Configuring PPP over ATM2 Encapsulation

For ATM2 IQ interfaces, you can configure PPP over AAL5 encapsulation, as described in RFC 2364, *PPP over AAL5*. PPP over ATM encapsulation associates a PPP link with an ATM AAL5 PVC.

The JUNOS software supports three PPP over ATM encapsulation types:

- atm-ppp-llc—PPP over AAL5 LLC.
- atm-ppp-vc-mux—PPP over ATM AAL5 multiplex.
- atm-mlppp-llc—Multilink PPP over ATM AAL5 LLC. For this encapsulation type, your routing platform must be equipped with a Link Services or Voice Services PIC. MLPPP over ATM encapsulation is not supported on ATM2 IQ OC48 interfaces.

To enable PPP over ATM encapsulation, include the encapsulation statement, specifying the atm-mlppp-llc, atm-ppp-llc, or atm-ppp-vc-mux encapsulation type:

[Unresolved xref] (atm-mlppp-llc | atm-ppp-llc | atm-ppp-vc-mux);

You can include this statement at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

When you configure PPP over ATM encapsulation, you can enable PPP Challenge Handshake Authentication Protocol (CHAP) and keepalives on the logical interface. For more information about PPP CHAP and keepalives, see Configuring the PPP Challenge Handshake Authentication Protocol and Configuring Keepalives.

NOTE: When you use PPP over ATM encapsulation, we recommend that you not include the oam-period statement in the configuration. Instead, we recommend that you enable keepalives to detect connection failures.

Example: Configuring PPP over ATM2 IQ Encapsulation

Configure three logical interfaces with PPP over ATM encapsulation:

```
[edit interfaces]
at-0/1/0 {
    atm-options {
        pic-type atm2;
        vpi 0;
        vpi 2;
    }
    unit 0 {
        encapsulation atm-ppp-llc;
```

```
      ppp-options {
        chap {
          access-profile pe-B-ppp-clients;
          local-name "pe-A-at-0/1/0";
        }
      }
      keepalives interval 5 up-count 6 down-count 4;
      vci 0.120;
      family inet address 192.168.13.13/30;
    }
    unit 1 {
      encapsulation atm-ppp-vc-mux;
      vci 2.120;
      keepalives interval 6 up-count 6 down-count 4;
      family inet address 192.168.14.13/30;
    }
    unit 2 {
      encapsulation atm-ppp-vc-mux;
      ppp-options {
        chap {
          passive;
          access-profile pe-A-ppp-clients;
          local-name "pe-A-at-0/1/0";
        }
      }
      keepalives interval 5 up-count 6 down-count 4;
      vci 2.121;
      family inet address 192.168.15.13/30;
    }
  }
```

Configuring Multilink PPP over ATM2 IQ Encapsulation

```
[edit interfaces]
at-0/0/0 {
  atm-options {
    pic-type atm2;
    vpi 10;
  }
  unit 0 {
    encapsulation atm-mlppp-llc;
    ppp-options {
      chap {
        access-profile pe-B-ppp-clients;
        local-name " pe-A-at-0/0/0";
      }
    }
    keepalive interval 5 up-count 6 down-count 4;
    vci 10.120;
    family mlppp {
      bundle ls-0/3/0.0;
    }
  }
}
at-0/0/1 {
  atm-options {
    pic-type atm2;
    vpi 11;
```

```
    }
    unit 1 {
      encapsulation atm-mlppp-llc;
      ppp-options {
        chap {
          access-profile pe-B-ppp-clients;
          local-name " pe-A-at-0/0/0";
        }
      }
      keepalive interval 5 up-count 6 down-count 4;
      vci 11.120;
      family mlppp {
        bundle ls-0/3/0.0;
      }
    }
}
at-1/2/3 {
    atm-options {
      pic-type atm2;
      vpi 12;
    }
    unit 2 {
      encapsulation atm-mlppp-llc;
      ppp-options {
        chap {
          access-profile pe-B-ppp-clients;
          local-name " pe-A-at-0/0/0";
        }
      }
      keepalive interval 5 up-count 6 down-count 4;
      vci 12.120;
      family mlppp {
        bundle ls-0/3/0.0;
      }
    }
}
...
ls-0/3/0 {
    encapsulation multilink-ppp;
    interleave-fragments;
    keepalive;
    unit 0 {
      mrru 4500;
      short-sequence;
      fragment-threshold 16320;
      drop-timeout 2000;
      encapsulation multilink-ppp;
      interleave-fragments;
      minimum-links 8;
      family inet {
        address 10.10.0.1/32 {
          destination 10.10.0.2;
        }
      }
      family iso;
      family inet6 {
```

```
            address 8090::0:1/128 {
               destination 8090::0:2;
            }
         }
      }
      ...
}
```