The Router Plugins system architecture
Router Plugins Vocabulary

6-tuple: **src_addr**, **dest_addr**, **protocol**, **interface**, **src_port**, **dst_port**

flow: **IP packet stream that is identified by a specific 6-tuple**

flow table: **cache of existing flow information**

loadable kernel module: **executable code which can be dynamically loaded into the NetBSD kernel and subsequently executed.**

gate: **spot in the NetBSD IP processing code where Router Plugin modules can be executed.**
Router Plugins Vocabulary (continued)

**plugin:** A loadable kernel module (lkm) that can be associated with a Router Plugin gate.

**instance:** An instantiation of a plugin.

**filter:** A 6-tuple, possibly containing wildcards.

**filter table:** For each gate, there may be a number of filters. The collection of these filters is a filter table.

**dag:** Filter tables are implemented as a directed acyclic graph (dag).
Router Plugins Control Path

- **Plugin Manager**
  - set filters
  - register with a filter

- **RSVP/Route Daemons**
  - send messages to plugins
  - set filters

- **Association Identification Unit (AIU)**
  - (flow and filter tables)

- **Plugin Control Unit (PCU)**
  - Firewall
  - IP Sec
  - Packet Scheduling

User Space Kernel

Washington University in St. Louis
Data Path

Association Identification Unit (AIU)

Plugin Control Unit
- Firewall
- IP Sec
- Packet Scheduling

classifypacket
lookupfiltersettingsforplugintype“Firewall”
findplugin
WhichPlugin?
callplugin

IP
Firewall
IP Sec
Packet Scheduling
Data Path

Plugin Control Unit

Association Identification Unit (AIU)

Firewall

IP Sec

Packet Scheduling

return plugin to gate

which plugin?
call plugin
Data Path

Association Identification Unit (AIU)

Plugin Control Unit

Firewall
IP Sec
Packet Scheduling

which plugin?
return plugin to gate
call plugin

Firewall
IP Sec
Packet Scheduling
## AIU Implementation: DAG

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: 129.*</td>
<td>192.94.233.10</td>
<td>TCP</td>
</tr>
<tr>
<td>2: 128.252.153.1</td>
<td>128.252.153.7</td>
<td>UDP</td>
</tr>
<tr>
<td>3: 128.252.153.1</td>
<td>128.252.153.7</td>
<td>TCP</td>
</tr>
<tr>
<td>4: 128.252.153.*</td>
<td>*</td>
<td>UDP</td>
</tr>
</tbody>
</table>
AIU Implementation: Flow Table

AIU

Filter Table

Flow Table

<,,,,,,>

Plugin Control Unit

Firewall

IP Sec

Packet Scheduling

Classify

Firewall

IP Sec

Packet Scheduling
AIU Implementation: Flow Table

AIU
Filter Table
Flow Table
FW2 IPS3 PS1

Piugin Control Unit
Firewall IP Sec Packet Scheduling

Firewall IP Sec Packet Scheduling

Router Plugins (Crossbow)
AIU Implementation: Flow Table

Flow Table

Filter Table

AIU

Plugin Control Unit

Firewall IP Sec Packet Scheduling

Filter Table

Firewall IP Sec Packet

<,,,,,,>

FW2 IPS3 PS1

Classify

IP

Firewall

IP Sec

Packet Scheduling

Router Plugins (Crossbow)
AIU Implementation: Flow Table

- **AIU**
  - Filter Table
  - Flow Table
    - FW2
    - IPS3
    - PS1

- **Plugin Control Unit**
  - Firewall
  - IP Sec
  - Packet Scheduling

- **Router Plugins (Crossbow)**
Router Plugins - The Control Path

Plugin Manager

RSVP/SSP/Route daemons

User Kernel

IPv4/IPv6-core

Association Identification Unit (AIU)

Plugin Control Unit (PCU)

- sets filters
- message to plugins
- registers instance
- registers callback fct
- forwards messages

Filters:
- IPSEC
- PS
- BMP
- FW

- SEC1
- SEC2
- SEC3
- PS1
- PS2
- PS3
- BMP1
- BMP2
- BMP3
- FW1
- FW2
- FW3
Crossbow library

Plugin Manager

RSVP/SSP/Route daemons

Crossbow Socket

User Kernel

IPv4/IPv6-core

Plugin Control Unit (PCU)

Association Identification Unit (AIU)

Crossbow Library

registers callback fct

forwards messages

registers instance

message to plugins

sets filters

User

Kernel

Router Plugins (Crossbow)
Review: Router Plugins Vocabulary

6-tuple: src_addr, dest_addr, protocol, interface, src_port, dst_port
flow: IP packet stream that is identified by a specific 6-tuple
flow table: cache of existing flow information
loadable kernel module: executable code which can be dynamically loaded into the NetBSD kernel and subsequently executed.
gate: spot in the NetBSD IP processing code where Router Plugin modules can be executed.
dynamic gate: bind plugin instances to a filters (hence, different flows MAY cause execution of different instances or plugins?)
static gate: bind plugin instances to gate (all flows execute same inst.)
plugin: A loadable kernel module (lkm) that can be associated with a Router Plugin gate.
instance: An instantiation of a plugin.
filter: A 6-tuple, possibly containing wildcards.
filter table: For each dynamic gate, there may be a number of filters. The collection of these filters is referred to as a filter table.
dag: Filter tables are implemented as a directed acyclic graph (dag).
RP - additional facts

• Plugins create instances
  – Instances are addressable run-time configurations of plugins
  – Every instance has a handle (16 bit number)
  – System interfaces mainly with instances

• Plugins are addressable using 32 bit identifier (plugin code)

• Filters, Gates, and Instances are addressable using handles (16 or 32 bit numbers)

• Hybrid implementation of IPv4/v6

• Crossbow <=> Router Plugins