Example: Multiple Port Mirroring with Next-Hop Groups Configuration

When you need to analyze traffic containing more than one packet type, or you wish to perform multiple types of analysis on a single type of traffic, you can implement multiple port mirroring and next-hop groups. You can make up to 16 copies of traffic per group and send the traffic to next-hop group members. A maximum of 30 groups can be configured on a router at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET/SDH, aggregated Ethernet, loopback (lo0), or administrative (fxp0) interfaces. To send port-mirrored traffic to multiple flow servers or packet analyzers, you can use the next-hop-group statement at the [edit forwarding-options] hierarchy level.

Figure 1: Active Flow Monitoring—Multiple Port Mirroring with Next-Hop Groups Topology Diagram

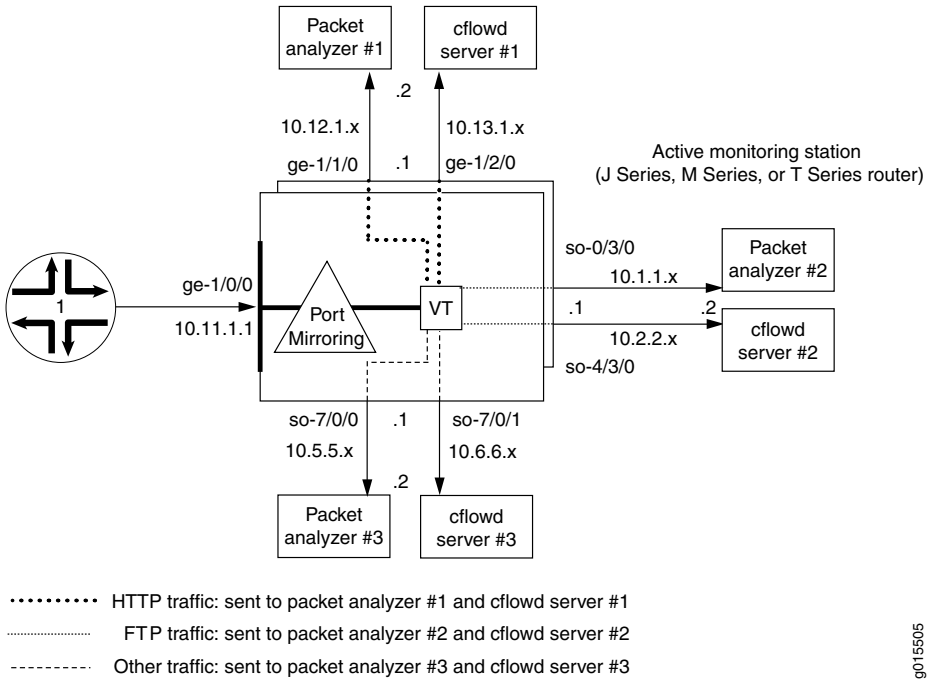

Figure 1 shows an example of how to configure multiple port mirroring with next-hop groups. All traffic enters the monitoring router at interface ge-1/0/0. A firewall filter counts and port-mirrors all incoming packets to a Tunnel Services PIC. A second filter is applied to the tunnel interface and splits the traffic into three categories: HTTP traffic, FTP traffic, and all other traffic. The three types of traffic are assigned to three separate next-hop groups. Each next-hop group contains a unique pair of exit interfaces that lead to different groups of packet analyzers and flow servers.

```
[edit]
interfaces {
   ge-1/0/0 { # This is the input interface where packets enter the router.
      unit 0 {
         family inet {
```

```
        filter {
            input mirror_pkts; # Here is where you apply the first filter.
        }
        address 10.11.1.1/24;
      }
    }
}
ge-1/1/0 { # This is an exit interface for HTTP packets.
    unit 0 {
      family inet {
        address 10.12.1.1/24;
      }
    }
}
ge-1/2/0 { # This is an exit interface for HTTP packets.
    unit 0 {
      family inet {
        address 10.13.1.1/24;
      }
    }
}
so-0/3/0 { # This is an exit interface for FTP packets.
    unit 0 {
      family inet {
        address 10.1.1.1/30;
      }
    }
}
so-4/3/0 { # This is an exit interface for FTP packets.
    unit 0 {
      family inet {
        address 10.2.2.1/30;
      }
    }
}
so-7/0/0 { # This is an exit interface for all remaining packets.
    unit 0 {
      family inet {
        address 10.5.5.1/30;
      }
    }
}
so-7/0/1 { # This is an exit interface for all remaining packets.
    unit 0 {
      family inet {
        address 10.6.6.1/30;
      }
    }
}
vt-3/3/0 { # The tunnel interface is where you send the port-mirrored traffic.
    unit 0 {
      family inet;
    }
    unit 1 {
      family inet {
        filter {
```

```
            input collect_pkts;  # This is where you apply the second firewall filter.
          }
        }
      }
    }
}
forwarding-options {
    port-mirroring { # This is required when you configure next-hop groups.
      family inet {
        input {
          rate 1; # This port-mirrors all packets (one copy for every packet received).
        }
        output { # Sends traffic to a tunnel interface to enable multiport mirroring.
          interface vt-3/3/0.1;
          no-filter-check;
        }
      }
    }
    next-hop-group ftp-traffic { # Point-to-point interfaces require you to specify the
      interface so-4/3/0.0; # interface name.
      interface so-0/3/0.0;
    }
    next-hop-group http-traffic { # Configure a next hop for all multipoint interfaces.
      interface ge-1/1/0.0 {
        next-hop 10.12.1.2;
      }
      interface ge-1/2/0.0 {
        next-hop 10.13.1.2;
      }
    }
    next-hop-group default-collect {
      interface so-7/0/0.0;
      interface so-7/0/1.0;
    }
}
firewall {
    family inet {
      filter mirror_pkts { # Apply this filter to the input interface.
        term catch_all {
          then {
            count input_mirror_pkts;
            port-mirror; # This action sends traffic to be copied and port-mirrored.
          }
        }
      }
      filter collect_pkts { # Apply this filter to the tunnel interface.
        term ftp-term { # This term sends FTP traffic to an FTP next-hop group.
          from {
            protocol ftp;
          }
          then next-hop-group ftp-traffic;
        }
        term http-term { # This term sends HTTP traffic to an HTTP next-hop group.
          from {
            protocol http;
          }
```

```
            then next-hop-group http-traffic;
        }
        term default { # This sends all remaining traffic to a final next-hop group.
            then next-hop-group default-collectors;
        }
    }
  }
}
```

Published: 2010-04-15