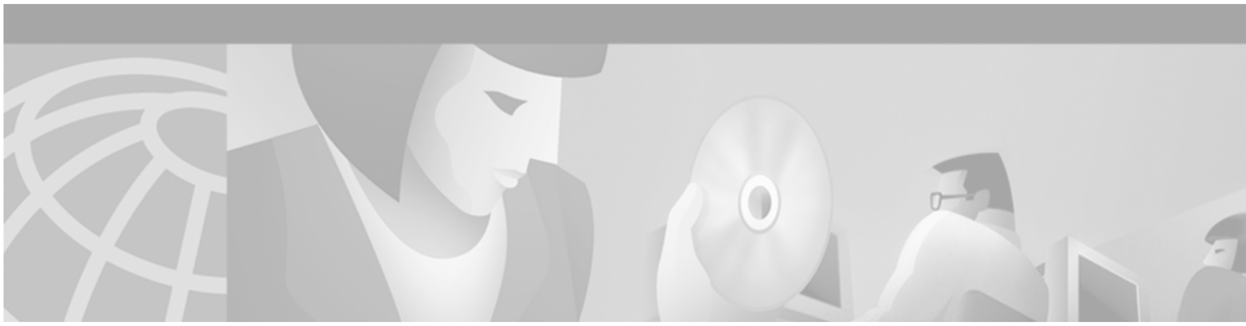


Configuring Synchronized Clocking

The Cisco MC3810 supports voice and video streams in addition to traditional data streams. Because voice and video streams are real-time streams and originate from synchronous devices, it is important to configure the synchronous clocking to prevent data corruption and data loss. For a description of the commands used to configure synchronized clocking, refer to the *Cisco IOS Multiservice Applications Command Reference* publication.

Synchronous Clocking Concepts

Due to the real-time nature of voice and video, more configuration and planning is required for voice traffic than is required for traditional data traffic. Because voice and video streams are real-time and continuous, the information is normally generated by the source device and received by the destination device at a synchronized fixed rate. If the source and destination clocking is not synchronized, meaning the devices generate information at different rates, there will be a loss of information as one side overruns and the other side underruns.

As a result, for voice and video configurations a single master clock source must be configured to make the network synchronous. The master clock must be used as the clock source for all devices on the network, even when the voice traffic is compressed.

Clocking mismatches can be caused by a variety of configuration problems. The following situations can cause problems:

- Multiple clock sources on the network that are not synchronized.

  In back-to-back voice systems where the two devices are using different clock sources that are not synchronized, data loss can occur when one device over-runs the voice stream and the other device under-runs the voice stream.

  In situations where there is a minor clock mismatch, the Cisco MC3810 may be able to process the mismatch in its internal voice coders, in the same way that the voice coders handle minor network delay and jitter. The voice waveform will be degraded but often not noticeably.

  However, when the Cisco MC3810 is using circuit emulation services (CES) to send video traffic, no similar clock compensation is possible as the CES must be in synchronous mode. As a result, when video traffic is sent over a nonsynchronized network, data corruption may occur. This situation will cause video devices connected to the MC3810 to lose frame synchronization and enter a frame-search mode, causing noticeable data loss. Because of these requirements, the network clocks must be synchronized when processing video traffic on the Cisco MC3810.

- Layer 1 conflicts

  Layer 1 conflicts can take place when a Cisco MC3810 with two Multiflex Trunks (MFTs) is placed at the border of two separately clocked T1 or E1 networks and is forced to resolve the clock difference between the networks. As a result, DS1 clock and frame slips can occur, which can result in lengthy reframe times, and can cause an attached DS1 device to declare the line down.

Configuring the Cisco MC3810 to a Synchronous Clocked Network

To ensure a synchronized system, you must configure a master clock somewhere within the network, and distribute and recover the clock throughout the network. This will allow end devices at opposite ends of the network to reference a common clock source. If you cannot configure a synchronized system, then you can configure multiple clock sources on your network as long as they are accurate enough that the clocking on both clock sources will match.

You can statically configure the Cisco MC3810 to receive or generate clocking using one of the following scenarios:

- Obtain the synchronous clock from a network device attached to controller T1 or E1 0 and distribute the clocking to the other controller and the universal input/output (UIO) serial ports.
- Obtain the synchronous clock from a network device attached to Controller T1 or E1 1, and distribute the clocking to the other controller and the UIO serial ports.
- Obtain the synchronous clock from a network device directly attached to serial port 0 (in DTE mode only) and distribute the clocking to the other serial ports and to both controllers.
- Generate the clock internally on the Cisco MC3810 and distribute the clocking to all interfaces.
- When in T1 or E1 mode, all MFT types can provide either line or internal clocking. When one controller is configured to line clocking (obtaining the clocking from the network), the other controller must be configured to internal clocking (obtaining the clocking internally from the other controller).

Note Configuring a clock source from the digital voice module (DVM) is supported if the installed DVM is either hardware version 4.50 or later, and the system control board (SCB) is version 6.05 or later. To verify the hardware version of the SCB, enter the **show version** command and check the entry for the Cisco MC3810 processor revision. To verify the hardware version of the DVM, enter the **show controller T1/E1** command and check the HWVersion entry.

For more information on how to configure clocking for these scenarios, see the "Configuring the Clock Source for the Cisco MC3810" section later in this chapter.

In addition, you can define a hierarchy of potential clock sources so that when the primary clock source goes down, the Cisco MC3810 can automatically switch to a backup clock source. For more information, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this chapter.

Configuring the Clock Source for the Cisco MC3810

Because of the different ways that PSTNs provide clocking and how data networks provide clocking, there may be incompatibilities when the Cisco MC3810 is used to integrate voice and data networks. As a result, the Cisco MC3810 must synchronize the disparate clocking, and you must be careful in how you configure your clock sources. The clocking can be derived from one of the following sources:

- The PBX
- The video CODEC (for video applications)
- The ATM or Frame Relay WAN carrier
- The Cisco MC3810 internal clock

Depending on the configuration, you must determine how to configure the appropriate interface on the Cisco MC3810 for the clocking configuration. Each interface provides different clocking support, and depending on the interface used, the commands required to configure the clocking are different. You must also determine whether the Cisco MC3810 interface will be the DCE or the DTE in the configuration.

The following sections provide configuration tasks:

- Configuring the Cisco MC3810 to Obtain Clocking from the Network
- Configuring the Cisco MC3810 to Use the Internal Clock Source

Configuring the Cisco MC3810 to Obtain Clocking from the Network

This section describes several scenarios for statically configuring clocking on the Cisco MC3810. This section includes the following procedures:

- Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to a T1/E1 Controller
- Configuring a T1/E1 Controller to Loop-Time the Clocking Back to the Network Clock Source
- Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to Serial 0

Note The procedures in this section statically configure the clock source for the interfaces. If the clock source fails, these procedures do not configure a backup clock source. For information on configuring a hierarchy of backup clock sources, see the “Configuring a Hierarchy of Clock Sources for Backup Purposes” section later in this chapter.

Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to a T1/E1 Controller

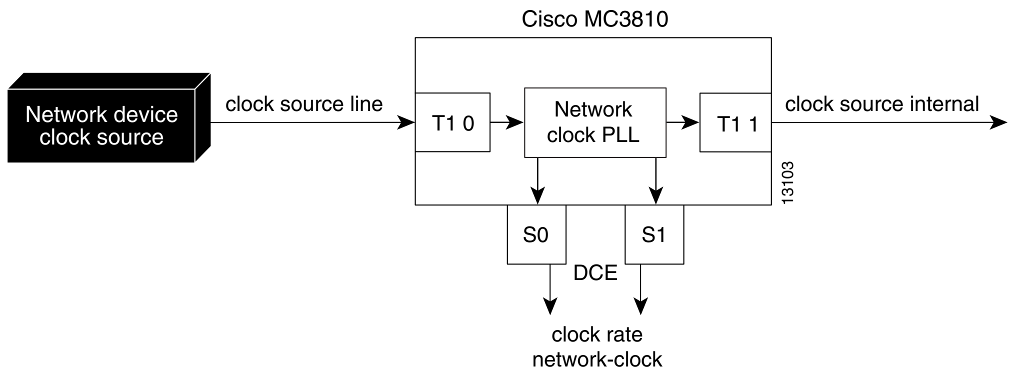
When the Cisco MC3810 recovers clocking from a network device attached to a T1 or E1 controller, the clock recovery circuit on the controller will place a recovered 2 MHz clock on the common circuit toward the network clock phase-lock-loop (PLL). Once the network-clock PLL circuit receives the valid 2 MHz clock from the controller, the network clock PLL synchronizes to the recovered clock and redistributes the clock to the rest of the system. The other T1/E1 controller and the serial ports on the Cisco MC3810 then derive their clocking from the network clock PLL.

When you configure a T1/E1 controller to recover clocking from a network device, configure the **clock-source** controller configuration command to the **line** setting.

> **Note** Do not configure both T1/E1 controllers to the **line** setting. Doing so will cause both controllers to attempt to drive the network clock PLL at the same time. If you configure both T1/E1 controllers to **line**, there will be clocking conflicts. You will not receive an error message if you misconfigure the clocking in this way. Instead, configure one controller for line timing and the other controller for internal or loop timing.
>
> The one exception to this rule is if you configure backup clocks to dynamically activate if the primary clock fails. For more information, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this chapter.

Figure 116 shows an example in which the Cisco MC3810 obtains its clock source from a network device attached to controller T1/E1 0 (the MFT).

Figure 116 Obtaining the Clock Source from a Network Device Attached to Controller T1/E1 0



To make sure the network is synchronized, configure the attached network device that obtains its clocking from the Cisco MC3810 (from the T1/E1 controller clock source set to **internal**) to derive its clock from the T1/E1 signal sent by the Cisco MC3810. If the T1/E1 signal received from the attached network device is not synchronous with the Cisco MC3810 network clock, then frame and clock slips will occur at the T1/E1 controller, causing loss of data.

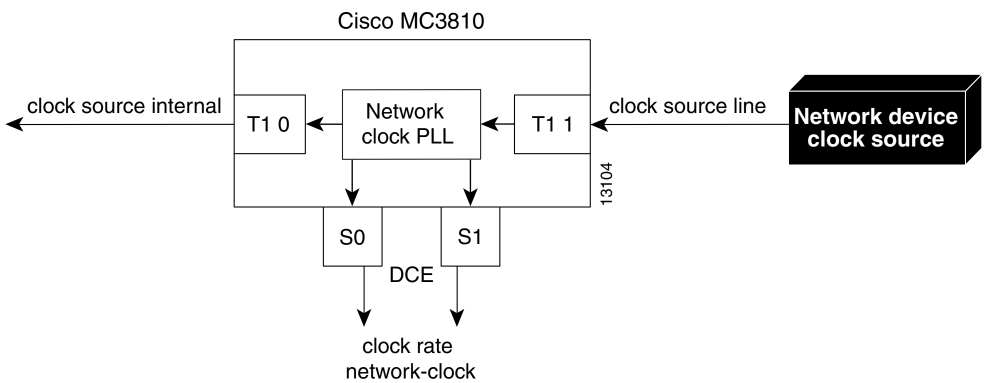
To configure the Cisco MC3810 to obtain its clock source from a network device attached to controller T1/E1 0, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`0`**	Enters controller configuration mode for controller T1/E1 0.
Step 2	`Router(config-controller)#` **`clock source line`**	Configures controller T1/E1 0 to obtain the Cisco MC3810 clock source from an attached network device.
Step 3	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`1`**	Enters controller configuration mode for controller T1/E1 1.
Step 4	`Router(config-controller)#` **`clock source internal`**	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL.
Step 5	`Router(config-controller)#` **`network-clock base-rate`** `{`**`56k`** `|` **`64k`**`}`	Sets the network clock base rate for the serial ports. The default is 56k.

	Command	Purpose
Step 6	`Router(config)#` **`interface serial 0`**	Enters interface configuration mode for serial 0.
Step 7	`Router(config-if)#` **`clock rate network-clock`** *`rate`*	Configures the network clock speed for DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command in Step 5. Repeat Steps 6 and 7 for serial port 1.
Step 8	`Router(config)#` **`exit`**	Exits configuration mode.
Step 9	`Router#` **`show network-clocks`**	Displays the network clock configuration.

Figure 117 shows an example in which the Cisco MC3810 obtains its clock source from a network device attached to controller T1/E1 1 (the DVM).

Figure 117 Obtaining the Clock Source from a Network Device Attached to Controller T1/E1 1



To configure the Cisco MC3810 to obtain its clock source from a network device attached to controller T1/E1 1, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`1`**	Enters controller configuration mode for controller T1/E1 1.
Step 2	`Router(config-controller)#` **`clock source line`**	Configures controller T1/E1 1 to obtain the Cisco MC3810 clock source from an attached network device.
Step 3	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`0`**	Enters controller configuration mode for T1/E1 0 to configure the clock source for the MFT.
Step 4	`Router(config-controller)#` **`clock source internal`**	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL.
Step 5	`Router(config-controller)#` **`network-clock base-rate`** `{`**`56k`** `|` **`64k`**`}`	Sets the network clock base-rate for the serial ports. The default is 56k.
Step 6	`Router(config)#` **`interface serial 0`**	Enters interface configuration mode for serial 0.

	Command	Purpose
Step 7	`Router(config-if)#` **`clock rate network-clock`** *`rate`*	Configures the network clock speed for DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command in Step 5. Repeat Steps 6 and 7 for serial port 1.
Step 8	`Router(config)#` **`exit`**	Exits configuration mode.
Step 9	`Router#` **`show network-clocks`**	Displays the network clock configuration.

Configuring a T1/E1 Controller to Loop-Time the Clocking Back to the Network Clock Source

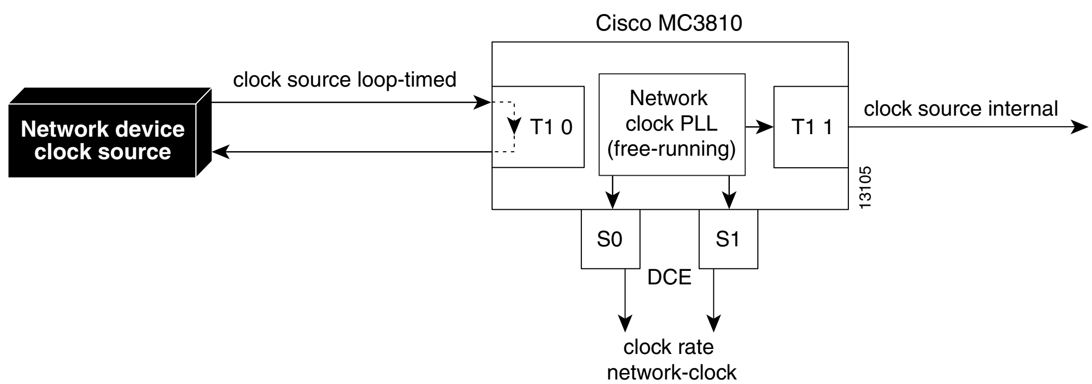
When you configure a T1/E1 controller to loop-time the clocking back to a network device, you configure the **clock-source** controller command to the **loop-timed** setting. The **clock-source** command on the other T1/E1 controller should in most cases be set to the **internal** setting.

When a controller's clock source is set to loop-timed, the internal network clock PLL is placed into free-running mode.

Note Use caution when configuring the controller clock source to loop-timed. This setting should only be used in certain cases, such as when there are two master clocks but you can only obtain clocking from one master clock at a time. Using the functionality to configure a hierarchy of clock sources, you can configure a controller set to loop-timed clock source to become the Cisco MC3810 clock source if the primary clock source fails. For more information about configuring a hierarchy of clock sources, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this chapter.

Figure 118 shows an example of a configuration in which the input clock source on the MFT is loop-timed back to the clock source device.

Figure 118 Loop-Timed Clock Source on a T1/E1 Controller



To configure the Cisco MC3810 to use loop-timed clock mode on controller T1/E1 0, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`controller {T1	E1} 0`**	Enters controller configuration mode for T1/E1 0.
Step 2	`Router(config-controller)#` **`clock source loop-timed`**	Configures controller T1/E1 0 to take the clock from the receive line and send it back to the source.
Step 3	`Router(config)#` **`controller {T1	E1} 1`**	Enters controller configuration mode for T1/E1 1.
Step 4	`Router(config-controller)#` **`clock source internal`**	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL. **Note** To configure controller T1 1 for loop-timed mode, follow the same configuration procedure, but change the controller that will be configured for loop-timed mode.
Step 5	`Router(config-controller)#` **`network-clock base-rate {56k	64k}`**	Sets the network clock base rate for the serial ports. The default is 56k.
Step 6	`Router(config)#` **`interface serial 0`**	Enters interface configuration mode for serial 0.
Step 7	`Router(config-if)#` **`clock rate network-clock`** *`rate`*	Configures the serial 0 network clock speed for DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command in Step 5. Repeat Steps 6 and 7 for serial port 1.
Step 8	`Router(config)#` **`exit`**	Exits configuration mode.
Step 9	`Router#` **`show network-clocks`**	Displays the network clock configuration.

Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to Serial 0

If serial interface 0 is configured as a DTE, it can accept clocking from the attached DCE and use the clocking to drive the network-clock PLL on the Cisco MC3810. The clocking is then distributed to the T1/E1 controllers and to serial interface 1.

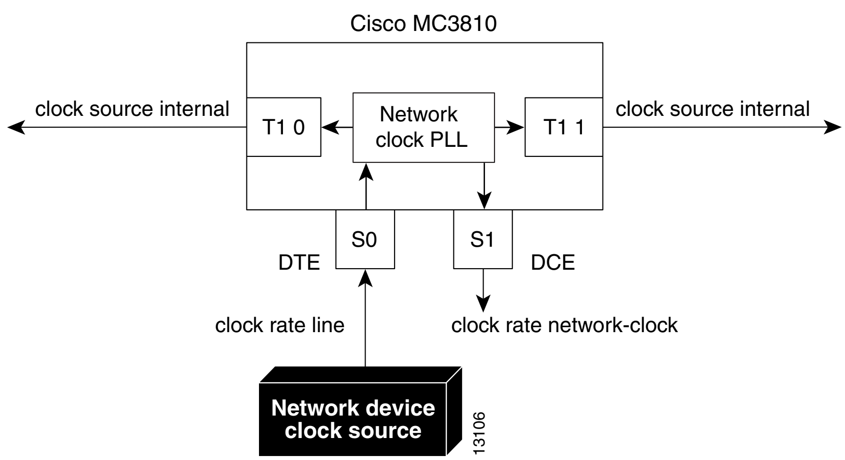
Because the input to the network clock PLL must be 2 MHz, a clock multiplier circuit is used to multiply the incoming clock on serial 0 to 2 MHz in 8 Hz increments. This multiplier is configured using the **clock-rate line** serial interface command. This command is valid only when serial 0 is configured as the DTE device.

Note To recover clocking over serial interfaces, the Cisco MC3810 can recover clocking only from a device attached to serial 0 in DTE mode. It cannot recover clocking from a device attached to serial 1, or to serial 0 in DCE mode.

Note When Q.SIG, ISDN, or the BRI Voice Module (BVM) is enabled, Serial 1 is normally configured for DCE. If Serial 1 is configured as a DTE, you need to make sure that the clock driving serial 1 comes from the same source as the clock driving the system. When Q.SIG, ISDN, or the BVM is enabled, the CPU takes the serial 1 data in time-slot mode that is driven by the system clock. If this clock is different from the clock driving the data into Serial 1, there will be cyclic redundancy check (CRC) errors and the line will not come up.

Figure 119 shows an example of the Cisco MC3810 obtaining clocking from a network device attached to Serial 0.

Figure 119 Clock Source from a Network Device Attached to Serial 0



To configure the Cisco MC3810 to use a network device attached to serial port 0 as the clock source, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`network-clock base-rate`** `{`**`56k`** `|` **`64k`**`}`	Sets the network clock base rate for the serial ports. The default is 56k.
Step 2	`Router(config)#` **`network-clock-select 1 serial 0`**	Configures the network clock PLL to use the multiplied 2 Hz. clock from serial 0.
Step 3	`Router(config)#` **`interface serial 0`**	Enters interface configuration mode for serial 0.
Step 4	`Router(config-if)#` **`clock rate line`** *`rate`*	Configures the network clock line rate on serial 0 acting in DTE mode. The rate value is the rate of the incoming clock, and this value must be a multiple of 8 kHz.
Step 5	`Router(config)#` **`interface serial 1`**	Enters interface configuration mode for serial 1.
Step 6	`Router(config-if)#` **`clock rate network-clock rate`**	Configures the network clock line rate for serial 1 acting in DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 7	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`0`**	Enters controller configuration mode for T1/E1 0.

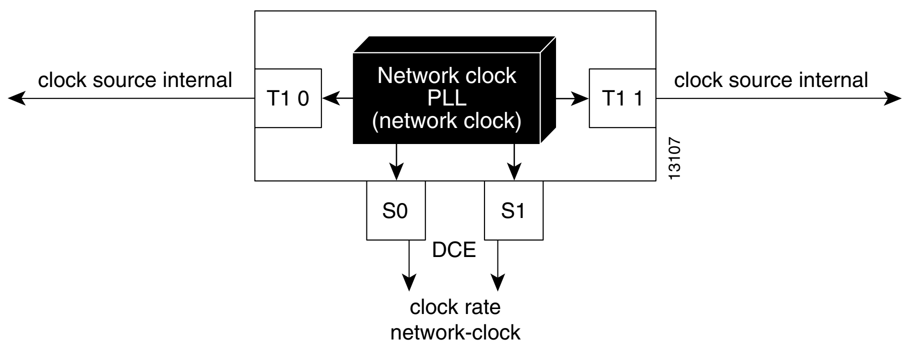
	Command	Purpose
Step 8	`Router(config-controller)# clock source internal`	Configures controller T1/E1 0 to obtain its clocking from the internal network clock PLL.
Step 9	`Router(config)# controller {T1 | E1} 1`	Enters controller configuration mode for T1/E1 1.
Step 10	`Router(config-controller)# clock source internal`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL.
Step 11	`Router(config)# exit`	Exits configuration mode.
Step 12	`Router# show network-clocks`	Displays the network clock configuration.

Configuring the Cisco MC3810 to Use the Internal Clock Source

When you configure the Cisco MC3810 to use the internal clock source, the clock source for both T1/E1 controllers is set to **internal** and the master clocking is generated from the Cisco MC3810 2 MHz network clock PLL. The internal clock source is accurate to a Stratum 4 level (±0.01 percent).

Figure 120 shows an example of the Cisco MC3810 using its internal clock source and propagating it outward onto the associated networks.

Figure 120 Using the Cisco MC3810 Internal Clock Source



To configure the Cisco MC3810 to use its internal 2 MHz clock as the clock source, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock base-rate {56k | 64k}`	Sets the network clock base rate for the serial ports. The default is 56k.
Step 2	`Router(config)# interface serial 0`	Enters interface configuration mode for serial 0.
Step 3	`Router(config-if)# clock rate network-clock` *rate*	Configures the network clock line rate on serial 0 acting in DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 4	`Router(config)# interface serial 1`	Enters interface configuration mode for serial 1.

	Command	Purpose
Step 5	`Router(config-if)#` **`clock rate network-clock`** *`rate`*	Configures the network clock line rate on serial 1 acting in DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 6	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`0`**	Enters controller configuration mode for T1/E1 0.
Step 7	`Router(config-controller)#` **`clock source internal`**	Configures controller T1/E1 0 to obtain its clocking from the internal network clock PLL.
Step 8	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`}` **`1`**	Enters controller configuration mode for T1/E1 1.
Step 9	`Router(config-controller)#` **`clock source internal`**	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL.
Step 10	`Router(config)#` **`exit`**	Exits configuration mode.
Step 11	`Router#` **`show network-clocks`**	Displays the network clock configuration.

Note When using the internal Cisco MC3810 clock source as the master clock, make sure to configure any other network devices directly attached to the Cisco MC3810 T1/E1 controllers and serial ports to obtain their clocking from the Cisco MC3810.

Configuring a Hierarchy of Clock Sources for Backup Purposes

The previous configurations apply when a static network clock source is desired with a single clock source. In some conditions, you may want to define a hierarchy of clock sources so that if the primary clock source fails, the system can be configured to use a secondary source, rather than switching to the internal clock (as in the previous configuration sections).

Using the **network-clock-select** command, you can configure a dynamic hierarchy of clock sources that are used if the primary clock source fails. Each clock source is assigned a priority. A higher priority number of a clock source places that source higher in the clocking hierarchy. The highest clock source priority is used as the default.

When a clock source fails, the Cisco MC3810 switches to the clock source in the hierarchy with the next highest priority. For example, if the clock source with priority 1 (the highest priority) fails, the Cisco MC3810 switches to the clock source with priority 2. Then, if the clock source with priority 2 fails, the Cisco MC3810 then switches to the clock source with priority 3 (assuming that the clock source with priority 1 has not become active in the meantime.)

If the module providing the clock experiences a failure (for example, if the T1/E1 controller experiences a Loss-of-Signal or Loss-of-Frame), then the clock source will be switched.

Note If you shut down a controller that is the current clock source, the shutdown will not cause the clock source to be switched.

To configure a hierarchy of clock sources for backup purposes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`network-clock-select`** *`1-4`* `[`**`serial 0`** `|` **`system`** `|` *`controller`*`]`	Specifies the highest priority clock source that will provide timing to the system backplane pulse code modulation (PCM) bus. Specify 1 and select the clock source to be assigned the highest priority. This clock source will be used as the default clock source unless it fails. Repeat Step 1, specifying each clock source with a lower priority. If the clock source with the highest priority fails, the clock source with the next priority is used. You can configure four clock sources: one primary clock source and up to three backup clock sources. For example, if serial 0 is priority 1 and controller T1 0 is priority 2, when serial 0 goes down the system will switch to controller T1 0 as the clock source.
Step 2	`Router(config)#` **`network-clock-switch`** `[`*`switch-delay`* `|` **`never`**`] [`*`restore-delay`* `|` **`never`**`]`	Configures the amount of time the network clock will wait before switching to a different clock, and the amount of time the current network clock will wait before recovering. The *switch-delay* option sets the duration the system waits before switching to the clock source with the next highest priority (as configured with the **network-clock-select** command). The *restore-delay* option sets the duration before the current network clock source recovers.
Step 3	`Router(config)#` **`controller`** `{`**`T1`** `|` **`E1`**`} {`**`0`** `|` **`1`**`}`	If one of the controllers will be used as a clock source in the hierarchy, enter controller configuration mode for the T1/E1 controller.
Step 4	`Router(config-controller)#` **`clock source line`**	Configures the controller to obtain the Cisco MC3810 clock source from an attached network device. If the other controller will be used as a potential clock source in the hierarchy, repeat Steps 3 and 4. **Note** To prevent clock source conflicts, make sure to configure both controllers to **clock source line** *after* configuring the **network-clock-select** commands. For more information about how clock source conflicts are resolved using this feature, see the section following this procedure.
Step 5	`Router(config)#` **`interface serial 0`**	If serial interface 0 will be used as a potential clock source in the hierarchy, enter interface configuration mode for serial 0.

	Command	Purpose
Step 6	`Router(config-if)# clock rate line rate`	Configures the network clock line rate on serial 0 acting in DTE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command.
Step 7	`Router(config)# exit`	Exits configuration mode.
Step 8	`Router# show network-clocks`	Displays the network clock configuration.

When you configure a hierarchy of clock sources, each potential clock source must be preconfigured to a mode that enables the Cisco MC3810 to derive the clock from that source. For example, if a controller will be a potential clock source, the controller clock source must be configured to **line**. If the controller clock source is configured to **internal**, the controller cannot be configured as a potential backup clock source using the **network-clock-select** command.

In the normal configuration, configuring both controllers to clock source line causes clocking conflicts. However, when configuring a hierarchy of clock sources, because only one controller is used as the primary clock source at one time, the conflict is prevented.

The following rules apply to configuring the clock source hierarchy:

- If a controller is a potential clock source in the hierarchy, the controller clock source must be configured to **line**.
- If a controller is a potential clock source in the hierarchy but is not currently being used as the clock source, the clock source setting for that controller is automatically switched to **loop-timed**. This is a temporary state set by the software to prevent a clocking conflict. If the controller becomes the clock source because another clock source fails, the clock source setting for the controller switches to **line**.

  In this situation, even though the setting for the controller clock is switched to loop-timed, the actual configuration remains **line**. This is the difference between the preconfigured state and the temporary "set state" of the controller.
- If either controller is the active clock source, the network clock PLL switch is thrown in the direction of the active clock. The system clock is recovered from the controller with the active clock source.
- If serial interface 0 is the active clock source, the clock source settings for both controllers are automatically set to **loop-timed,** and the network clock PLL switch is thrown in the direction of the serial port. The system clock is driven by a clock recovered from the DTE serial 0 interface, which has been multiplied from (n x 8000) Hz to 2 MHz.
- If the internal system clock is the active clock source, the clock source settings for both controllers are automatically set to **loop-timed,** and the network clock PLL switch is thrown in the direction of the controllers. Because both controllers are in the **loop-timed** state, neither clock provides a recovered clock to drive the PLL, resulting in a free-running, or internally timed, system clock.

The following is a configuration example showing a hierarchy of clock sources:

```
network-clock-select 1 t1 0
network-clock-select 2 t1 1
network-clock-select 3 serial0
network-clock-select 4 system
network-clock-switch 10 10

controller t1 0
 clock source line

controller t1 1
 clock source line

interface serial0
 clock rate line 64000
```

In this configuration, controller T1 0 is the primary clock source, and the clock source is configured to **line**. Controller T1 1 is a backup clock source and although the clock source is configured to **line**, the system temporarily sets the clock source to the **loop-timed** state.

If the controller T1 0 clock source fails, the system switches to use controller T1 1 as the clock source. The clock source **loop-timed** “set state” on controller T1 1 is switched to the preconfigured **line** state.