



Configuring TCP Header Compression

First Published: January 30, 2006
Last Updated: June 19, 2006

Header compression is a mechanism that compresses the IP header in a packet before the packet is transmitted. Header compression reduces network overhead and speeds up the transmission of either Real-Time Transport Protocol (RTP) or Transmission Control Protocol (TCP) packets.

Cisco provides two types of header compression: RTP header compression and TCP header compression. This module describes the concepts and tasks related to configuring TCP header compression.



Note

TCP header compression is configured on a per-interface (or subinterface) basis. If you want to configure TCP header compression on a per-class basis, see the “[Configuring Class-Based RTP and TCP Header Compression](#)” module.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the “[Feature Information for Configuring TCP Header Compression](#)” section on [page 15](#).

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Configuring TCP Header Compression, page 2](#)
- [Information About Configuring TCP Header Compression, page 2](#)
- [How to Configure TCP Header Compression, page 3](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2007 Cisco Systems, Inc. All rights reserved.

- [Configuration Examples for TCP Header Compression, page 10](#)
- [Additional References, page 12](#)
- [Glossary, page 14](#)
- [Feature Information for Configuring TCP Header Compression, page 15](#)

Prerequisites for Configuring TCP Header Compression

- Before configuring TCP header compression, read the information in the “[Header Compression](#)” module.
- You must configure TCP header compression on both ends of the network.

Information About Configuring TCP Header Compression

Before configuring TCP header compression, you should understand the following concepts:

- [TCP Header-Compression Keywords, page 2](#)
- [Maximum Compressed IP Header Size and TCP Header Compression, page 3](#)

TCP Header-Compression Keywords

When you configure TCP header compression, you can specify the circumstances under which the TCP packets are compressed and the format that is used when the packets are compressed. These circumstances and formats are defined by the following keywords:

- **passive**
- **iphc-format**
- **ietf-format**

These keywords (described below) are available with many of the quality of service (QoS) commands used to configure TCP header compression, such as the **ip tcp header-compression** command. For more information about the **ip tcp header-compression** command, these keywords, and the other QoS commands, see the [Cisco IOS Quality of Service Solutions Command Reference](#).

The **passive** Keyword

By default, the **ip tcp header-compression** command compresses outgoing TCP traffic. If you specify the **passive** keyword, outgoing TCP traffic is compressed only if *incoming* TCP traffic on the *same* interface is compressed. If you do not specify the **passive** keyword, *all* outgoing TCP traffic is compressed.

The **passive** keyword is ignored for PPP interfaces.

The **iphc-format** Keyword

The **iphc-format** keyword indicates that the IP Header Compression (IPHC) format of header compression will be used. For PPP and HDLC interfaces, when the **iphc-format** keyword is specified, RTP header compression is also enabled. Since both TCP and RTP header compression are enabled, both TCP and UDP packets are compressed.

The **iphc-format** keyword is not available for interfaces that use Frame Relay encapsulation.

**Note**

The header compression format (in this case, IPHC) must be the same at *both* ends of the network. That is, if you specify the **iphc-format** keyword on the local router, you must also specify the **iphc-format** keyword on the remote router.

The ietf-format Keyword

The **ietf-format** keyword indicates that the Internet Engineering Task Force (IETF) format of header compression will be used. For HDLC interfaces, the **ietf-format** keyword compresses only TCP packets. For PPP interfaces, when the **ietf-format** keyword is specified, RTP header compression is also enabled. Since both TCP header compression and RTP header compression are enabled, both TCP packets and UDP packets are compressed.

The **ietf-format** keyword is not available for interfaces that use Frame Relay encapsulation.

**Note**

The header compression format (in this case, IETF) must be the same at *both* ends of the network. That is, if you specify the **ietf-format** keyword on the local router, you must also specify the **ietf-format** keyword on the remote router.

Maximum Compressed IP Header Size and TCP Header Compression

With TCP header compression, you can configure the maximum size of the compressed IP header by using the **ip header-compression max-header** command.

The **ip header-compression max-header** command allows you to define the maximum size of the IP header of a packet to be compressed. Any packet with an IP header that exceeds the maximum size is sent uncompressed. For more information about the **ip header-compression max-header** command, see the [Cisco IOS Quality of Service Solutions Command Reference](#).

How to Configure TCP Header Compression

This section contains the following tasks:

- [Enabling TCP Header Compression on an Interface, page 3](#) (required)
- [Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation, page 5](#) (optional)
- [Changing the Maximum Size of the Compressed IP Header, page 7](#) (optional)
- [Changing the Number of Header-Compression Connections, page 7](#) (optional)
- [Displaying Header-Compression Statistics, page 9](#) (optional)

Enabling TCP Header Compression on an Interface

To enable TCP header compression on an interface, perform the following steps.

**Note**

To enable TCP header compression on an interface that uses Frame Relay encapsulation, skip these steps and complete the steps in the [“Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation”](#) section on page 5 instead.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation** *encapsulation-type*
5. **ip address** *ip-address mask* [**secondary**]
6. **ip tcp header-compression** [**passive** | **iphc-format** | **ietf-format**]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> • Enter the interface type and the interface number.
Step 4	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface. <ul style="list-style-type: none"> • Enter the encapsulation method.
Step 5	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 209.165.200.225 255.255.255.224	Sets a primary or secondary IP address for an interface. <ul style="list-style-type: none"> • Enter the IP address and mask for the associated IP subnet.

	Command or Action	Purpose
Step 6	<pre>ip tcp header-compression [passive iphc-format ietf-format]</pre> <p>Example: Router(config-if)# ip tcp header-compression ietf-format</p>	Enables TCP header compression.
Step 7	<pre>end</pre> <p>Example: Router(config-if)# end</p>	(Optional) Exits interface configuration mode.

Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation

To enable TCP header compression on an interface that uses Frame Relay encapsulation, perform the following steps.

Restrictions

The encapsulation type is specified by using either the **cisco** or **ietf** keyword of the **frame-relay interface-dlci** command. The **cisco** keyword specifies Cisco proprietary encapsulations, and the **ietf** keyword specifies IETF encapsulations. However, note the following points about these keywords:

- Frame Relay interfaces do not support IETF encapsulations when TCP header compression is enabled. Therefore, the **ietf** keyword is not available for Frame Relay interfaces and is not listed in the command syntax shown below.
- The **cisco** keyword is available for use on point-to-point subinterfaces *only*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **encapsulation frame-relay**
5. **ip address** *ip-address mask* [**secondary**]
6. **frame-relay interface-dlci** *dlci* [**cisco**]
7. **frame-relay ip tcp header-compression** [**passive**]
or
frame-relay map ip *ip-address dlci* [**broadcast**] **tcp header-compression** [**active** | **passive**]
[**connections** *number*]
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none">Enter the interface type and the interface number.
Step 4	encapsulation frame-relay Example: Router(config-if)# encapsulation frame-relay	Enables Frame Relay encapsulation.
Step 5	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 209.165.200.225 255.255.255.224	Sets a primary or secondary IP address for an interface. <ul style="list-style-type: none">Enter the IP address and mask for the associated IP subnet.
Step 6	frame-relay interface-dlci <i>dlci</i> [cisco] Example: Router(config-if)# frame-relay interface-dlci 20	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay interface on the router or access server. <ul style="list-style-type: none">Enter the DLCI number.
Step 7	frame-relay ip tcp header-compression [passive] Example: Router(config-if)# frame-relay ip tcp header-compression	Enables TCP header compression for all Frame Relay maps on a physical interface.
	or	
	frame-relay map ip <i>ip-address dlci</i> [broadcast] tcp header-compression [active passive] [connections <i>number</i>] Example: Router(config-if)# frame-relay map ip 10.108.175.200 190 tcp header-compression active	Assigns to an IP map header-compression characteristics that differ from the compression characteristics of the interface with which the IP map is associated. <ul style="list-style-type: none">Enter the IP address, DLCI number, and any optional keywords and arguments.
Step 8	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Changing the Maximum Size of the Compressed IP Header

By default, the maximum size of the compressed IP header is 168 bytes. When you configure TCP header compression, you can change this size to suit the needs of your network.

To change the maximum size of the compressed IP header, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip header-compression max-header** *max-header-size*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> • Enter the interface type and the interface number.
Step 4	ip header-compression max-header <i>max-header-size</i> Example: Router(config-if)# ip header-compression max-header 100	Specifies the maximum size of the compressed IP header. <ul style="list-style-type: none"> • Enter the maximum size of the compressed IP header, in bytes.
Step 5	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Changing the Number of Header-Compression Connections

For PPP and HDLC interfaces, the default is 16 compression connections. For interfaces that use Frame Relay encapsulation, the default is 256 compression connections.

To change the default number of header-compression connections, perform the following steps.

Implications of Changing the Number of Header-Compression Connections

Each header-compression connection sets up a compression cache entry, so you are in effect specifying the maximum number of cache entries and the size of the cache. Too few cache entries for the specified interface can lead to degraded performance, and too many cache entries can lead to wasted memory. Choose the number of compression connections according to the network requirements.

Restrictions

Header-Compression Connections on HDLC and Frame Relay Interfaces

For HDLC interfaces and Frame Relay interfaces (that is, interfaces that use Frame Relay encapsulation), the number of header-compression connections on *both sides* of the network must match. That is, the number configured for use on the local router must match the number configured for use on the remote router.

Header-Compression Connections on PPP Interfaces

For PPP interfaces, if the header-compression connection numbers on both sides of the network do not match, the number used is “autonegotiated.” That is, any mismatch in the number of header-compression connections between the local router and the remote router will be automatically negotiated to the lower of the two numbers. For example, if the local router is configured to use 128 header-compression connections, and the remote router is configured to use 64 header-compression connections, the negotiated number will be 64.



Note This autonegotiation function applies to PPP interfaces *only*. For HDLC interfaces and interfaces that use Frame Relay encapsulation, no autonegotiation occurs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number [name-tag]*
4. **ip tcp compression-connections** *number*
or
frame-relay ip tcp compression-connections *number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> [<i>name-tag</i>] Example: Router(config)# interface serial0	Configures an interface type and enters interface configuration mode. <ul style="list-style-type: none"> Enter the interface type and the interface number.
Step 4	ip tcp compression-connections <i>number</i> Example: Router(config-if)# ip tcp compression-connections 150	Specifies the total number of TCP header compression connections that can exist on an interface. <ul style="list-style-type: none"> Enter the number of compression connections. Note This command can be used for PPP interfaces, HDLC interfaces, or interfaces that use Frame Relay encapsulation.
	or	
	frame-relay ip tcp compression-connections <i>number</i> Example: Router(config-if)# frame-relay ip tcp compression-connections 150	Specifies the maximum number of TCP header compression connections that can exist on an interface that use Frame Relay encapsulation. <ul style="list-style-type: none"> Enter the number of compression connections. Note This command can be used for interfaces that use Frame Relay encapsulation <i>only</i> .
Step 5	end Example: Router(config-if)# end	(Optional) Exits interface configuration mode.

Displaying Header-Compression Statistics

You can display header-compression statistics, such as the number of packets sent, received, and compressed, by using either the **show ip tcp header-compression** command or the **show frame-relay ip tcp header-compression** command.

To display header-compression statistics, perform the following steps.

SUMMARY STEPS

- enable**
- show ip tcp header-compression** [*interface-type interface-number*] [**detail**]

or

```
show frame-relay ip tcp header-compression [interface type number]
```

3. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p>Example: Router> enable</p>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<pre>show ip tcp header-compression [interface-type interface-number] [detail]</pre> <p>Example: Router# show ip tcp header-compression</p>	Displays TCP/IP header compression statistics.
	or	
	<pre>show frame-relay ip tcp header-compression [interface type number]</pre> <p>Example: Router# show frame-relay ip tcp header-compression</p>	Displays Frame Relay TCP/IP header compression statistics for one or all interfaces.
Step 3	<pre>end</pre> <p>Example: Router# end</p>	(Optional) Exits privileged EXEC mode.

Configuration Examples for TCP Header Compression

This section provides the following configuration examples:

- [Enabling TCP Header Compression on an Interface: Example, page 11](#)
- [Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example, page 11](#)
- [Changing the Maximum Size of the Compressed IP Header: Example, page 11](#)
- [Changing the Number of Header-Compression Connections: Example, page 11](#)
- [Displaying Header-Compression Statistics: Example, page 12](#)

Enabling TCP Header Compression on an Interface: Example

In the following example, TCP header compression is enabled on serial interface 0.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# encapsulation ppp
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# ip tcp header-compression ietf-format
Router(config-if)# end
```

Enabling TCP Header Compression on an Interface That Uses Frame Relay Encapsulation: Example

In the following example, TCP header compression is enabled on serial interface 0. Frame Relay encapsulation has been enabled on this interface by using the **encapsulation frame-relay** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# encapsulation frame-relay
Router(config-if)# ip address 209.165.200.225 255.255.255.224
Router(config-if)# frame-relay interface-dlci 20
Router(config-if)# frame-relay ip tcp header-compression
Router(config-if)# end
```

Changing the Maximum Size of the Compressed IP Header: Example

In the following example, the maximum size of the compressed IP header (100 bytes) has been specified by using the **ip header-compression max-header** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip header-compression max-header 100
Router(config-if)# end
```

Changing the Number of Header-Compression Connections: Example

In the following example, the number of header-compression connections has been changed to 150 by using the **ip tcp compression-connections** command.

```
Router> enable
Router# configure terminal
Router(config)# interface serial0
Router(config-if)# ip tcp compression-connections 150
Router(config-if)# end
```

Displaying Header-Compression Statistics: Example

You can use the **show ip tcp header-compression** command to display header-compression statistics such as the number of packets received, sent, and compressed. The following is sample output from the **show ip tcp header-compression** command:

```
Router# show ip tcp header-compression serial10

TCP/IP header compression statistics:
Interface Serial10 (compression on, IETF)
  Rcvd:   53797 total, 53796 compressed, 0 errors, 0 status msgs
         0 dropped, 0 buffer copies, 0 buffer failures
  Sent:   53797 total, 53796 compressed, 0 status msgs, 0 not predicted
         1721848 bytes saved, 430032 bytes sent
         5.00 efficiency improvement factor
  Connect: 16 rx slots, 16 tx slots,
          1 misses, 0 collisions, 0 negative cache hits, 15 free contexts
          99% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

Additional References

The following sections provide references related to configuring TCP header compression.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference
Frame Relay	“Frame Relay Queueing and Fragmentation at the Interface” module
Header compression overview	“Header Compression” module
RTP header compression	“Configuring RTP Header Compression” module
Class-based RTP and TCP header compression	“Configuring Class-Based RTP and TCP Header Compression” module
IPHC profiles and header compression	“Configuring Header Compression Using IPHC Profiles” module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1144	<i>Compressing TCP/IP Headers for Low-Speed Serial Links</i>
RFC 2507	<i>IP Header Compression</i>
RFC 3544	<i>IP Header Compression over PPP</i>

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Glossary

compression—The running of a data set through an algorithm that reduces the space required to store the data set or the bandwidth required to transmit the data set.

DLCI—data-link connection identifier. A value that specifies a permanent virtual circuit (PVC) or switched virtual circuit (SVC) in a Frame Relay network. In the basic Frame Relay specification, DLCIs are locally significant (connected devices might use different values to specify the same connection). In the Local Management Interface (LMI) extended specification, DLCIs are globally significant (DLCIs uniquely specify individual end devices).

encapsulation—A method of wrapping data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit. Also, when dissimilar networks are bridged, the entire frame from one network is simply placed in the header used by the data link layer protocol of the other network.

full header (header refresh)—An uncompressed header that updates or refreshes the context for a packet stream. It carries a context identifier (CID) that will be used to identify the context. Full headers for non-TCP packet streams also carry the generation of the context that they update or refresh.

HDLC—High-Level Data Link Control. A bit-oriented synchronous data link layer protocol developed by the International Organization for Standardization (ISO). Derived from Synchronous Data Link Control (SDLC), HDLC specifies a data encapsulation method on synchronous serial links using frame characters and checksums.

header—A chain of subheaders.

IETF—Internet Engineering Task Force. A task force that consists of over 80 working groups responsible for developing Internet standards.

IPHC—IP Header Compression. A protocol capable of compressing both TCP and UDP headers.

PPP—Point-to-Point Protocol. A protocol that provides router-to-router and host-to-network connections over synchronous and asynchronous circuits.

regular header—A normal, uncompressed header. A regular header does not carry a context identifier (CID) or generation association.

subheader—An IPv6 base header, an IPv6 extension header, an IPv4 header, a UDP header, an RTP header, or a TCP header.

TCP—Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

UDP—User Datagram Protocol. A connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

Feature Information for Configuring TCP Header Compression

Table 1 lists the release history for this feature..

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

For information on a feature in this technology that is not documented here, see the “Header-Compression Features Roadmap” module.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring TCP Header Compression

Feature Name	Releases	Feature Information
This table is intentionally left blank because no features were introduced or modified in Cisco IOS Release 12.2(1) or a later release. This table will be updated when feature information is added to this module.	—	—

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco StadiumVision, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn is a service mark; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0804R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2007 Cisco Systems, Inc. All rights reserved.