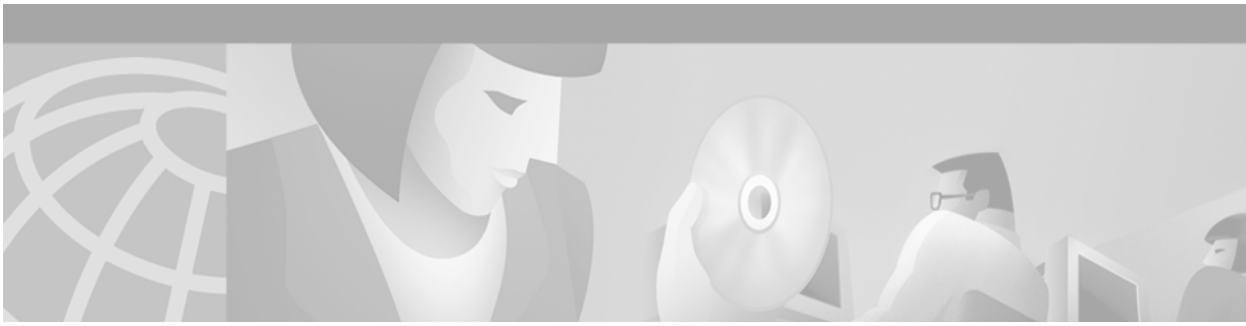

Input Translation Table Management Enhancements

This feature module describes the software enhancements that provide alternatives in managing the Input Translation Table (ITT) resource in the Cisco 6400 series Node Switch Processor (NSP) and includes the following sections:

Feature Overview

This software feature adds three enhancements that display and alter switch behavior in managing the ITT resource.

The ITT:

- Is a hardware data structure used in the Cisco 6400 series for handling the incoming cells.
- Consists of entries that for virtual circuit (VC) switching are allocated as a contiguous block, with each block dedicated to a virtual path identifier (VPI) on an interface.
  - Each entry specifies whether a virtual channel identifier (VCI) is valid within a VPI.
  - The allocated blocks must be a power of two in sizes such as 16, 32, 64, and so on.
- Is used only when both interfaces through which the VC transits are up.

The Node Switch Processor (NSP) has a single ITT, organized in two banks of 32K entries each. When a VC is created, a block of entries is allocated in the ITT for that VPI. The block size should be a power two that is greater than or equal to the VCI value. This limits the use of large VCI values and the distribution of VCIs on VPIs as the number of VCs approaches 32K. When an additional VC is added

to a VPI that requires a larger block size than the current block, the current block is copied to new larger block, and the original block is freed. This leaves a series of small blocks that are unused. ITT memory is fragmented because of this growing technique.

The Cisco 6400 series has three new features to:

- Minimize fragmentation
- Shrink ITT blocks
- View used and unused ITT blocks.

New configuration commands to minimize fragmentation enable the NSP to automatically determine the minimum ITT block size needed to support the PVCs configured for each interface and VPI. When an interface comes up, the **minblock** command specifies the ITT block size requested for a VPI on that interface.

By default, the ITT blocks grow as necessary to accommodate high VCI values for a given port and VPI, but ITT space is not returned unless the entire ITT block is free. A new command is introduced that reduces the size of an ITT block when a VC with a high numbered VCI is deleted.

A command to display details of the used and unused ITT blocks is also introduced. This gives a picture of the quantity and quality of ITT utilization at a given time. The output of the **show** command allows you to view details of the free blocks by size and bank, the aggregate free space left, and the location of in-use blocks.

Note The new configuration commands listed in this document require additional processing, which reduces call setup rates and increases memory usage when the commands are enabled.

Benefits

- Reduces fragmentation in ITT blocks
- Displays ITT allocation
- Autoshrinks ITT blocks

Restrictions

This feature is supported only on the Cisco IOS Release 12.1(4)DB and later releases on the Cisco 6400 series.

Related Features and Technologies

None

Related Documents

- *Cisco 6400 Software Configuration and Command Reference Guide*
- *Optimizing the Number of Virtual Connections on the Cisco 6400*

Supported Platforms

The ITT Management Enhancements feature is supported on the Cisco 6400 NSP.

Supported Standards, MIBs, and RFCs

None

Prerequisites

None

Configuration Tasks

See the following sections for Cisco 6400 NSP configuration tasks:

- Specifying a Minimum Block Size
- Enabling Automatic Determination of Minimum ITT Block Size
- Reclaiming ITT Space Dynamically

Specifying a Minimum Block Size

If you know the highest VCI that will be required for a particular interface and VPI combination, you can specify it as the minimum ITT block size for that combination. Specifying a minimum block size reduces fragmentation by preventing block expansion. Use the **force** keyword to ensure that the block size is not overidden by the autominblock analysis.

To specify a minimum block size for a particular interface and VPI, complete the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Switch(config)# interface atm slot/subslot/port`	Specifies an ATM interface.
Step 2	`Switch(config-if)# atm input-xlate-table minblock vpi vpi-value block-size force`	Specifies the minimum block size (as a power of 2) for a VPI. Use the **force** keyword.
Step 3	`Switch(config-if)# atm input-xlate-table minblock vpi vpi-value block-size force`	Repeats this command for as many VPIs as required.
Step 4	`Switch(config-if)# exit`	Returns to global configuration mode.

Verifying Minimum ITT Block Size

To verify that you successfully configured minimum ITT block size for an interface and VPI, use the **more system:running-config** EXEC command. Observe the block size for each interface and VPI using the **show atm-input-xlate-table inuse** command.

Enabling Automatic Determination of Minimum ITT Block Size

The NSP can automatically determine the optimal ITT block size for every configured interface and VPI. Based on the analysis of PVCs and Soft PVCs, it can insert **minblock** commands for each interface and VPI combination.

To enable the automatic analysis and determination of minimum block size, use the following command in global configuration mode:

Command	Purpose
`Switch(config)# atm input-xlate-table autominblock`	Enables the automatic determination of minimum block size for each interface and VPI combination.

Verifying Automatic Determination of Minimum ITT Block Size

To verify that you successfully enabled **autominblock command**, use the **more system:running-config** EXEC command. Observe that the output includes **atm input-xlate-table autominblock** command in the running configuration.

Reclaiming ITT Space Dynamically

When high numbered VCs are deleted from the configuration, it is possible to shrink the corresponding ITT block in place and release the unused ITT resources . To enable automatic shrinking of ITT blocks, use the following command in global configuration mode:

Command	Purpose
`Switch(config)# atm input-xlate-table autoshrink`	Enables the automatic shrinking of ITT blocks inplace when high numbered VCs are deleted.

Verifying Automatic Shrinking of ITT Blocks

To verify that you successfully enabled the autoshrink command, use the **more system:running-config** EXEC command. Observe the block size using the **show atm input-xlate-table inuse** command.

Troubleshooting Tips

- To minimize fragmentation, avoid using large VCI values.
- To conserve processor and memory resource, disable autoshrink when the task is complete.

Monitoring and Maintaining ITT

The EXEC mode command **show atm input-xlate-table** allows you to view a comprehensive picture of ITT utilization, including the blocks that are used and available and the ports where the blocks are allocated. The output of the command shows details of the free blocks by size and bank, the aggregate free space left, and the location of in-use blocks. When you use the **show** command with the **inuse** keyword, the output of the command shows a detailed list of in-use blocks by port and VPI to which they are dedicated.

Command	Purpose
`Switch> show atm input-xlate-table [inuse]`	Displays the location and size of used and unused ITT blocks. When used with the optional **inuse** keyword, the command displays the location and size of ITT block for each interface and VPI.

Configuration Examples

This section provides the following configuration examples:

- Specify Minimum Block Size
- Enable Automatic Determination of Minimum ITT Block Size
- Enable Automatic Shrinking of ITT Blocks

Specify Minimum Block Size

The following example shows how to specify minimum block size for VPIs on an interface by manual configuration:

```
!
interface atm 1/0/0
    atm input-xlate-table minblock vpi 0 1024 force
    atm input-xlate-table minblock vpi 1 2048 force
    atm input-xlate-table minblock vpi 4 1024 force
    exit
!
```

Enable Automatic Analysis of Minimum ITT Block Size

The following example shows how to enable automatic analysis of minimum ITT block requirements for each VPI on an interface when generating a startup configuration file:

```
!
atm input-xlate-table autominblock
!
```

Enable Automatic Shrinking of ITT Blocks

The following example shows how to shrink an ITT block in place and release the unwanted ITT resources when high numbered VCs are deleted from the configuration:

```
!
atm input-xlate-table autoshrink
interface atm 1/0/0
    no atm pvc 0 1010
    no atm pvc 0 1011
    exit
no atm input-xlate-table autoshrink
!
```

Command Reference

This section documents three new configuration mode commands and one new exec mode command. All other commands used with this feature are documented in the Cisco IOS Release 12.1 command reference publications and *ATM Switch Router Command Reference* publication.

- **atm input-xlate-table autominblock**
- **atm input-xlate-table autoshrink**
- **atm input-xlate minblock**
- **show atm input-xlate-table**

atm input-xlate-table autominblock

To enable automatic determination of minimum Input Translation Table block sizes for all virtual path identifiers (VPI's) populated with permanent virtual circuits (PVC's) and Soft PVC source legs, use the **atm input-xlate-table autominblock** global configuration mode command. To disable autominblock mode, use the **no** form of this command.

atm input-xlate-table autominblock

no atm input-xlate-table autominblock

Syntax Description

This command has no arguments or keywords.

Defaults

Autominblock mode is disabled.

Command Modes

Global configuration mode

Command History

Release	Modification
12.1(4)DB	This command was introduced on the Cisco 6400 series.

Usage Guidelines

When you enable the autominblock mode on a Node Switch Processor (NSP), the system analyzes all interfaces and determines minimum ITT block sizes for all VPIs populated with PVCs and Soft PVC source legs that can use the ITT (but a point to-multipoint leaf would not be included). The autominblock mode also updates the analysis when virtual circuits (VC's) are added or deleted. If you generate the configuration command-line interface (CLI), for example, by using the **more system:running-config** command, **minblock** commands are inserted as appropriate for all VPIs populated with PVCs on all external ATM interfaces.

You can specify a minimum block size for an interface and VPI combination by using the **force** keyword, even when the **autominblock** command is enabled. See the following table for the effect of **minblock** commands in different situations:

Enable autominblock command	Use force keyword	Effect
true	true	The *blocksize* value is not overridden by automatic analysis.
true	false	The *blocksize* value can be overridden by automatic analysis.
false	true	The command is accepted.
false	false	The command is not accepted.

Note Enabling the autominblock mode does not alter the usage state of ITT at the moment the command is invoked in a running system. Since ITT memory may have already been allocated, autominblock mode alters only the subsequent behavior of the system.

Similarly, autominblock mode does not eliminate fragmentation generated as a result of VCs configured after the autominblock mode is enabled, but ensures that the software allocates optimal-size blocks on interface flaps subsequent to the definition of a PVC with a maximum VCI.

To ensure that the software subsequently allocates optimal-sized ITT blocks—even after you restart the system—enable the autominblock mode before or after all PVCs are defined. Save the configuration by using the **copy running-config startup-config** command.

You do not need to set up a cross-connect to specify a minimum block through autominblock analysis. Because autominblock analysis considers PVC half legs, you can use these half legs to establish the maximum intended virtual channel identifier (VCI) range, which helps to eliminate the growth of ITT block sizes in the future.

Note When the initial attempt to allocate an ITT block fails and the actual size needed is less than the specified minimum block size, the software retries the allocation by using the actual size.

The system analyzes the ITT needs of VP tunnel subinterfaces and configures the parent physical interface accordingly. Enabling this command mode increases the use of processor and memory resources.

Examples

The following example shows how to enable autominblock mode:

```
!
atm input-xlate-table autominblock
!
```

Related Commands

Command	Description
atm input-xlate-table minblock	Specifies the minimum ITT block size for a specific VPI on an interface.

atm input-xlate-table autoshrink

To shrink the existing Input Translation Table (ITT) blocks in-place when high numbered virtual circuits (VC's) are deleted, use the **atm input-xlate-table autoshrink** configuration mode command. To disable autoshrink, use the **no** form of this command.

atm input-xlate-table autoshrink

no atm input-xlate-table autoshrink

Syntax Description

This command has no arguments or keywords.

Defaults

Disabled.

Command Modes

Global configuration mode

Command History

Release	Modification
12.1(4)DB	This command was introduced on the Cisco 6400 series.

Usage Guidelines

Be careful when enabling this command mode, because it increases the use of processor and memory resources.

Note This command cannot override the *blocksize* hints specified by using the **minblock/autominblock** commands.

Examples

The following example shows how to shrink an ITT block in place and release the unrequired ITT resources when high numbered VCs are deleted from the configuration:

```
!
atm input-xlate-table autoshrink
!
interface atm 1/0/0
    no atm pvc 0 1010
    no atm pvc 0 1011
    exit
!
no atm input-xlate-table autoshrink
```

Related Commands

Command	Description
atm input-xlate-table minblock	Specifies the minimum ITT block size for a specific virtual path identifier (VPI) on an interface.
atm input-xlate-table autominblock	Enables automatic determination of minimum ITT block size needed for each VPI populated by PVC and Soft PVC source legs.

atm input-xlate-table minblock

To specify the minimum Input Translation Table (ITT) block size for a specific virtual path identifier (VPI) on an interface, use the **atm input-xlate-table minblock** interface configuration mode command. To remove the minimum block size specification, use the **no** form of this command.

atm input-xlate-table minblock vpi *vpi- value blocksize* **force**

no atm input-xlate-table minblock vpi *vpi-value*

Syntax Description

vpi-value	VPI to which the command applies. Values are in the range 0 to 255.
blocksize	Block size rounded up to the smallest power of 2 greater than or equal to the entered value. Values are in the range 32 to 16,384.
force	Indicates a user-entered value that is not overridden by autominblock analysis and is saved in configuration generation.

Defaults

No minimum block size is specified.

Command Modes

Interface configuration mode

Command History

Release	Modification
12.1(4)DB	This command was introduced on the NSP.

Usage Guidelines

Ensure that the block size you specify corresponds to the desired virtual circuit (VC) usage, so that the system resource is not overused. ITT resource is used only when a connection is installed (when both interfaces that the VC transits are up).

This command is particularly useful when you anticipate the needs of switched virtual circuits (SVC's) transiting an interface and VPI, and you want to avoid the fragmentation associated with ITT growth. If the initial ITT block allocation fails and the actual block size needed [to accomodate a specific virtual channel identifier (VCI)] is less than the specified minimum block size, the software retries the allocation by using the actual size.

Note You cannot use this command on a virtual path (VP) tunnel subinterface. To specify minimum block size for a VP tunnel subinterface, configure this command on the main interface.

Examples

The following example shows how to specify minimum block size for VPIs on an interface by manual configuration:

```
!
interface atm 1/0/0
    atm input-xlate-table minblock vpi 0 1024 force
    atm input-xlate-table minblock vpi 1 2048 force
    atm input-xlate-table minblock vpi 4 1024 force
    exit
!
```

Related Commands

Command	Description
atm input-xlate-table autominblock	Automatically determines the minimum ITT block size needed for each VPI that is populated by PVCand Soft PVC source legs.

show atm input-xlate-table

To view the Input Translation Table utilization details, use the **show atm input-xlate table** command in EXEC mode.

show atm input-xlate table [inuse]

Syntax Description		
	inuse	Shows a detailed list of in-use blocks by port and virtual path identifier (VPI) .

Defaults

No default behavior or values.

Command Modes

EXEC

Command History

Release	Modification
12.1(4)DB	This command was introduced on the NSP.

Examples

The following example shows how to use the command to view the ITT utilization details—with and without the **inuse** keyword:

```
Switch>show atm input-xlate-table
Input Translation Table Free Blocks:
Block-start   Size     Bank
1             1         0
2             2         0
4             4         0
8             8         0
16            16        0
32            32        0
64            64        0
17408         64        0
128           128       0
17536         128       0
256           256       0
17664         256       0
512           512       0
17920         512       0
1024          1024      0
2048          2048      0
18432         2048      0
4096          4096      0
20480         4096      0
8192          8192      0
24576         8192      0
32769         1         1
32770         2         1
32772         4         1
32776         8         1
32784         16        1
32800         32        1
49248         32        1
32832         64        1
49152         64        1
49344         64        1
32896         128       1
33024         256       1
49408         256       1
33280         512       1
49664         512       1
33792         1024      1
50176         1024      1
34816         2048      1
51200         2048      1
36864         4096      1
53248         4096      1
40960         8192      1
57344         8192      1

Input Translation Table Total Free = 64350

Input Translation Table In Use (display combines contiguous blocks):
 Inuse-start   Inuse-end   Size
0             0            1
16384         17407        1024
17472         17535        64
32768         32768        1
49216         49247        32
49280         49343        64
```

The output of the command with the **inuse** keyword is:

```
Switch>show atm input-xlate-table inuse
Interface        VPI  VP/VC Address Size
ATM0/1/0         0      VC  17472   64
ATM0/1/0         2      VP  32768   1
ATM0/1/2         0      VC  49216   32
ATM0/1/2         2      VP  0       1
ATM1/0/0         0      VC  49280   64
ATM1/0/0         9      VC  16384   1024
```

Related Commands

Command	Description
atm input-xlate-table autominblock	Automatically determines the minimum ITT block size needed for each VPI populated by permanent virtual circuits (PVC's) and Soft PVC source legs.
atm input-xlate-table autoshrink	Shrinks the existing ITT blocks in place when high-numbered virtual circuits (VC's) are deleted.

Glossary

ATM—Asynchronous Transfer Mode. International standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and TI.

ITT—Input Translation Table. A hardware data structure that is used in the Cisco 6400 series to handle the incoming cells.

NSP—Node Switch Processor. The main system processor card in the Cisco 6400 series, residing in chassis slot 0A or 0B. The NSP ATM switch engine and processor and most memory components. The NSP runs the system software, which maintains and executes the management functions that control the system.

PVC—permanent virtual circuit or connection. Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection. Compare with SVC. See also virtual circuit.

SoftPVC—Soft PVCs are a hybrid between switched and permanent connections. Soft PVCs are specified by source and destination VPI/VCI values and the destination ATM address. They are then set up through signaling but, unlike SVCs, remain up until manually torn down.

subinterface—One of a number of virtual interfaces on a single physical interface.

SVC—switched virtual circuit. Virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic. Called a switched virtual connection in ATM terminology. Compare with PVC.

VC—Logical circuit created to ensure reliable communication between two network devices. A virtual circuit is defined by a VPI/VCI pair, and can be either permanent (PVC) or switched (SVC). Virtual circuits are used in Frame Relay and X.25. In ATM, a virtual circuit is called a virtual channel. Sometimes abbreviated VC. See also PVC, SVC, VCD, virtual route, and VPI.

VCC—virtual channel connection. Logical circuit, made up of VCLs, that carries data between two end points in an ATM network. Sometimes called a virtual circuit connection. See also VCD, VCL, and VPI.

VCD—virtual circuit descriptor.

VCI—virtual channel identifier. 16-bit field in the header of an ATM cell. The VCI, together with the VPI, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next network VCL that a cell needs to transit on its way to its final destination. The function of the VCI is similar to that of the DLCI in Frame Relay. Compare with DLCI. See also VCL and VPI.

VCL—virtual channel link. Connection between two ATM devices. A VCC is made up of one or more VCLs. See also VCC.

VP—Logical grouping of virtual circuits that connect two sites. See also virtual circuit.

VPC—virtual path connection. Grouping of VCCs that share one or more contiguous VPL. See also VCC and VPL.

VPI—virtual path identifier. Eight-bit field in the header of an ATM cell. The VPI, together with the VCI, is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. ATM switches use the VPI/VCI fields to identify the next VCL that a cell needs to transit on its way to its final destination. The function of the VPI is similar to that of the DLCI in Frame Relay. Compare with DLCI. See also VCD and VCL.

VPL—virtual path link. Within a virtual path, a group of unidirectional VCLs with the same end points. Grouping VCLs into VPLs reduces the number of connections to be managed, thereby decreasing network control overhead and cost. A VPC is made up of one or more VPLs.