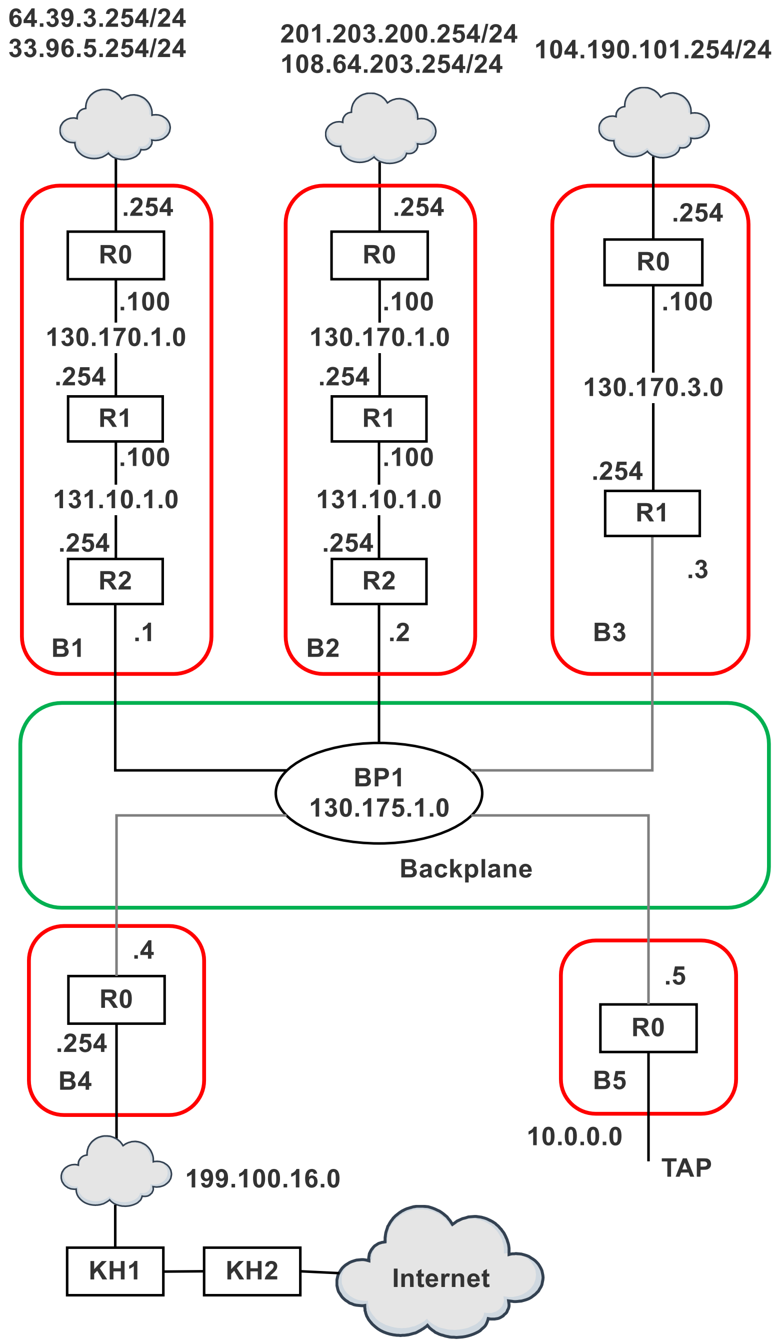
Network

```
Device=network, net_num
IP=network_ip/CIDR
Parms = {
Optional set of parms based on the needs of the network
/parms
/network
```

Where:

1. net_num is a unique identifier within the board (starts at 0).
2. IP defines the IP address range of the network and CIDR is the netmask for the network
3. The optional params are used to either pass information to the network or to be used by other tools within ISEAGE, examples are link data rate, link capacity, etc.

An example configuration file is shown below for a simple test network. The test network is shown in the diagram below:



This is the config file for the figure above (first board only)

```
#this is a test file to be used by ISEFlow
#this uses the new config file format
VER=1,0
#
# Global definitions
```

```
#
Globals=   {
      BPnet=0,130.175.1.0,24
           board=1,2,1,130.175.1.1
           board=2,2,1,130.175.1.2
           board=3,1,1,130.175.1.3
           board=4,0,1,130.175.1.4
           board=5,0,1,130.175.1.5
      Parms = {
           name=Primary Backplane
           #board= board, router, interface, IP
      /parms
      BPnet=1,10.10.10.0,24
      Parms = {
           name=KeyHole Backplane
      /parms
      #define default link for type = IP
      DLINK=IP
      Parms = {
           name=generic link
      /parms

      #define gloabl link 1
      GLINK=1,IP
      Parms = {
           name=IP link with loss
           # 1 percent loss rate
           loss=1.0
      /parms
/globals

#
# Start board definitions
#
board=1
connections={
# router,interface => router,interface, type
# router,interface => backplane, type
R0,0 => O
R0,1 => O
R0,2 => R1,0, IP
R1,1 => R2,0, L, 1
R2,1 => B0, G, 1
#R2:2 => B1, IP
/connections

device=router,0
if_out=0,64.39.3.254,24,00:00:0c:31:01:aa
if_out=1,33.96.5.254,24,00:00:0c:31:01:bb
if=2,130.170.1.100,24
# Dest IP, mask, next IP, interface
r_table=64.39.3.0,24,64.39.3.254,0
```

```
r_table=33.96.5.0,24,33.96.5.254,1
r_table=0.0.0.0,24,130.170.1.254,2
Parms = {
name=Outside router 1
/parms
/router

device=router,1
if=0,130.170.1.254,24
if=1,131.10.1.100,24
#To outside
r_table=64.39.3.0,24,130.170.1.100,0
r_table=33.96.5.0,24,130.170.1.100,0
#default router 2
r_table=0.0.0.0,24,131.10.1.254,1
Parms = {
name=b1r1
/parms
/router

device=router,2
if=0,131.10.1.254
if=1,130.175.1.1,24
#if=2,10.10.10.1,in
#Toward outside
r_table=64.39.3.0,24,131.10.1.100,0
r_table=33.96.5.0,24,131.10.1.100,0
r_table=130.170.1.0,24,131.10.1.100,0
#proxy board 4
r_table=199.100.16.0,24,130.175.1.4,1
#default board 5 router 1
r_table=0.0.0.0,24,130.175.1.5,1
Parms = {
name=b1r2
/parms
/router

#define link 1
device=link,1,IP
Parms = {
name=IP link with loss
# 1 percent loss rate
loss=1.0
/parms
/link

/board
```