Airline Product Set with MATIP

Feature Overview

The Airline Product Set (ALPS) feature transports airline protocol data across a TCP/IP network to a mainframe. ALPS provides connectivity between agent set control units (ASCUs) and a host airline reservation systems.

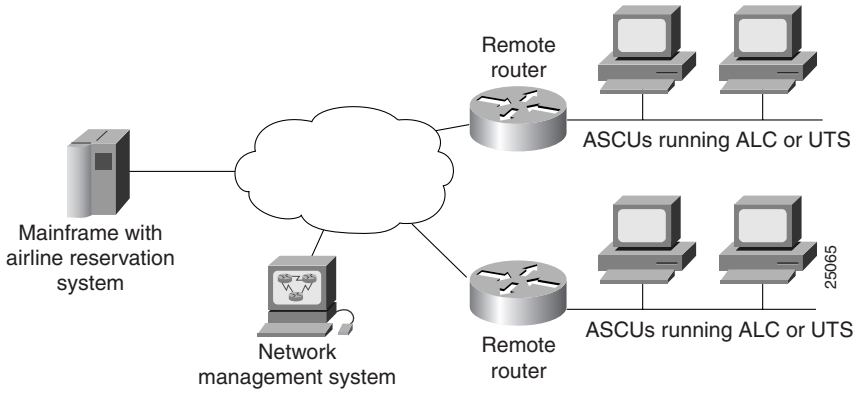
The first two phases of ALPS began the network migration to TCP/IP without requiring any changes in the hardware or software of the endstations (ASCUs and mainframes). ALPS phase I and II developed a new protocol, ALPS Tunneling Protocol (ATP), to tunnel airline protocol traffic (ALC or UTS data) through the TCP/IP network between peer Cisco routers. ALPS phase I provided support for the ALC protocol and the transport of the data from the ASCUs to a reservations system on an IBM mainframe. ALPS phase II provided support for the UTS protocol and the transport of the data from the ASCUs to a reservations system on a Unisys host system. This feature module can be used in conjunction with the Cisco IOS Release 12.0(2)T Airline Product Set feature module.

ALPS phase III, ALPS with MATIP, provides support for Mapping of Airline Traffic over Internet Protocol (MATIP). MATIP is an industry standard protocol for transporting airline protocol traffic across a TCP/IP network. This feature enables the end-to-end delivery of ALC and UTS data streams between a Cisco router and the mainframe using TCP/IP. This feature removes the X.25 (AX.25 or EMTOX) requirements for communication with the host reservation system by enabling TCP/IP communication between the router and the airline host reservation system.

This document includes the following sections:

Figure 1 shows the basic ALPS topology and the protocols implemented in the feature. Three major components provide the end-to-end transportation of airline protocol traffic across the network: the P1024B Airline Control (ALC) or P1024C (UTS) protocol, the TCP/IP-based MATIP protocol conversion, and the TCP/IP access to the mainframe.

Figure 1 Airline Product Set with MATIP Architecture



Benefits

- Provides an end-to-end solution for airlines or central reservation systems.
- Allows airlines to replace their existing hardware and software with Cisco routers because the ALPS feature is integrated in the Cisco IOS software. For customers who already use Cisco routers, this feature allows them to consolidate networking overhead and functionality.
- Enables the end-to-end delivery of ALC and UTS data between a remote router or gateway and the mainframe using TCP/IP encapsulation.
- Eliminates network overhead for error detection and transmission logic associated with X.25 links.
- Replaces IBM front-end processors (FEPs) with Channel Interface Processors (CIPs).
- Eliminates the use of dedicated, leased, slow-speed ALC and UTS serial lines and migrates the reservation system networks to a modern networking paradigm.Once the mainframe reservation system is enabled to use TCP/IP, new applications can be written for PCs or NCs.
- Supports standards-based MATIP protocol for transporting data across the TCP/IP network.

Restrictions

The ALPS with MATIP feature supports only type A conversational protocol traffic. The ALPS with MATIP feature does not support MATIP type A host-to-host protocol traffic and MATIP type B messaging protocol traffic.

Remote routers must have the Cirrus Logic CD2430 chipset on a synchronous serial interface module to connect to the ALC or UTS ASCUs. See the “Supported Platforms” section for more information.

Related Features and Technologies

The ALPS MATIP feature is an enhancement to the existing ALPS features that are documented in the "Airline Product Set" chapters of the *Cisco IOS Release 12.0 Bridging and IBM Networking Configuration Guide* and the *Cisco IOS Release 12.0 Bridging and IBM Networking Command Reference* and the *Airline Product Set* feature module for Cisco IOS Release 12.0(2)T.

Related Documents

- Cisco IOS Release 12.0 *Bridging and IBM Networking Configuration Guide*
- Cisco IOS Release 12.0 *Bridging and IBM Networking Command Reference*
- Cisco IOS Release 12.0(2)T Airline Product Set feature module

Supported Platforms

The ALPS feature is supported on the following remote router platforms:

- Cisco 2520, 2521, 2522, and 2523
- Cisco 3600 series
- Cisco 4500
- Cisco 4700

Note The Cisco 4500 and Cisco 4700 platforms must have a high-density, low-speed serial card installed. Sixteen low-speed ports are available for performing the remote router functions.

Supported MIBs, RFCs, and Standards

MIBs

No new or modified MIBs are supported by this feature.

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

- RFC 2351, *Mapping of Airline Reservation, Ticketing, and Messaging Traffic over IP*, May 1998

Standards

- *P1024B Communication Control Protocol Specification*, Society International Telecommunications Aeronautics
- *P1024C Communication Control Protocol Specification*, Society International Telecommunications Aeronautics
- *MATIP Implementation Guide*, Society International Telecommunications Aeronautics

Prerequisites

The ALPS with MATIP feature is available on the Cisco IOS software IP Plus images.

Configuration Tasks

The following sections provide configuration tasks for the Airline Product Set with MATIP feature. Each task in the list indicates if the task is optional or required.

- Configuring the Remote Routers (Required)
- Customizing the Service Messages (Optional)
- Updating a Circuit (Optional)
- Verifying ALPS (Required)

Configuring the Remote Routers

Perform the tasks in the following sections to configure the ALPS feature on the remote routers:

- Specifying the ALPS Local Peer IP Address
- Specifying the ALPS Remote Peer IP Address
- Specifying the ALPS Circuit
- Specifying Each ASCU

Specifying the ALPS Local Peer IP Address

You must identify an IP address as an ALPS local peer on the remote router. Only one ALPS local peer is permitted on a router.

To specify the ALPS local peer IP address, use the following commands in global configuration mode:

Step	Command	Purpose
1	**alps local-peer** *ipaddress*	Specifies an IP address to use as the ALPS local peer on the remote router.
2	**alps keepalive** [**interval** *time*] [**retry** *count*]	Enables TCP keepalives for ALPS TCP peer connections.

Specifying the ALPS Remote Peer IP Address

You must specify a partner IP address (remote peer) on the remote router. The peer connection may be permanent or dynamic (established on demand).

To specify the partner IP address for one or more TCP peer connections to the configured IP address, use the following command in global configuration mode:

Command	Purpose
alps remote-peer *ip-addr* [**protocol** {*atp* | *matip-a*}] [**status-interval** *interval*] [**status-retry** *retries*] [**dynamic** [*inact-timer*] [**no-circuit** *no-circ-timer*]] [**tcp-qlen** [*num*]]	Specifies the partner IP address.

Specifying the ALPS Circuit

An ALPS circuit is a communication path across a TCP connection for one or more ASCUs. The ALPS circuit must have a configured association with an ALPS remote peer to establish a connection to the host. Additionally, an ALPS circuit configuration may specify a different remote peer as a backup peer to the host. Each MATIP circuit maps to a single TCP connection. For ATP, ALPS circuits can be multiplexed across to a single TCP connection.

To specify an ALPS circuit, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	**alps circuit** *name*	Specifies an ALPS circuit at the remote router and enters ALPS circuit submode.
2	**alps primary-peer** *ip-addr* [**backup-peer** *ip-addr*]	Specifies the primary TCP peer and an optional backup peer for this ALPS circuit.
3	**alps local-hld** *loc-hld* **remote-hld** *rem-hld*	Specifies the local high-level designator (HLD) for this ALPS circuit.
4	**alps connection-type permanent** *retry-timer*	(Optional) Specifies that this circuit should be established when the circuit is enabled.
5	**alps lifetime-timer** *timer*	(Optional) Specifies how long messages can be queued in the ALPS circuit queue.
6	**alps service-msg-interval** *seconds*	(Optional) Specifies the interval between the transmission of a service message to an ASCU and the transmission of a PLEASE RETRY message. The PLEASE RETRY message is transmitted only to ASCUs that use circuits with a dynamic connection type.
7	**alps service-msg-list** *list*	(Optional) Defines the service message list to be used for this circuit.
8	**alps matip-close-delay** *time*	(Optional) Specifies the interval between the closing and reopening of the MATIP circuit connection.
9	**alps enable-circuit**	Enables the circuit.

Specifying Each ASCU

You must configure each ASCU within the context of the serial interface configuration. You must configure ASCU addressing information and association with an ASCU. You can configure the timers, maximum frame sizes, and retry values optional configuration parameters for each ASCU. Appropriate default parameters are used for unspecified parameters. Once you configure the first ASCU, you can configure additional ASCUs using only Steps 8 through 14.

To specify an ASCU, use the following commands, beginning in global configuration mode:

Step	Command	Purpose
1	**interface** *type number*	Configures an interface and enter interface configuration mode.
2	**encapsulation** [**alc** | **uts**]	Specifies the protocol to be used on the serial interface.
3	**alps t1** *delay*	(Optional) Specifies the timeout delay between the transmission of an ALC poll message and the receipt of the first character of the poll message response.
4	**alps t2** *delay*	(Optional) Specifies the timeout delay between receipt of the first character of the response to a poll message and the receipt of a Go Ahead message.

Step	Command	Purpose
5	**alps n1** *errors*	(Optional) Specifies the threshold of consecutive errors logged before an ASCU is declared down.
6	**alps n2** *polls*	(Optional) Specifies the number of polls that must be correctly replied to before an ASCU is declared up.
7	**alps servlim** *polls*	(Optional) Specifies the number of polls of the ASCU UP list allowed between two successive polls of the ASCU DOWN list.
8	**alps ascu** *id*	Specifies a physical ASCU identity (the ASCU interchange address value for ALC) and enter ALPS ASCU submode.
9	**alps default-ckt** *name*	Specifies the ALPS circuit that this ASCU uses.
10	**alps a1-map** *a1-value* **a2-map** *a2-value*	Specifies the A1 and A2 logical ASCU identification information.
11	**alps retry-option** [**resend** | **reenter**]	(Optional) Specifies the retry option when an ALC message with a bad cyclic check character (CCC) is received.
12	**alps max-msg-length** *value*	(Optional) Specifies maximum input message length.
13	**alps error-display** *number1 number2*	(Optional) Specifies where error messages are displayed.
14	**alps enable-ascu**	Begins polling the ASCU.

Customizing the Service Messages

You can customize the contents of the service message list. To specify the service message number and the content of the message, use the following command in global configuration mode:

Command	Purpose
alps service-msg-list *list* **number** *number msg*	Specifies service message numbers and content.

Updating a Circuit

You can clear or update the circuits on the ALPS network. If a specific name is entered, the update action will be executed only on a configured circuit with that name; otherwise, the action will be performed on all configured circuits. If the circuit uses the ATP protocol, an update consists of a closing and re-opening of the ALPS circuit (the same action performed when clearing the circuit). If the circuit is a MATIP circuit, the update results in the sending of a configuration update (in the form of a MATIP Session Open command). You can update the circuit only on enabled or active (opened or opening state) ALPS circuits.

To update one or more ALPS circuits, use the following command in EXEC mode:

Command	Purpose
alps update-circuit [*name*]	Specifies name of circuit to update.

Verifying ALPS

Perform the tasks in the following steps to verify the ALPS feature:

Step 1 Verify that the connection between the router and the ASCU is up by polling the ASCU. Enter the **show alps ascu** command and check the state field. UP indicates that the ASCU is responding to the polling. DOWN indicates that the connection is not responding to the polling.

```
router#show alps ascu
interface    dlc   id   a1   a2   circuit    pkt_tx  pkt_rx  state
-------------------------------------------------------------------------------
Serial6      ALC   42   60   70   CKT_ALC_1  416     416     UP
Serial6      ALC   45   60   72   CKT_ALC_1  600     600     UP
Serial6      ALC   48   62   78   CKT_ALC_2  0       0       DOWN
Serial7      UTS   21   22   13   CKT_UTS    4830    4830    UP
```

Step 2 Verify that the peer between the router and the host is connected. Enter the **show alps peer** command and check the state field. OPENED indicates that the circuit is connected. DISCONN indicates that the circuit is disconnected.

```
router#show alps peers
local_peer : ip_address = 192.168.25.2

ip_address     conn_id             state    pkt_t  pkt_rx
-------------------------------------------------------------------------------
192.168.20.3   MATIP_A_CKT_UTS     OPENED   1023   1023
192.168.70.2   MATIP_A_CKT_ALC_1   OPENED   4852   4757
192.168.70.2   MATIP_A_CKT_ALC_2   OPENED   1      1
192.168.70.3   MATIP_A_CKT_ALC_1   DISCONN  0      0
192.168.70.3   MATIP_A_CKT_ALC_2   DISCONN  0      0
```

Monitoring and Maintaining ALPS

To monitor the status of the ALPS feature, use the following commands in EXEC mode:

Command	Purpose
show alps ascu [*interface*] [*id*] [**detail**]	Displays the status of the ALPS ASCU.
show alps circuits [**name** *name*] [**detail**]	Displays the status of the ALPS circuits.
show alps peers [**ipaddress** *addr*] [**detail**]	Displays the status of the ALPS remote peers.

Configuration Examples

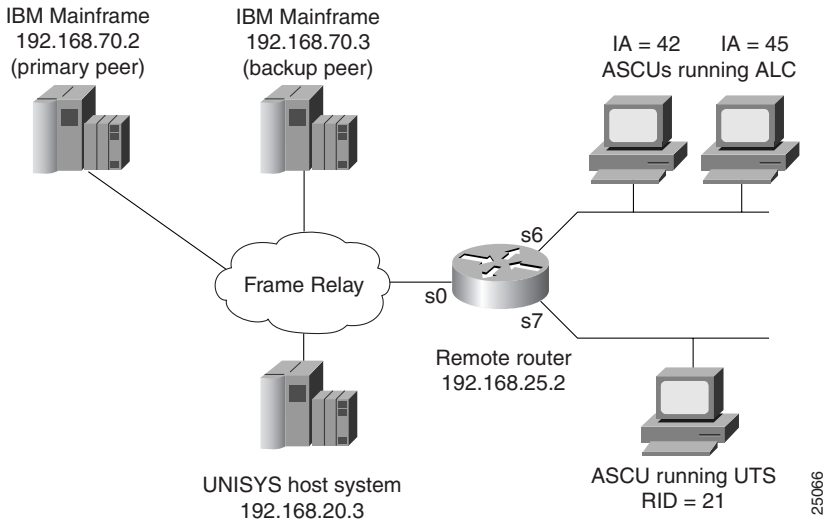
This section provides the following configuration example:

- ALPS with MATIP Configuration for ALC and UTS

ALPS with MATIP Configuration for ALC and UTS

Figure 2 shows a simple example of a router topology for the ALPS with MATIP feature. The configuration corresponding to this topology follows.

Figure 2 Router Topology for the ALPS with MATIP Configuration Example



ALC/UTS Router Configuration

```
(config)# hostname alps-rcpe
(config)# alps local-peer 192.168.25.2
(config)# alps keepalive interval 45 retry 2
(config)# alps remote-peer 192.168.20.3 protocol matip-a dynamic status-interval 60
(config)# alps remote-peer 192.168.70.2 protocol matip-a dynamic 0 no-circuit 10
(config)# alps remote-peer 192.168.70.3 protocol matip-a dynamic 45
(config)# alps enable-alarms peer 192.168.70.2
(config)# alps enable-alarms ascu
!
(config)# alps circuit CKT_ALC_1
(config-alps-circ)# alps primary-peer 192.168.70.2 backup-peer 192.168.70.3
(config-alps-circ)# alps connection-type permanent
(config-alps-circ)# alps local-hld 2525
(config-alps-circ)# alps enable-circuit

(config)# alps circuit CKT_ALC_2
(config-alps-circ)# alps primary-peer 192.168.70.2 backup-peer 192.168.70.3
(config-alps-circ)# alps mpx single hdr none
(config-alps-circ)# alps local-hld 2526
(config-alps-circ)# alps enable-circuit
!
(config)# alps circuit CKT_UTS
(config-alps-circ)# alps primary-peer 192.168.20.3
(config-alps-circ)# alps mpx single
(config-alps-circ)# alps idle-timer 90
(config-alps-circ)# alps local-hld 2527
(config-alps-circ)# alps enable-circuit
(config-alps-circ)# alps service-msg-interval 2
!
(config)# interface Loopback0
(config-if)# ip address 192.168.25.2 255.255.255.0

(config)# interface Serial0
(config-if)# ip address 210.100.50.2 255.255.255.0
```

```
(config-if)# encapsulation frame-relay IETF
(config-if)# frame-relay map ip 210.100.60.2 40
(config-if)# frame-relay map ip 210.100.70.2 50

!

(config)# interface Serial6
(config-if)# encapsulation alc
(config-if)# alps t1 6
(config-if)# alps t2 8
(config-if)# alps poll-pause 100
(config-if)# clockrate 9600
!
(config-if)# alps ascu 42
(config-alps-ascu)# alps default-circuit CKT_ALC_1
(config-alps-ascu)# alps a1-map 60 a2-map 70
(config-alps-ascu)# alps enable-ascu
!
(config-if)# alps ascu 45
(config-alps-ascu)# alps default-circuit CKT_ALC_1
(config-alps-ascu)# alps a1-map 60 a2-map 72
(config-alps-ascu)# alps enable-ascu
!
(config)# interface Serial7
(config-if)# encapsulation uts
(config-if)# alps n3 4
(config-if)# alps poll-pause 125
(config-if)# clockrate 4800
!
(config-if)# alps ascu 21
(config-alps-ascu)# alps default-circuit CKT_UTS
(config-alps-ascu)# alps a1-map 22 a2-map 13
(config-alps-ascu)# alps enable-ascu
!
```

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publication.

- **alps a1-map a2-map**
- **alps local-hld remote-hld**
- **alps matip-close-delay**
- **alps remote-peer**
- **alps update-circuit**
- **show alps ascu**
- **show alps circuits**
- **show alps peers**

In Cisco IOS Release 12.0(1)T or later, you can search and filter the output for **show** and **more** commands. This functionality is useful when you need to sort through large amounts of output, or if you want to exclude output that you do not need to see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search or filter on:

command **|** {**begin | include | exclude**} *regular-expression*

Following is an example of the **show atm vc** command in which you want the command output to begin with the first line where the expression "PeakRate" appears:

show atm vc | begin PeakRate

For more information on the search and filter functionality, refer to the Cisco IOS Release 12.0(1)T feature module titled *CLI String Search*.

alps a1-map a2-map

To specify the A1 and A2 logical ASCU identification information, use the **alps a1-map a2-map** ALPS ASCU submode command. To remove the specification of the A1 and A2 logical ASCU identification information, use the **no** form of this command.

alps a1-map *a1-value* **a2-map** *a2-value*

no alps a1-map *a1-value* **a2-map** *a2-value*

Syntax Description

a1-value	A1 logical ASCU identification: • ALC range—Hexadecimal number in the range 0 to 0xFF. • UTS range—Hexadecimal number in the range 0 to 0xFF.
a2-value	A2 logical ASCU identification: • ALC range—Hexadecimal number in the range 0 to 0xFF. • UTS range—Hexadecimal number in the range 0 to 0xFF.

Defaults

No default behavior or values.

Command Modes

ALPS ASCU submode

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	The range was modified.

Example

The following example specifies the A1 identification as 0x4C and the A2 identification as 0x20:

```
alps a1-map 4C a2-map 20
```

Related Commands

Command	Description
show alps ascu	Displays the status of the ALPS ASCU.

alps local-hld remote-hld

To specify the local and remote HLDs to use for this ALPS circuit, use the **alps local-hld remote-hld** ALPS circuit submode command. To remove the definition from the configuration, use the **no** form of this command.

alps local-hld *loc-hld* **remote-hld** *rem-hld*

no alps local-hld *loc-hld* **remote-hld** *rem-hld*

Syntax Description

loc-hld	Local HLD to use for ALPS circuit. Hexadecimal number in the range 1 to FFFF.
rem-hld	Remote HLD to use for ALPS circuit. Hexadecimal number in the range 1 to FFFF.

Defaults

No default behavior or values.

Command Modes

ALPS circuit submode

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	This command was modified. The **remote-hld** keyword is not applicable for MATIP.

Usage Guidelines

The **remote-hld** keyword is not applicable for ALPS with MATIP.

Examples

The following example specifies the local HLD as 4B10:

```
alps local-hld 4B10
```

Related Commands

Command	Description
alps circuit	Specifies an ALPS circuit at the remote router and enters ALPS circuit submode.
show alps circuits	Displays the status of the ALPS circuits.

alps matip-close-delay

To specify the interval between the closing and re-opening of MATIP circuit connections, use the **alps matip-close-delay** ALPS circuit submode command. To restore the definition to the default value, use the **no** form of this command.

alps matip-close-delay *time*

no alps matip-close-delay *time*

Syntax Description

time	Minimum number of seconds between the closing and re-opening of an ALPS MATIP circuit. The range is 1 to 90 seconds. The default is 10 seconds.

Defaults

The default value is 10 seconds.

Command Modes

ALPS circuit submode

Command History

Release	Modification
12.0(5)T	This command was introduced.

Examples

The following example specifies a close delay time of 20 seconds

```
alps matip-close-delay 20
```

Related Commands

Command	Description
show alps circuits	Displays the status of the ALPS circuits.

alps remote-peer

To specify the partner IP address, use the **alps remote-peer** global configuration command. To remove the definition from the configuration, use the **no** form of this command.

alps remote-peer *ip-addr* [**protocol** {*atp* | *matip-a*}] [**status-interval** *interval*] [**status-retry** *retries*] [**dynamic** [*inact-timer*] [**no-circuit** *no-circ-timer*]] [**tcp-qlen** [*num*]]

no alps remote-peer *ip-addr* [**protocol** {*atp* | *matip-a*}] [**status-interval** *interval*] [**status-retry** *retries*] [**dynamic** [*inact-timer*] [**no-circuit** *no-circ-timer*]] [**tcp-qlen** [*num*]]

Syntax Description

ip-addr	IP address of the peer.
protocol	(Optional) Specifies the type of encapsulation for the connection.
atp	ALPS Tunneling Protocol encapsulation. This encapsulation is the default.
matip-a	MATIP Type A (conversational) encapsulation.
status-interval	(Optional) Specifies amount of time, in seconds, between sending of MATIP status messages. The messages verify the integrity of the TCP connection.
interval	Number of seconds between status messages. The range is 0 to 300 seconds. The default value is 0 (off).
status-retry	(Optional) Specifies number of times to retry sending a MATIP status message before the peer connection is closed.
retries	Number of retries. The range is 0 to 100 retries. The default value is 2.
dynamic	(Optional) Allows the TCP connection to the host peer to be opened only when there is data to be transferred to the host reservation system.
inact-timer	Length of inactivity, in seconds, after which the connection is closed. The range is 0 to 300 seconds. The default is 30 seconds. A value of zero indicates that the timer is disabled.
no-circuit	(Optional) Specifies amount of time, in seconds, that a peer will stay connected while no circuits are using the peer connection. This parameter is valid only if the dynamic parameter is first configured.
no-circ-timer	Number of seconds before which the timer will expire. The range is 0 to 3600 seconds. The default is 90 seconds.
tcp-qlen	(Optional) Specifies the maximum length of a TCP queue for peer connections.

num	Number of packets allowed in the TCP queue. The range is 26 to 100 packets. The default is 50 packets.

Defaults

The default for the **dynamic** argument is 30 seconds.

The default for the **no-circuit** argument is 90 seconds.

The default for the **tcp-qlen** argument is 50 packets.

The default for the **status-interval** argument is 0 (off).

The default for the **status-retry** argument is 2.

Command Modes

Global configuration mode

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	The **protocol**, **status-interval**, **status-retry** keywords and the **no-circuit** option were added.

Usage Guidelines

When the protocol option is configured for MATIP, the peer connection is dynamic.

When the protocol option is configured for ATP, the peer connection is permanent.

The **no-circuit** option within the dynamic keyword does not apply to permanent (ALC/UTS) connections.

The **status-interval** and **status-retry** options apply only to the MATIP protocol.

Issuing the **no alps remote-peer** command does the following:

- Closes TCP connection.
- Notifies the partner TCP peer that this connection is closed.
- Notifies the ALPS circuits using this TCP peer that the connection is closed.

Examples

The following example specifies a MATIP peer connection at IP address 172.22.0.92. Status messages will be sent every 20 seconds and will be 2 times before the connection is closed. The maximum TCP length is 30:

```
alps remote-peer 172.22.0.92 protocol matip-a status-interval 20 status-retry 2
tcp-qlen 30
```

Related Commands

Command	Description
show alps peers	Displays the status of the ALPS remote peers.

alps update-circuit

To update one or more ALPS circuits, use the **alps update-circuit** EXEC command. If a circuit name is specified, then only that circuit will be updated; otherwise, all circuits will be updated.

alps update-circuit [*name*]

Syntax Description

name	Specifies name of circuit to update.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

If the **alps update-circuit** command is issued for a circuit that is using the ATP protocol, the circuit will be closed and reopened.

If the **alps update-circuit** command is issued for a circuit that is using the MATIP protocol, a configuration update will be sent in the form of a MATIP Session Open command.

The **alps update-circuit** command is effective only for ALPS circuits that are enabled and active (opening or opened state).

Examples

The following example specifies that circuit 1 has been updated:

```
alps update-circuit CKT-1
```

Related Commands

Command	Description
alps circuit	Specifies an ALPS circuit at the remote router and enters ALPS circuit submode.
alps enable-circuit	Specifies the circuit to be activated when data is received from an ASCU.
show alps circuits	Displays status of the ALPS circuits.

show alps ascu

To display the status of the ALPS ASCU, use the **show alps ascu** EXEC command.

show alps ascu [*interface* [*ia*]] [**detail**]

Syntax Description

interface [*ia*]	(Optional) Combined interface and ASCU Interchange Address (IA). If the interface and ASCU are specified, the status for only the ASCU on that interface is displayed. If only the interface is specified, all ASCUs defined on that interface are displayed. If the interface and ASCU are not specified, then all ASCUs defined are displayed.
detail	(Optional) Displays detailed output.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	This command was modified.

Examples

The following example shows output from the **show alps ascu** command:

```
router#show alps ascu
interface  dlc id a1 a2 circuit        pkt_tx      pkt_rx      state
-------------------------------------------------------------------------
Serial3    ALC 41 41 41 CKT-1          0           0           UP
Serial3    ALC 42 41 42 CKT-1          0           0           UP
Serial3    ALC 45 41 45 CKT-1          29          1484        UP
```

Related Commands

Command	Description
alps ascu	Specifies a physical ASCU identity.

show alps circuits

To display the status of the ALPS circuits, use the **show alps circuits** EXEC command. If a circuit name is specified, then only the status of that circuit will be displayed; otherwise, the status of all circuits will be displayed.

show alps circuits [**name** *name*] [**detail**]

Syntax Description

name *name*	Displays only the status of that circuit.
detail	(Optional) Displays detailed output.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	The output of this command was modified.

Examples

The following example shows output from the **show alps circuits** command:

```
router#show alps circuits name ckt-1 detail
CKT-1:dlc = ALC, conn_type = DYN, state = OPEN, uptime = 00:00:06
  down reason = noReason
  pri_peer = 10.227.50.106, sec_peer = 0.0.0.0
  curr_peer = 10.227.50.106, MATIP_A_CKT-1
  local_hld = 7F7F, remote_hld = 7F7F
  emtox:hostlink = 255, x121 = 1133470125
  lifetime_tmr = 4, idle_tmr = 60, retry_tmr = 0
  pkt_tx = 2239, byte_tx = 94244, pkt_rx = 2238, byte_rx = 94208
  src_corr = 0, dst_corr = 0
  drops_q_overflow = 388, drops_ckt_disabled = 30
  drops_lifetime_tmr = 24, drops_invalid_ascu = 0
  ascus:(45,41)
```

Related Commands

Command	Description
show alps peers	Displays the status of the ALPS remote peers.

show alps peers

Use the **show alps peers** EXEC command to display the status of the ALPS partner peers. If an IP address is specified, then only the status of that peer will be displayed; otherwise, the status of all peers will be displayed.

show alps peers [**ipaddress** *addr*] [**detail**]

Syntax Description

ipaddress *addr*	Displays only the status of that ASCU.
detail	(Optional) Displays detailed output.

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	This output of this command was modified.

Examples

The following example shows output from the **show alps peers detail** command:

```
router#show alps peers detail

TCP:10.227.50.106, conn_id = MATIP_A_CKT-2
  protocol = MATIP_A, fport = 350, lport = 11592
  type = DYN, create = ADMIN, state = OPENED, uptime = 00:00:53
  down reason = unknown
  pkt_tx = 1071, byte_tx = 37264, pkt_rx = 1066, byte_rx = 36010
  Drops:giants = 0, q_overflow = 0, peer_down = 0, ver_mismatch = 0
  active_ckts:CKT-2
```

Related Commands

Command	Description
alps primary-peer	Specifies the primary TCP peer and an optional backup peer for this ALPS circuit.
alps remote-peer	Specifies the partner IP address.

Debug Commands

This section documents new or modified **debug** commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

- **debug alps ascu**
- **debug alps peer**
- **debug alps snmp**

debug alps ascu

To enable debugging for ALPS ASCUs, use the **debug alps ascu** EXEC command. To disable debugging, use the **no** form of this command.

[**no**] **debug alps ascu** {**event | packet | detail | all**} [*interface* [*ascu id*]]

Syntax Description

event	Display ASCU events or protocol errors.
packet	Display transmitted or received packets.
detail	Display all ASCU protocol events.
all	Enable event, packet, and detail debugging.
interface	(Optional) Enable debugging on a specified interface.
ascu id	(Optional) Enable debugging for a specified ASCU.

Defaults

Debugging is off.

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	This command was modified.

Usage Guidelines

To enable debugging for a group of ASCUs enter a separate command for each ASCU interface and IA combination.

Examples

The following output is from the **debug alps ascu event** command, showing events or protocol errors for ASCU 42 on interface Serial7:

```
router#debug alps ascu event Serial7 42
ALPS ASCU: T1 expired for ascu 42 on i/f Serial7
ALPS ASCU: DOWN event while UP for ascu 42 on i/f Serial7 : C1 count = 1
```

The following output is from the **debug alps ascu detail** command, showing all protocol events for ASCU 42 on interface Serial6:

```
router#debug alps ascu detail Serial6 42
ALPS ASCU: Tx ALC POLL MSG (3 bytes + CCC) to ascu 42 on i/f Serial6
ALPS ASCU: ALC GO AHD MSG rcvd from ascu 42 on i/f Serial6
ALPS ASCU: Tx ALC POLL MSG (3 bytes + CCC) to ascu 42 on i/f Serial6
ALPS ASCU: ALC GO AHD MSG rcvd from ascu 42 on i/f Serial6
```

```
ALPS ASCU: Tx ALC POLL MSG (3 bytes + CCC) to ascu 42 on i/f Serial6
ALPS ASCU: Rx ALC DATA MSG (14 bytes + CCC) from ascu 42 on i/f Serial6, fwd ckt
RTP_MATIP
ALPS ASCU: ALC GO AHD MSG rcvd from ascu 42 on i/f Serial6
ALPS ASCU: Tx ALC DATA MSG (14 bytes + CCC) to ascu 42 on i/f Serial6
ALPS ASCU: Tx ALC POLL MSG (3 bytes + CCC) to ascu 42 on i/f Serial6
```

The following output is from the **debug alps ascu packet** command, showing all packets transmitted or received for ASCU 42 on interface Serial6:

```
router#debug alps ascu packet Serial6 42
ALPS ASCU: Tx ALC SERVICE MSG (18 bytes + CCC) to ascu 42 on i/f Serial6
0405B530:                   02321D26 0C261616
0405B540: 140C0D18 26163135 0611C6
ALPS ASCU: Rx ALC DATA MSG (14 bytes + CCC) from ascu 42 on i/f Serial6, fwd ckt
RTP_MATIP
040730B0:                   42607866 65717866
040730C0: 65717966 755124
ALPS ASCU: Tx ALC DATA MSG (14 bytes + CCC) to ascu 42 on i/f Serial6
0405B540:                     022038 26253138
0405B550: 26253139 263511E4
```

debug alps peer

Use the **debug alps peer** EXEC command to enable event or packet debugging for ALPS peers. To disable debugging, use the **no** form of this command.

[**no**] **debug alps peer** {**event | packet**} [*ipaddr*]

Syntax Description

event	Specifies debugging for an event.
packet	Specifies debugging for a packet.
ipaddr	(Optional) Remote peer IP address.

Defaults

If no IP address is specified, then debugging is enabled for every peer connection.

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	The **packet** keyword was added. The format for the output was modified for consistency.

Usage Guidelines

To enable debugging for a single remote ALPS peer, specify the peer IP address.

To enable debugging for a set of remote peers, enter the command for each peer IP address.

Examples

The following output is from the **debug alps peer packet** command:

```
router#debug alps peer packet
ALPS PEER:Peer (10.227.50.106, MATIP_A_CKT-1) - TX Peer Data Msg (18 bytes)
040A5320:                        01 00001241
040A5330:45546B5F 6F4F7757 67477B5B 51
ALPS PEER:Peer (10.227.50.106, MATIP_A_CKT-1) - RX Peer Data Msg (18 bytes)
04000550:         01000012 4145546B 5F6F4F77
04000560:5767477B 5B51
ALPS PEER:Peer (10.227.50.106, MATIP_A_CKT-1) - TX Peer Data Msg (18 bytes)
0409F6E0:               01 00001241 45546B5F
0409F6F0:6F4F7757 67477B5B 51
ALPS PEER:Peer (10.227.50.106, MATIP_A_CKT-1) - RX Peer Data Msg (18 bytes)
04000680:                  01000012 4145546B
04000690:5F6F4F77 5767477B 5B51
```

debug alps snmp

Use the **debug alps snmp** EXEC command to enable debugging for ALPS SNMP agents. To disable debugging, use the **no** form of this command.

[**no**] **debug alps snmp**

Syntax Description

This command has no arguments or keywords.

Defaults

Debugging for SNMP agents is not enabled.

Command History

Release	Modification
11.3(6)T	This command was introduced for limited availability.
12.0(1)	This command was available for general release.
12.0(5)T	This command was added to the documentation.

Examples

The following output is from the **debug alps snmp** command. The first line shows a circuit event status change. The second line shows an ASCU status change. The third line shows a peer connection status change.

```
ALPS CktStatusChange Notification for circuit CKT-1
ALPS AscuStatusChange Notification for ascu (Serial3, 41)
PeerConnStatusChange Notification for peer (10.227.50.106, MATIP_A_CKT-1)
```

Glossary

agent set control unit—See ASCU.

Airline Control Protocol—See ALC.

Airline Product Set—See ALPS.

Airline protocol—A generic term that refers to the airline reservation system data and the protocols such as P1024B (ALC), P1024C (UTS), and MATIP that are used to transport the data between the mainframe and the ASCUs.

ALC—Airline Control Protocol. P1024B, a data link layer polled protocol that runs in full-duplex mode over synchronous serial (V.24) lines and uses the binary-coded decimal (BCD) character set.

ALPS—Cisco's Airline Product Set feature. The feature includes all the components involved in transporting airline protocol data across a TCP/IP based network to mainframe reservation systems.

ALPS circuit—Communication path across a TCP connection between a host reservation system and an ASCU.

ALPS Tunneling Protocol—See ATP.

ASCU—Agent set control unit. An airline reservations system terminal controller.

ASCU Interchange Address—See IA.

ATP—ALPS Tunneling Protocol. A protocol used to transport ALPS data across a TCP/IP network between an ALC/UTS router and an AX.25/EMTOX router. It consists of a set of messages (or primitives) to activate and deactivate ALPS ATP circuits and pass data.

High-Level Designator—See HLD.

HLD—High-Level Designator.

IA—ASCU Interchange Address. Specifies a physical ASCU identity.

Management Information Base—See MIB.

Mapping of Airline Traffic over Internet Protocol—See MATIP.

MATIP—Mapping of Airline Traffic over Internet Protocol. A standard defined in RFC 2351 for transporting airline reservation, ticketing, and messaging traffic over TCP/IP.

messaging traffic—See Type B traffic.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

remote router—Routers with the ALPS feature that are physically connected to the ASCUs.

Simple Network Management Protocol—See SNMP.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security.

transactional traffic—See Type A traffic.

Type A traffic—Transactional traffic. Typically, this is conversational traffic exchanged between a host and its ASCUs for terminal queries and responses for reservation systems. There is another form of type A traffic called host-to-host.

Type B traffic—Messaging traffic. Typically, this is e-mail application traffic in IATA-compliant format.

Universal Terminal Support—See UTS.

UTS—Universal Terminal Support. P1024C is a data link layer protocol that runs in full-duplex mode over synchronous serial (V.24) lines and uses the ASCII character set.