APPENDIX C

Optimizing the Number of Virtual Connections on the Cisco 6400

This appendix describes how to optimize the number of supported virtual connections on the Cisco 6400 carrier-class broadband aggregator by:

- Properly assigning virtual path identifier (VPI) and virtual channel identifier (VCI) values
- Reducing Input Translation Table (ITT) memory fragmentation

For general information on configuring virtual connections, see the "Configuring Virtual Connections" chapter of the *ATM Switch Router Software Configuration Guide*.

Note The method of assigning VPIs and VCIs can affect how you configure virtual connections on network devices that are connected to the Cisco 6400.

This appendix includes the following sections:

An Overview of the ITT and Virtual Connection Limitations

The Cisco 6400 supports a maximum of 32K virtual connections, a limit determined by a hardware data structure in the node switch processor (NSP) called the ITT. The ITT consists of two banks (Bank 0 and Bank 1), each of which can hold a maximum of 32K entries. Each configured virtual connection occupies two ITT entries (one for each direction of cell flow). Unidirectional connections (such as point-to-multipoint connections) occupy only one ITT entry.

ITT entries are *not* maintained for:

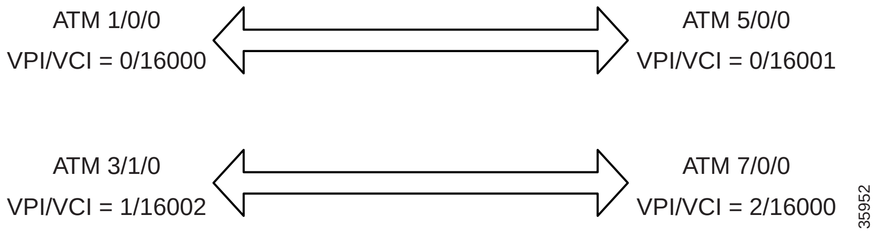
- Permanent virtual circuit (PVC) legs that are not connected
- Connection legs on ATM 0/0/0
- PVCs with one or both interfaces in the down state
- Any connection that does not receive cells (such as point-to-multipoint leaves)

How VCI Values Limit the Number of Virtual Connections

Each ATM interface supports VPIs as large as 8 bits (0 to 255) and VCIs as large as 14 bits (0 to 16,383). While these ranges provide a broad selection of VPI/VCI combinations per interface (up to 4,194,304), the method that you use to select these combinations can affect how many virtual connections you can configure on the Cisco 6400.

The ITT allocates resources in blocks of adjacent entries where each block size, in bits, must be a power of 2. Each VPI and ATM port combination requires a dedicated ITT block, and the block size must be greater than the largest VCI. As a result, using unnecessarily large VCI values can dramatically reduce the number of supported virtual connections.

Figure C-1 Example of Two PVCs Using Extremely High VCI Values



In Figure C-1, two PVCs are configured between four ATM ports. In this example, all VCIs are close to the maximum allowed VCI value of 16383. Because the ITT block size must be a power of 2, each of the four ATM port/VPI/VCI combinations require 16K of allocated ITT resources. As a result, these two PVCs exhaust all possible ITT resources, and additional ATM port and VPI combinations cannot be configured.

How ITT Fragmentation Limits the Number of Virtual Connections

Each VPI and ATM port combination requires a dedicated ITT block, and the block size must be greater than the largest VCI. If you configure a VCI greater than the current size of an existing ITT block, the block must expand to the next power of 2 block size that can accommodate the new VCI. The method of ITT block expansion, however, often results in many small and unusable fragments, and further limits the number of virtual connections configurable on the Cisco 6400.

For an existing ITT block to expand in size:

1. The ITT allocates a new block within the same bank. The block size is determined by the largest VCI value, rounded up to the next power of 2.
2. The ITT copies the entries from the original block to the new block.
3. The ITT frees the original block from allocation.

As the ITT allocates, expands, and frees its blocks, the total memory breaks into fragments of used memory and free memory. The total free memory can be larger than the size of any single block, but the fragments might be too small to use.

Guidelines for Maximizing the Number of Virtual Connections

Use the following methods to maximize the number of virtual connections on the Cisco 6400:

- Assigning VCI Values to Maximize the Number of Entries per Block
- Specifying the Minimum ITT Block Size
- Using Automatic Determination of the Minimum ITT Block Size
- Shrinking ITT Blocks
- Displaying ITT Allocation

Assigning VCI Values to Maximize the Number of Entries per Block

For each ATM port and VPI combination, assign VCIs using the following guidelines:

Step 1 Begin configuring virtual connections with VCI value 32. VCI values 0 through 31 are reserved by the ATM Forum for particular functions, such as the Interim Local Management Interface (ILMI).

Step 2 Incrementally assign VCI values for additional virtual connections for the ATM port and VPI combination. Avoid skipping any numbers. This incremental assignment prevents the ITT from allocating a larger block than is necessary for the virtual connections.

Example

Suppose you want to configure five virtual connections between ATM 1/0/1 (VPI = 0) and ATM 8/0/0 (VPI = 5). Configure the virtual connections in the order shown in Table C-1. Notice that the VCI values increase without skipping any numbers.

Table C-1 Virtual Connections Between ATM 1/0/1 (VPI = 0) and ATM 8/0/0 (VPI = 5)

VPI/VCI Values on ATM 1/0/1	VPI/VCI Values on ATM 8/0/0
0/32	5/32
0/33	5/33
0/34	5/34
0/35	5/35
0/36	5/36

Verifying VCI Values

To verify that you optimally configured virtual connections on a particular ATM interface, use the **show atm vc interface atm** *slot/subslot/port* EXEC command. Check that for each VPI value, the VCI values start at 32 and do not skip any numbers.

Specifying the Minimum ITT Block Size

If you know the maximum VCI that will be used for a particular ATM port and VPI combination, you can use the highest VCI to determine the minimum ITT block size for that ATM port and VPI combination. Specifying the minimum block size reduces fragmentation by avoiding block expansion as virtual connections are created.

Note This functionality is available in Cisco IOS Release 12.1(4)DB and later releases.

To specify the minimum ITT block size for an ATM port and VPI combination, complete the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	`Switch(config)#` **`interface atm`** *`slot/subslot/port`*	Specifies the ATM interface.
Step 2	`Switch(config-if)#` **`atm input-xlate-table minblock vpi`** *`vpi-value block-size`* **`force`**	Specifies the VPI and minimum ITT block size. The block size is rounded up to the smallest power of 2. The **force** keyword enables the specified block size to override the size determined by the **autominblock** keyword.

Note If the system cannot accommodate the specified minimum ITT block size, the system tries to allocate a block large enough to accommodate the VCI of the virtual connection being created.

Example

In the following example, the minimum ITT block sizes are specified for virtual connections on ATM 1/0/0 with VPI values 0, 1, and 4:

```
!
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
  !
```

Verifying the Minimum ITT Block Size

To verify successful configuration of the minimum ITT block size, use the **more system:running-config** EXEC command. Make sure that the **atm input-xlate-table minblock** command appears for the correct interfaces and VPI values.

To verify that the minimum block size was allocated, use the **show atm input-xlate-table inuse** EXEC command. Check the Size field for the ATM port and VPI combinations that you configured.

Using Automatic Determination of the Minimum ITT Block Size

The NSP can automatically track the size of the required ITT block as virtual connections are created and deleted. The required block size is stored in the running configuration and is used to optimally allocate ITT resources after an interface flap. When you enter the **copy system:running-config nvram:startup-config** EXEC command, the required block size is also saved to the startup configuration, so that optimal ITT allocation occurs at the next reload.

Note This functionality is available in Cisco IOS Release 12.1(4)DB and later releases.

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

Command	Purpose
`Switch(config)# atm input-xlate-table autominblock`	Enables the system to determine the optimal ITT block size for every ATM port and VPI on which PVCs or Soft PVC source legs are configured. Automatically inserts the **minblock** keyword version of the command with the optimal block size for each ATM port and VPI combination.

Note The **no** version of the **atm input-xlate-table autominblock** command deletes all manually or automatically configured **minblock** keyword entries, except the entries that include the **force** keyword.

Example

In the following example, the system determines the optimal ITT block size for all ATM port and VPI combinations in the configuration. The system automatically inserts the **minblock** keyword version of the command for every ATM port and VPI combination, except when manually entered with the **force** keyword.

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

Verifying Automatic Determination of the Minimum ITT Block Size

To verify that you successfully enabled automatic determination of the minimum ITT block size, use the **more system:running-config** EXEC command. Make sure that the **atm input-xlate-table autominblock** command appears in the running configuration.

To verify that the minimum block size is allocated globally, use the **show atm input-xlate-table inuse** EXEC command. Check the Size field for every ATM port and VPI on which PVCs or Soft PVC source legs are configured.

Shrinking ITT Blocks

Once an ITT block expands, it does not automatically shrink if the block becomes larger than necessary for the ATM port and VPI combination. The unused portion of the block, especially when adjacent to an unused fragment of ITT memory, can instead be used to create another ITT block.

Note This functionality is available in Cisco IOS Release 12.1(4)DB and later releases.

To enable ITT blocks to shrink in place when they are larger than necessary for the current virtual connections, use the following command in global configuration mode:

Command	Purpose
`Switch(config)#` **`atm input-xlate-table autoshrink`**	Enables ITT blocks to shrink in place when they are larger than necessary to accommodate the configured virtual connections.

Note Enable block shrinking only when you actively remove virtual connections or high VCI values. Block shrinking significantly increases processor and memory resource usage, and can affect system performance.

Example

In the following example, block shrinking is enabled while PVCs with high VCIs are deleted. Then block shrinking is disabled to prevent draining processor and memory resources.

```
Switch(config)# atm input-xlate-table autoshrink
Switch(config)# interface atm 1/0/0
Switch(config-if)# no atm pvc 0 1010
Switch(config-if)# no atm pvc 0 1011
Switch(config-if)# exit
Switch(config)# no atm input-xlate-table autoshrink
```

Verifying ITT Block Shrinking

To verify ITT block shrinking, use the **show atm input-xlate-table inuse** EXEC command in combination with the **show atm vc interface atm** *slot/subslot/port* EXEC command. The displayed size of the blocks in use should be just large enough to accommodate the displayed VCIs.

Displaying ITT Allocation

To display ITT allocation, use the following command in EXEC mode:

Command	Purpose
`Switch>` **`show atm input-xlate-table`** `[`**`inuse`**`]`	Displays ITT allocation, including both used and unused ITT memory. Optionally, with the **inuse** keyword, displays only the allocated ITT blocks and the ATM interfaces and VPI values associated with each block.

Note This functionality is available in Cisco IOS Release 12.1(4)DB and later releases.

Examples

In the following example, the **inuse** keyword is used to display all allocated ITT blocks and their associated ATM interfaces and VPIs:

```
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
```

In the following example, the **inuse** keyword is excluded to display both the used and free ITT blocks:

```
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
```