

## Chapter 19

# Real-Time Operational Mode Commands

Table 36 summarizes the command-line interface (CLI) commands you can use to monitor and troubleshoot router interfaces in real time. In the table, the commands are grouped by functionality. In the remainder of this chapter, they are explained alphabetically.

**Table 36: Commands for Real-Time Monitoring and Troubleshooting of Router Interfaces**

Task or Information to Monitor	CLI Command
Interface statistics	monitor interface on page 297
TCP dump	monitor traffic on page 300
Automatic Protection Switching (APS)	show aps on page 303

## Monitor Traffic Match Conditions

In the monitor traffic command, you can specify an expression to match by using the matching option and including the expression in quotation marks:

```
monitor traffic matching "expression"
```

*expression* is one or more of the match conditions listed in Table 37. To combine expressions, use the logical operators listed in Table 38 on page 295. For match conditions that have a numeric value, you can specify them using the arithmetic and relational operators listed in Table 39 on page 295.

**Table 37: Match Conditions for the Monitor Traffic Command**

Match Condition	Description
Entity Type	Entity types match packets according to hostnames and addresses, network addresses, port names and numbers, and subnet masking.
host [ <i>address</i>   <i>hostname</i> ]	Matches packets that contain the specified address or hostname. The host match condition can be prepended with the protocol match conditions arp, ip, or rarp, or any of the directional match conditions.
network <i>address</i>	Matches packets with source or destination addresses containing the specified network number.
network <i>address</i> mask <i>mask</i>	Matches packets containing the specified network address and subnet mask.

Match Condition	Description
port [ <i>port-number</i>   <i>port-name</i> ]	Matches packets containing the specified source or destination TCP or UDP port number or port name.  In place of the numeric port address, you can specify a text synonym (the port numbers are also listed). The following are examples of port address text synonyms: bgp (179), dhcp (67), domain (53).
Directional	Refines entity type match conditions based on source and destination addressing. Directional match conditions are used only in conjunction with other match conditions.
destination	Matches packets going to the specified destination. This match condition can be prepended to any of the entity type match conditions.
source	Matches packets from a specified source. This match condition can be prepended to any of the entity type match conditions.
source and destination	Matches packets that contain the specified source and destination addresses. This match condition can be prepended to any of the entity type match conditions.
source or destination	Matches packets containing either of the specified addresses. This match condition can be prepended to any of the entity type match conditions.
Packet Length	Matches packets according to packet size.
less <i>value</i>	Matches packets shorter than or equal to the specified value, in bytes.
greater <i>value</i>	Matches packets longer than or equal to the specified value, in bytes.
Protocol	Restricts matching condition to a particular protocol.
arp	Matches all ARP packets.
ether	Matches all Ethernet packets.
ether [ <i>broadcast</i>   <i>multicast</i> ]	Matches broadcast or multicast Ethernet frames. This match condition can be prepended with source and destination
ether protocol [ <i>address</i>   ( <i>arp</i>   <i>ip</i>   <i>rarp</i> )]	Matches packets with the specified Ethernet address or Ethernet packets of the specified protocol type.  The ether protocol arguments arp, ip, and rarp are also independent match conditions, so they must be preceded with a backslash (\) when used in the ether protocol match condition.
icmp	Matches all ICMP packets.
ip	Matches all IP packets.
ip [ <i>broadcast</i>   <i>multicast</i> ]	Matches broadcast or multicast IP packets.
ip protocol [ <i>address</i>   ( <i>icmp</i>   <i>igrp</i>   <i>tcp</i>   <i>udp</i> )]	Matches packets with the specified address or protocol type.  The ip protocol arguments icmp, tcp, and udp are also independent match conditions, so they must be preceded with a backslash (\) when used in the ip protocol match condition.
isis	Matches all IS-IS routing messages.
rarp	Matches all RARP packets.
tcp	Matches all TCP datagrams.
udp	Matches all UDP datagrams.

Table 38: Logical Operators for the Monitor Traffic Command

Logical Operator (Highest to Lowest Precedence)	Description
!	Logical NOT. If the first condition does not match, the next condition is evaluated.
&&	Logical AND. If the first condition matches, the next condition is evaluated. If the first condition does not match, the next condition is skipped.
	Logical OR. If the first condition matches, the next condition is skipped. If the first condition does not match, the next condition is evaluated.
()	Group operators to override default precedence order. Parentheses are special characters, each of which must be preceded by a backslash (\).

You can use relational operators to compare arithmetic expressions composed of integer constants, binary operators, a length operator, and special packet data accessors.

The arithmetic expression matching condition uses the following syntax:

```
monitor traffic matching "ether[0] & 1 != 0" "arithmetic_expression
relational_operator arithmetic_expression"
```

The packet data accessor uses the following syntax:

```
protocol [byte-offset <size>]
```

The optional *size* field represents the number of bytes examined in the packet header. The available values are 1, 2, or 4 bytes.

The following example captures all multicast traffic:

```
user@host> monitor traffic matching "ether[0] & 1 != 0"
```

Table 39: Arithmetic and Relational Operators for the Monitor Traffic Command

Arithmetic or Relational Operator	Description
<b>Arithmetic Operator</b>	
+	Addition operator.
-	Subtraction operator.
/	Division operator.
&	Bitwise AND.
*	Bitwise exclusive OR.
	Bitwise inclusive OR.
<b>Relational Operator (Highest to Lowest Precedence)</b>	
< =	If the first expression is less than or equal to the second, the packet matches.

<b>Arithmetic or Relational Operator</b>	<b>Description</b>
> =	If the first expression is greater than the second, the packet matches.
<	If the first expression is less than the second, the packet matches.
>	If the first expression is greater than the second, the packet matches.
=	If the compared expressions are equal, the packet matches.
!=	If the compared expressions are unequal, the packet matches.

## monitor interface

**Syntax** monitor interface <interface-name | traffic>

**Description** Display real-time statistics about a physical interface, updating them every second. The output of this command also shows the amount that each field has changed since you started the command or since you cleared the counters by using the C key. This command also checks for and displays common interface failures, such as SONET/SDH and T3 alarms, loopbacks detected, and increases in framing errors.

To control the output of the command while it is running, use the keys listed in Table 40.

**Table 40: Monitor Interface Output Control Keys**

Action	Key
Display information about the next interface. The monitor interface command scrolls through the physical or logical interfaces in the same order that they are displayed by the show interfaces terse command.	N
Display information about a different interface. The command prompts you for the name of a specific interface.	I
Freeze the display, halting the display of updated statistics.	F
Thaw the display, resuming the display of updated statistics.	T
Clear (zero) the current delta counters since monitor interface was started. It does not clear the accumulative counter. To clear the accumulative counter, see clear interfaces interval on page 11.	C
Stop the monitor interface command.	Q

**Options** *interface-name*—Name of a physical or logical interface.

*traffic*—Display traffic data for active interfaces.

**Required Privilege Level** trace

**See Also** show interfaces statistics on page 25  
monitor traffic on page 300

**Output Fields** router1—Hostname of the router.

Seconds—How long the monitor interface command has been running or how long since you last zeroed the counters.

Time—Current time (UTC).

Delay x/y/z—Time difference between when the statistics were displayed and the actual clock time.

x—Time taken for the last polling (in milliseconds).

y—Minimum time taken across all pollings (in msec).

z—Maximum time taken across all pollings (in msec).

Interface—Short description of the interface, including its name, status, and encapsulation.

Link—State of the link: Up, Down, or Test.

Current delta—Cumulative number for the counter in question since the time shown in the Seconds field, which is the time since you started the command or last zeroed the counters.

Statistics—For an explanation of the interface statistics, see the description of the show interfaces statistics detail command for the appropriate interface type.

**Sample Output: monitor interface (physical interface)**

```

user@host> monitor interface so-0/0/0
router1          Seconds: 19          Time: 15:46:29

Interface: so-0/0/0, Enabled, Link is Up
Encapsulation: PPP, Keepalives, Speed: OC48
Traffic statistics:          Current Delta
Input packets:             6045 (0 pps)          [11]
Input bytes:               6290065 (0 bps)       [13882]
Output packets:           10376 (0 pps)          [10]
Output bytes:             10365540 (0 bps)       [9418]
Encapsulation statistics:
Input keepalives:         1901                  [2]
Output keepalives:       1901                  [2]
NCP state: Opened
LCP state: Opened
Error statistics:
Input errors:             0                      [0]
Input drops:             0                      [0]
Input framing errors:    0                      [0]
Policed discards:       0                      [0]
L3 incompletes:         0                      [0]
L2 channel errors:      0                      [0]
L2 mismatch timeouts:   0                      [0]
Carrier transitions:     1                      [0]
Output errors:          0                      [0]
Output drops:          0                      [0]
Aged packets:          0                      [0]
Active alarms : None
Active defects: None
SONET error counts/seconds:
LOS count                1                      [0]
LOF count                1                      [0]
SEF count                1                      [0]
ES-S                    0                      [0]
SES-S                    0                      [0]
SONET statistics:
BIP-B1                  458871                 [0]
BIP-B2                  460072                 [0]
REI-L                   465610                 [0]
BIP-B3                  458978                 [0]
REI-P                   458773                 [0]
Received SONET overhead:
F1   : 0x00 J0   : 0x00 K1   : 0x00
K2   : 0x00 S1   : 0x00 C2   : 0x00
C2(cmp) : 0x00 F2   : 0x00 Z3   : 0x00
Z4   : 0x00 S1(cmp) : 0x00
    
```

Transmitted SONET overhead:  
 F1 : 0x00 J0 : 0x01 K1 : 0x00  
 K2 : 0x00 S1 : 0x00 C2 : 0xcf  
 F2 : 0x00 Z3 : 0x00 Z4 : 0x00

Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'

**Sample Output: monitor  
interface  
(logical interface)**

```
user@host> monitor interface so-1/0/0.0
host name                               Seconds: 16           Time: 15:33:39
                                      Delay: 0/0/1
Interface: so-1/0/0.0, Enabled, Link is Down
Flags: Hardware-Down Point-To-Point SNMP-Traps
Encapsulation: PPP
Local statistics:                       Current delta
Input bytes:                            0                   [0]
Output bytes:                            0                   [0]
Input packets:                           0                   [0]
Output packets:                           0                   [0]
Remote statistics:
Input bytes:                             0 (0 bps)           [0]
Output bytes:                             0 (0 bps)           [0]
Input packets:                            0 (0 pps)           [0]
Output packets:                            0 (0 pps)           [0]
Traffic statistics:
Destination address: 192.168.8.193, Local: 192.168.8.21
```

Next='n', Quit='q' or ESC, Freeze='f', Thaw='t', Clear='c', Interface='i'

**Sample Output: monitor  
interface traffic**

```
user@host> monitor interface traffic
host name                               Seconds: 15           Time: 12:31:09

Interface  Link  Input packets  (pps)  Output packets  (pps)
so-1/0/0   Down    0      (0)      0      (0)
so-1/1/0   Down    0      (0)      0      (0)
so-1/1/1   Down    0      (0)      0      (0)
so-1/1/2   Down    0      (0)      0      (0)
so-1/1/3   Down    0      (0)      0      (0)
t3-1/2/0   Down    0      (0)      0      (0)
t3-1/2/1   Down    0      (0)      0      (0)
t3-1/2/2   Down    0      (0)      0      (0)
t3-1/2/3   Down    0      (0)      0      (0)
so-2/0/0   Up      211035  (1)     36778  (0)
so-2/0/1   Up      192753  (1)     36782  (0)
so-2/0/2   Up      211020  (1)     36779  (0)
so-2/0/3   Up      211029  (1)     36776  (0)
so-2/1/0   Up      189378  (1)     36349  (0)
so-2/1/1   Down    0      (0)     18747  (0)
so-2/1/2   Down    0      (0)     16078  (0)
so-2/1/3   Up      0      (0)     80338  (0)
at-2/3/0   Up      0      (0)      0      (0)
at-2/3/1   Down    0      (0)      0      (0)
```

Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D

## monitor traffic

---

**Syntax** monitor traffic <absolute-sequence> <count *count*> <interface *interface*> <layer2-headers> <matching *matching*> <no-domain-names> <no-optimize> <no-resolve> <no-timestamp> <print-ascii> <print-hex> <size *size*> <brief | detail | extensive>

**Description** Print packet headers transmitted through network interfaces sent from or received by the Routing Engine.



**NOTE:** Using the monitor traffic command can negatively impact router performance. Not using print filtering functions, such as the count option or a matching expression, could impact packet throughput on your router.

---

- Options**
- absolute-sequence—(Optional) Print absolute TCP sequence numbers.
  - brief—(Optional) Display minimal protocol-related information. The monitor traffic command uses the brief option by default.
  - count *count*—(Optional) Specify the number of packet headers to display (0 through 1,000,000). The monitor traffic command quits automatically after displaying the number of packets specified.
  - detail—(Optional) Print packet data in moderate detail. For example, the detail option displays the time-to-live and type-of-service information in a TCP packet. For detailed output to be printed for some protocols, you must use the size option to increase the number of bytes of output that can be displayed for each packet matched by monitor traffic.
  - extensive—(Optional) Display the most extensive level of protocol-related output. For extensive output to be printed for certain protocols, you must use the size option to increase the output of each packet matched by monitor traffic. The extensive option displays link-level headers on each line.
  - interface *interface*—(Optional) Specify the interface on which monitor traffic prints packet data. If no interface is specified, monitor traffic prints packet data arriving on the lowest numbered interface.
  - layer2-headers—(Optional) Display the link-level header on each line.
  - matching *matching*—(Optional) Print packet headers that match a regular expression. Use matching expressions to define the level of detail with which monitor traffic filters and prints packet data.
- For information about match conditions, see “Monitor Traffic Match Conditions” on page 293.
- no-domain-names—(Optional) Suppress printing the domain portion of hostnames. With the no-domain-names option enabled, monitor traffic would print only team for the hostname team.company.net.
  - no-resolve—(Optional) Suppress symbolic addressing.
  - no-timestamp—(Optional) Suppress timestamps on printed packets.

`print-ascii`—(Optional) Print each packet in ASCII format.

`print-hex`—(Optional) Print each packet, except for the link-level header, in hexadecimal format.

`size size`—(Optional) Receive the specified number of bytes for each packet. The default size is 68 bytes and is adequate for capturing IP, ICMP, UDP, and TCP packet data. The monitor traffic command truncates printed packets if the matched data exceeds the configured `size`.

**Required Privilege Level** trace

**See Also** “Monitor Traffic Match Conditions” on page 293

**Sample Output: monitor traffic**

```
user@host> monitor traffic interface fxp0
listening on fxp0
18:17:28.800650 In server.home.net.723 > host1-0.lab.home.net.log
18:17:28.800733 Out host2-0.lab.home.net.login > server.home.net.7
18:17:28.817813 In host30.lab.home.net.syslog > host40.home0
18:17:28.817846 In host30.lab.home.net.syslog > host40.home0
18:17:28.819174 In host30.lab.home.net.syslog > host40.home0
...
```

**Sample Output: monitor traffic (relative-sequence)**

```
user@host> monitor traffic extensive no-domain-names no-resolve no-timestamp count 20 matching "tcp"
listening on fxp0
In 172.24.248.221.1680 > 192.168.4.210.23: . 396159737:396159737(0) ack 1664980689 win 17574 (DF) (ttl 121, id 50003)
Out 192.168.4.210.23 > 172.24.248.221.1680: P 1:40(39) ack 0 win 17680 (DF) [tos 0x10] (ttl 64, id 5394)
In 207.17.136.193.179 > 192.168.4.227.1024: P 4042775817:4042775874(57) ack 1845416593 win 16384
<nop,nop,timestamp 4935379 965690>: BGP [IBGP UPDAT]
...
```

**Sample Output: monitor traffic (absolute-sequence)**

```
user@host> monitor traffic extensive no-domain-names no-resolve no-timestamp count 20 matching "tcp"
absolute-sequence
listening on fxp0
In 207.17.136.193.179 > 192.168.4.227.1024: . 4042780859:4042780859(0) ack 1845421797 win 16384
<nop,nop,timestamp 4935628 965951> [tos 0xc0] (ttl )
In 207.17.136.193.179 > 192.168.4.227.1024: P 4042780859:4042780912(53) ack 1845421797 win 16384
<nop,nop,timestamp 4935628 965951>: BGP [IBGP UPDAT]
In 192.168.4.227.1024 > 207.17.136.193.179: P 1845421797:1845421852(55) ack 4042780912 win 16384
<nop,nop,timestamp 965951 4935628>: BGP [IBGP UPDAT]
...
```

**Sample Output: monitor traffic (count)**

```
user@host> monitor traffic count 2
listening on fxp0
04:35:49.814125 In my-server.home.net.1295 > my-server.work.net.telnet: . ack 4122529478 win 16798 (DF)
04:35:49.814185 Out my-server.work.net.telnet > my-server.home.net.1295: P 1:38(37) ack 0 win 17680 (DF) [tos 0x10]
```

**Sample Output: monitor traffic detail**

```
user@host> monitor traffic detail count 2
listening on fxp0
04:38:16.265864 In my-server.home.net.1295 > my-server.work.net.telnet: . ack 4122529971 win 17678 (DF) (ttl 121, id 6812)
04:38:16.265926 Out my-server.work.net.telnet.telnet > my-server.home.net.1295: P 1:38(37) ack 0 win 17680 (DF) [tos 0x10] (ttl 6)
```

**Sample Output: monitor traffic extensive**

```
user@host> monitor traffic extensive count 5 no-domain-names no-resolve
listening on fxp0
13:18:17.406933 In 192.168.4.206.2723610880 > 172.17.28.8.2049: 40 null (ttl 64, id 38367)
```

```

13:18:17.407577 In 172.17.28.8.2049 > 192.168.4.206.2723610880: reply ok 28 null (ttl 61, id 35495)
13:18:17.541140 In 0:e0:1e:42:9c:e0 0:e0:1e:42:9c:e0 9000 60:
                0000 0100 0000 0000 0000 0000 0000 0000
                0000 0000 0000 0000 0000 0000 0000 0000
                0000 0000 0000 0000 0000 0000 0000
13:18:17.591513 In 172.24.248.156.4139 > 192.168.4.210.23: . 3556964918:3556964918(0) ack 295526518 win 17601
(DF) (ttl 121, id 14)
13:18:17.591568 Out 192.168.4.210.23 > 172.24.248.156.4139: P 1:40(39) ack 0 win 17680 (DF) [tos 0x10] (ttl 64, id
52376)
    
```

## show aps

- Syntax** show aps <interface *interface-name* | group *group*> <brief | detail | extensive | summary>
- Description** Display information about Automatic Protection Switching (APS).
- Options** none—Same as brief.
- brief—(Optional) Display brief APS information.
- detail—(Optional) Display detailed APS information.
- extensive—(Optional) Display very detailed APS information.
- summary—(Optional) Display summary APS information.
- interface *interface-name*—(Optional) Display APS information for the named interface.
- group *group*—(Optional) Display APS information for the named group.
- Required Privilege Level** view
- Sample Output** Sample Output: show aps (standard) on page 305  
 Sample Output: show aps brief on page 305  
 Sample Output: show aps detail on page 305  
 Sample Output: show aps extensive on page 305  
 Sample Output: show aps summary on page 306  
 Sample Output: show aps group on page 306  
 Sample Output: show aps interface on page 306
- Options at a Glance** Table 41 summarizes the information included in the output of each show aps command option. In this table, output fields are listed in alphabetical order. The Output Fields section lists the output fields in the order in which they are displayed.

**Table 41: Show APS Output Field Summary (Alphabetical Order)**

Options	Field Description
Detail Extensive	adj—State of the neighbor adjacency: Up, Down, Init, or Unknown.
Detail Extensive	Channel state—Displays which circuit has been selected: Working or Protect.
All	Circuit—Circuit type: Working or Protect.
Detail Extensive	dead—Number of seconds before neighbor is declared dead.0

Options	Field Description
All	Group—Group name.
Extensive	Hello due—Time next hello packet will be sent.
All	Interface—Name of interface.
All	Intf State—State of the interface: Disabled or Enabled—and the circuit : Up, Degraded, Down, Unknown, or Non-existent.
Extensive	nbr K1— Displays the value of the SONET/SDH K1 byte requested to be transmitted by the neighbor.
Extensive	nbr paired req—Nonzero if the neighbor is requesting a particular K1 value due to a change in the paired circuit.
Extensive	neighbor revert time—Configured time, on the neighbor interface, to wait after the working circuit has again become functional before making the working circuit active again.
Detail Extensive	Neighbor—Address and state of neighbor interface. If the working and protect interfaces are on the same router, the neighbor address is displayed as 0.0.0.0.
Detail Extensive	Protect circuit is on—When both the working circuit and protect circuit are on the same router, displays the interface name of APS protect circuit.
Extensive	rcv K1—Displays the value of the SONET/SDH K1 byte received on this interface. (Valid only on the protect circuit.)
Extensive	Req K1—Displays the value of the SONET/SDH K1 byte requested to be transmitted by this circuit.
Extensive	Revert time—Configured time to wait after the working circuit has again become functional before making the working circuit active again.
Detail Extensive	Working circuit is on—When both the working circuit and protect circuit are on the same router, displays the interface name of APS working circuit.
Extensive	xmit K1—Displays the value of the SONET/SDH K1 byte being transmitted on this interface. (Valid only on the protect circuit.)

**Output Fields** Interface—Name of interface.

Group—Group name.

Circuit—Circuit type: Working or Protect.

Intf State—State of the interface and circuit:

Interface: Disabled or Enabled

Circuit: Up, Degraded, Down, Unknown, or Non-existent

Neighbor—(Detail and extensive output only) Address and state of neighbor interface. If the working and protect interfaces are on the same router, the neighbor address is displayed as 0.0.0.0.

adj—(Detail and extensive output only) State of the neighbor adjacency: Up, Down, Init, or Unknown.

dead—(Detail and extensive output only) Number of seconds before neighbor is declared dead.

Channel state—(Detail and extensive output only) Circuit that has been selected: Working or Protect.

Local mode—(Extensive output only) Mode in which the local router is configured to interoperate with SONET line-terminating equipment (LTE): unidirectional or bidirectional. The parenthetical value represents the mode type in the K2 byte.

Neighbor mode—(Extensive output only) Mode in which the neighboring device is operating: unidirectional or bidirectional. The parenthetical value represents the mode type in the K2 byte.

Protect circuit is on—(Detail and extensive output only) When both the working circuit and protect circuit are on the same router, displays the interface name of APS protect circuit.

Working circuit is on—(Detail and extensive output only) When both the working circuit and protect circuit are on the same router, displays the interface name of APS working circuit.

Req K1—(Extensive output only) Display the value of the SONET/SDH K1 byte requested to be transmitted by this circuit.

rcv K1—(Extensive output only) Display the value of the SONET/SDH K1 byte received on this interface. (Valid only on the protect circuit.)

xmit K1—(Extensive output only) Display the value of the SONET/SDH K1 byte being transmitted on this interface. (Valid only on the protect circuit.)

nbr K1—(Extensive output only) Display the value of the SONET/SDH K1 byte requested to be transmitted by the neighbor.

nbr paired req—(Extensive output only) Nonzero if the neighbor is requesting a particular K1 value due to a change in the paired circuit.

Revert time—(Extensive output only) Configured time to wait after the working circuit has again become functional before making the working circuit active again.

neighbor revert time—(Extensive output only) Configured time, on the neighbor interface, to wait after the working circuit has again become functional before making the working circuit active again.

Hello due—(Extensive output only) Time the next hello packet will be sent.

**Sample Output: show  
aps (standard)**

```
user@host> show aps
Interface Group          Circuit Intf State
t3-1/2/0:0 king         Working enabled, up
t3-1/3/0:0 king         Protect disabled, up
```

**Sample Output: show  
aps brief**

```
user@host> show aps brief
Interface Group          Circuit Intf State
t3-1/2/0:0 king         Working enabled, up
t3-1/3/0:0 king         Protect disabled, up
```

**Sample Output: show  
aps detail**

```
user@host> show aps detail
Interface Group          Circuit Intf State
t3-1/2/0:0 king         Working enabled, up
Neighbor 0.0.0.0, adj up, neighbor interface disabled, dead 2.670
Channel state Working
Protect circuit is on interface t3-1/3/0:0
```

```
t3-1/3/0:0 king          Protect disabled, up
Neighbor 0.0.0.0, adj up, neighbor interface enabled, dead 2.750
Channel state Working
Working circuit is on interface t3-1/2/0:0
```

**Sample Output: show  
aps extensive**

```
user@host> show aps extensive
Interface Group          Circuit Intf State
t3-1/2/0:0 king          Working enabled, up
Neighbor 0.0.0.0, adj up, neighbor interface disabled, dead 2.100
Channel state Working
  local-mode bidirectional(5), neighbor-mode bidirectional(5)
Protect circuit is on interface t3-1/3/0:0
Req K1 0x00, rcv K1 0x00, xmit K1 0x00, nbr K1 0x11, nbr paired req 0
Revert time 0, neighbor revert time 0
Hello due in 0.258
t3-1/3/0:0 king          Protect disabled, up
Neighbor 0.0.0.0, adj up, neighbor interface enabled, dead 2.400
Channel state Working
Working circuit is on interface t3-1/2/0:0
Req K1 0x11, rcv K1 0x00, xmit K1 0x00, nbr K1 0x00, nbr paired req 0
Revert time 0, neighbor revert time 0
Hello due in 0.076
```

**Sample Output: show  
aps summary**

```
user@host show aps summary
Interface Group          Circuit Intf State
t3-1/2/0:0 king          Working enabled, up
t3-1/3/0:0 king          Protect disabled, u
```

**Sample Output: show  
aps group**

```
user@host> show aps group king
Interface Group          Circuit Intf State
t3-1/3/0:0 king          Protect disabled, up
Neighbor 0.0.0.0, adj up, neighbor interface enabled, dead 2.900
Channel state Working
Working circuit is on interface t3-1/2/0:0
Req K1 0x11, rcv K1 0x00, xmit K1 0x00, nbr K1 0x00, nbr paired req 0
Revert time 0, neighbor revert time 0
Hello due in 0.633
t3-1/2/0:0 king          Working enabled, up
Neighbor 0.0.0.0, adj up, neighbor interface disabled, dead 2.860
Channel state Working
Protect circuit is on interface t3-1/3/0:0
Req K1 0x00, rcv K1 0x00, xmit K1 0x00, nbr K1 0x11, nbr paired req 0
Revert time 0, neighbor revert time 0
Hello due in 0.836
```

**Sample Output: show  
aps interface**

```
user@host> show aps interface t3-1/3/0:0
Interface Group          Circuit Intf State
t3-1/3/0:0 king          Protect disabled, up
Neighbor 0.0.0.0, adj up, neighbor interface enabled, dead 2.230
Channel state Working
Working circuit is on interface t3-1/2/0:0
Req K1 0x11, rcv K1 0x00, xmit K1 0x00, nbr K1 0x00, nbr paired req 0
Revert time 0, neighbor revert time 0
Hello due in 0.491
```

