

## Example Configuration for a Routing Matrix with a TX Matrix Plus Router

The following sections describe an example configuration for a routing matrix with a TX Matrix Plus router:

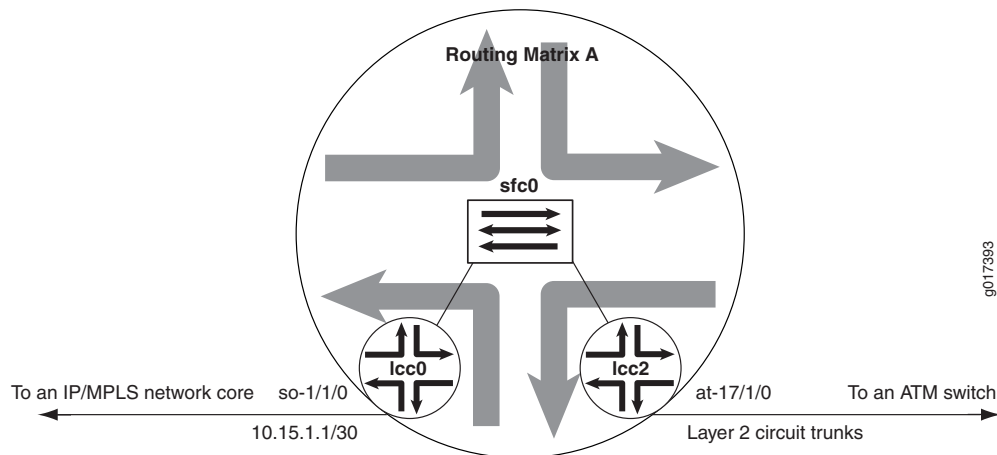
- Example Routing Matrix Topology on page 1
- Example Routing Matrix Configuration on page 2
- Verifying the Example Configuration on page 7

### Example Routing Matrix Topology

This example is based on **Routing Matrix A**, a basic routing matrix consisting of a TX Matrix Plus router (**sfc0**) and two T1600 routers (**lcc0** and **lcc2**). Figure 1 shows that **Routing Matrix A** is acting as a provider edge (PE) router in a Layer 2 circuit network. SONET interface **so-1/1/0** in the T1600 router designated as **lcc0** connects to an IP/MPLS core network. Asynchronous Transfer Mode 2 (ATM2) intelligent queuing (IQ) interface **at-17/1/0** in the T1600 router designated as **lcc2** runs Layer 2 circuit trunk mode to connect to an ATM switch.

For more information about Layer 2 circuit networks, see the *JUNOS VPNs Configuration Guide*.

**Figure 1: Example Topology of a Routing Matrix with a TX Matrix Plus Router**



Note the following key considerations for this routing matrix configuration:

- For most operations, you will manage the routing matrix as a single router, executing operational commands and committing configurations on the TX Matrix Plus router designated as **sfc0**.
- Create configuration groups for each Routing Engine in the routing matrix by using the special configuration groups for a routing matrix consisting of routers with dual Routing Engines: **re0**, **re1**, **lcc0-re0**, **lcc2-re0**, **lcc0-re1**, and **lcc2-re1**. In these Routing Engine configuration groups, configure hostnames, default routes, and management Ethernet interfaces.

- To configure interfaces, use the routing matrix FPC numbering convention of slots 0 through 31.
- To enable ATM2 IQ trunk mode and other chassis-based commands, include the `lcc lcc-number` statement at the `[edit chassis]` hierarchy level and use the hardware FPC slot numbers 0 through 7 of node `lcc2`.
- Configure most other processes as usual, such as routing, class of service (CoS), and firewalls.

## Example Routing Matrix Configuration

To configure the example **Routing Matrix A** described in the previous section, commit the following configuration at the TX Matrix Plus router:

```
[edit]
groups { # Create special configuration groups in a routing matrix.
  re0 { # Create a group for the master Routing Engine on TX Matrix Plus router sfc0.
    system {
      host-name mysfc0;
      backup-router 192.168.17.254;
    }
    interfaces { # Configure an IP address for the management interface logical port.
      em0 {
        unit 0 {
          family inet {
            address 192.168.77.158/21;
          }
        }
      }
    }
  }
  re1 { # Create a group for the backup Routing Engine on TX Matrix Plus router sfc0.
    system {
      host-name mysfc0_alt_re;
      backup-router 192.168.17.254;
    }
    interfaces { # Configure an IP address for the management interface logical port.
      em0 {
        unit 0 {
          family inet {
            address 192.168.77.168/21;
          }
        }
      }
    }
  }
  lcc0-re0 { # Create a group for the master Routing Engine on T1600 router lcc0.
    system {
      host-name mylcc0;
      backup-router 192.168.17.254 destination [10.0.0.0/8 192.168.0.0/16];
    }
    interfaces { # Configure an IP address for the management interface logical port.
      em0 {
        unit 0 {
          family inet {
```

```

        address 192.168.77.157/21;
    }
}
}
}
}
lcc0-re1 { # Create a group for the backup Routing Engine on T1600 router lcc0.
system {
    host-name mylcc0_alt_re;
    backup-router 192.168.17.254 destination [10.0.0.0/8 192.168.0.0/16];
}
interfaces { # Configure an IP address for the management interface logical port.
    em0 {
        unit 0 {
            family inet {
                address 192.168.77.169/21;
            }
        }
    }
}
}
lcc2-re0 { # Create a group for the master Routing Engine on T1600 router lcc2.
system {
    host-name mylcc2;
    backup-router 192.168.17.254 destination [10.0.0.0/8 192.168.0.0/16];
}
interfaces { # Configure an IP address for the management interface logical port.
    em0 {
        unit 0 {
            family inet {
                address 192.168.77.159/21;
            }
        }
    }
}
}
lcc2-re1 { # Create a group for the backup Routing Engine on T1600 router lcc2.
system {
    host-name mylcc2_alt_re;
    backup-router 192.168.17.254 destination [10.0.0.0/8 192.168.0.0/16];
}
interfaces { # Configure an IP address for the management interface logical port.
    em0 {
        unit 0 {
            family inet {
                address 192.168.77.192/21;
            }
        }
    }
}
}
}
apply-groups [ re0 re1 lcc0-re1 lcc2-re1 lcc0-re0 lcc2-re0 ]; # Enable inheritance.
system { # Configure system management properties.
    syslog {
        file messages {

```

```

        any any;
    }
}
chassis { # You must apply chassis commands to a specific T1600 router.
lcc 2 { # Specify the T1600 router and the FPC hardware slot of the node.
fpc 1 { # This FPC is equivalent to slot 17 in the routing matrix.
    pic 1 {
        atm-l2circuit-mode {
            trunk nni;
        }
    }
}
}
}
}
}
interfaces {
    so-1/1/0 { # This is a SONET interface at FPC 1, PIC 1, port 0
        mtu 9192; # on the T1600 router LCC0.
        unit 0 {
            family inet {
                address 10.15.1.1/30 {
                    destination 10.15.1.2;
                }
            }
            family iso;
            family mpls {
                filter {
                    input filter_1;
                }
            }
        }
    }
}
    at-17/1/0 { # This is an ATM2 IQ interface at FPC 1, PIC 1, port 0
        encapsulation atm-ccc-cell-relay; # on the T1600 router LCC2.
        atm-options {
            pic-type atm2;
            scheduler-maps { # CoS on an ATM2 IQ PIC works the same in a routing matrix.
                cos1 { # as it does in a standalone T1600 router.
                    forwarding-class ubr {
                        priority low;
                        transmit-weight percent 25;
                    }
                    forwarding-class nrtvbr {
                        priority low;
                        transmit-weight percent 25;
                    }
                    forwarding-class rtvbr {
                        priority low;
                        transmit-weight percent 25;
                    }
                    forwarding-class cbr {
                        priority high;
                        transmit-weight percent 25;
                    }
                }
                cos2 {

```

```

        forwarding-class ubr {
            priority low;
            transmit-weight percent 10;
        }
        forwarding-class nrtvbr {
            priority low;
            transmit-weight percent 20;
        }
        forwarding-class rtvbr {
            priority low;
            transmit-weight percent 30;
        }
        forwarding-class cbr {
            priority high;
            transmit-weight percent 40;
        }
    }
    cos3 {
        forwarding-class ubr {
            priority low;
            transmit-weight percent 40;
        }
        forwarding-class nrtvbr {
            priority low;
            transmit-weight percent 30;
        }
        forwarding-class rtvbr {
            priority low;
            transmit-weight percent 20;
        }
        forwarding-class cbr {
            priority high;
            transmit-weight percent 10;
        }
    }
}
unit 0 {
    trunk-id 0;
    trunk-bandwidth 10m;
    cell-bundle-size 2;
}
unit 1 {
    trunk-id 1;
    trunk-bandwidth 10m;
    cell-bundle-size 1;
    atm-scheduler-map cos1;
}
unit 2 {
    trunk-id 2;
    trunk-bandwidth 10m;
    cell-bundle-size 2;
    atm-scheduler-map cos2;
}
unit 3 {
    trunk-id 3;

```

```

        trunk-bandwidth 10m;
        cell-bundle-size 3;
        atm-scheduler-map cos3;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 127.0.0.1/32;
            address 10.255.77.158/32 {
                primary;
            }
        }
        family iso {
            address 47.0005.80ff.f800.0000.0108.0001.0102.5507.0158.00;
        }
        family inet6 {
            address 2001:db8::10:255:77:158/32 {
                primary;
            }
        }
    }
}
}
protocols { # You can configure protocols in the routing matrix as usual.
    mpls {
        interface so-1/1/0.0;
    }
    isis {
        interface so-1/1/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-1/1/0.0;
        interface lo0.0;
    }
    l2circuit {
        neighbor 10.255.71.97 {
            interface at-17/1/0.0 {
                virtual-circuit-id 100;
            }
            interface at-17/1/0.1 {
                virtual-circuit-id 101;
            }
            interface at-17/1/0.2 {
                virtual-circuit-id 102;
            }
            interface at-17/1/0.3 {
                virtual-circuit-id 103;
            }
        }
    }
}
class-of-service { # You can configure CoS in the routing matrix as usual.
    forwarding-classes {
        queue 0 ubr;
    }
}

```



- `show route summary`
- `show system uptime < all-lcc | lcc lcc-number | sfc sfc-number >`
- `show version < all-lcc | lcc lcc-number | sfc sfc-number >`

For detailed descriptions of how to use these commands to verify the routing matrix configuration, see [Verifying the Configuration of a Routing Matrix with a TX Matrix Plus Router](#).

- Related Topics**
- [Routing Matrix with a TX Matrix Plus Router Solutions Page](#)
  - [Overview of a Routing Matrix with a TX Matrix Plus Router](#)
  - [Roadmap for Configuring the Routing Matrix](#)
  - [Verifying the Configuration of the Routing Matrix](#)
  - [Merging Examples](#)
  - [Upgrading the JUNOS Software on the Routing Matrix](#)

---

Published: 2010-04-15