CISCO™

Optical Networking Systems IP Management Solutions

Randy Zhang

shortcut

Your Short Cut to Knowledge

ciscopress.com

About the Author

Randy Zhang, Ph.D. and CCIE No. 5659, is a network consulting engineer at Cisco. He has authored many white papers on subjects relating to IP and optical technologies and spoke on optical timing and ONS IP management at Cisco Networkers conferences. His white paper on timing and synchronization has widespread readership. Randy is a co-author of the Cisco Press book *BGP Design and Implementation*, published in 2003.

About the Technical Editor

Lee Shombert, Ph.D., is a software engineer currently with the Cisco Gigabit Switching Business Unit. Prior to his position at GSBU, he was in the Cisco Optical Networking Group, where he designed and developed many of the IP networking features on the ONS 15454 platform, including the firewall, SOCKS proxy, and craft access features.

Dedication

To Susan, Amy, and Ally

Acknowledgements

I would like to express my deep gratitude to Lee Shombert, my technical reviewer. Lee's technical expertise helped keep the project on a sound technical footing. He was always available to answer my technical questions. His reviews of the original proposal and manuscripts greatly raised the quality of this Short Cut.

I would like to give special thanks to my editor, Brett Bartow. He patiently worked with me through the process in generating the proposal, arranging reviews, and keeping things in order and on schedule.

I want to thank all the reviewers who assisted in various stages of the project; their efforts resulted in a much improved final product. Rudy Davis and Tim Gimmel reviewed many chapters under a tight timeline. Tony Phelps, Faraz Shamim, John Skochenski, and Catherine Yan reviewed and provided comments on one or more chapters. Sam Chang, David Friedman, Tony Phelps, and John Skochenski reviewed and provided suggestions for the original proposal.

CHAPTER 1

Introduction

Cisco Optical Networking Systems (ONS) are a portfolio of products providing optical services to Time Division Multiplexing (TDM) traffic and packet-based traffic in access, metropolitan, and long-haul networks. The transport mechanisms can be Synchronous Optical Networks (SONET), Synchronous Digital Hierarchy (SDH), or Wavelength Division Multiplexing (WDM). Following the typical Cisco naming conventions, ONS products are in the 15xxx series, which includes 152xx, 153xx, 154xx, 155xx, and 156xx. This book focuses primarily on the 153xx, 154xx, and 156xx series products that share a common code base, with a special emphasis on the most widely deployed ONS 15454 platform. With regard to IP management, there is virtually no difference for platforms that support the common code.

NOTE
Incidentally, the first-generation controller card is also called TCC. Because this card has long been out of service, TCC is now a generic name that refers to all the shelf controller cards on ONS 15454.

The IP management functionality is performed by the shelf controller card. Table 1-1 shows a list of controller cards for ONS 15454, which is generically called Timing and Communications Controller, or TCC.

TABLE 1-1 Different Versions of the ONS 15454 Controller Card

Name	CPU	RAM	Required Software Releases	LAN Port Wiring Method
TCC	80 MHz	128 MB	2.3 or older	With Ethernet repeaters
TCC+	80 MHz	256 MB	2.2 to 4.1	With Ethernet repeaters
TCC-I	50 MHz	64 MB	3.3 to 4.0	With Ethernet repeaters
TCC2	400 MHz	256 MB	4.0 or later	With Ethernet repeaters
TCC2P	400 MHz	256 MB	4.0 or later (5.0 or later for secure mode)	With an Ethernet switch

NOTE
Secure mode is a feature provided by TCC2P and is discussed in Chapter 8, "Security Options." The LAN port wiring method refers to how individual LAN (Ethernet) ports are internally wired and is discussed later in this chapter.

This chapter introduces the basic concepts and terminologies for the rest of the book and builds a foundation for later topics. This chapter presents the following subjects:

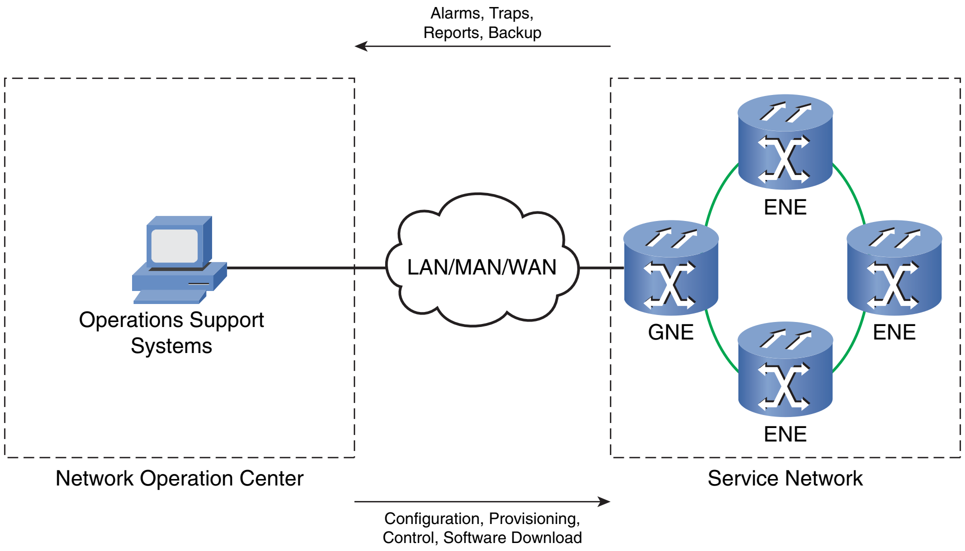
- Data Communication Network (DCN) architecture and components
- Basic IP configurations

- Interfaces and ports
- ONS management tools
- ML card management

Data Communication Network Architecture and Components

Data Communication Network (DCN) is the management network infrastructure that connects Operations Support Systems (OSS) and Network Elements (NE) so that messages can flow to accomplish Operations, Administration, Maintenance, and Provisioning (OAM&P). Figure 1-1 shows a high-level architecture and associated components.

FIGURE 1-1
DCN architecture and components


Alarms, Traps, Reports, Backup
Operations Support Systems
LAN/MAN/WAN
ENE
GNE
ENE
ENE
Network Operation Center
Service Network
Configuration, Provisioning, Control, Software Download

A DCN has three major components: the Network Operation Center (NOC), which houses the OSS; the Service Network, which contains all the NEs; and the LAN/MAN/WAN, which connects the two. The Service Network is where the client service is provided. Examples of client services are DS1, DS3, STS-3c, and Ethernet. Service Network components can be located in central offices, data centers, remote terminals, customer premises, and so on.

From the DCN perspective, NEs can be separated into two categories: Gateway NEs (GNE) and End or External NEs (ENE). ENEs use GNEs to reach OSS; in other words, GNEs provide the connectivity between OSS in the NOC and ENEs. As such, GNEs play a key role in the end-to-end NE management. Additionally, other devices, such as routers, might require the same kind of management connectivity (not shown in this figure), which will be discussed in later chapters.

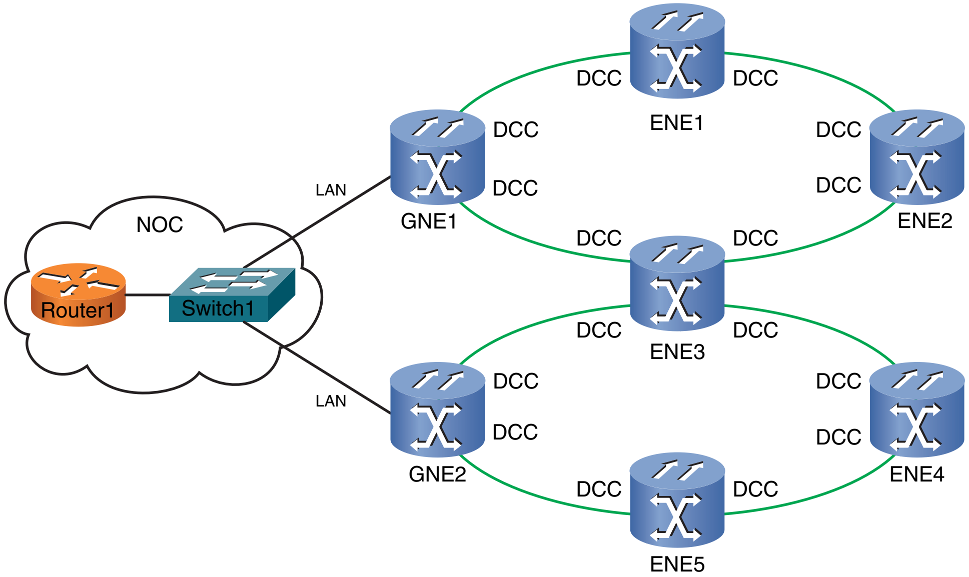
The connection between NOC and GNE is typically accomplished in an out-of-band fashion, a dedicated LAN connection for the specific purpose (as shown in Figure 1-1). Sometimes, such connections can be in-band due to the cost or availability of LAN connectivity. An in-band connection takes advantage of the Service Network circuits to connect a GNE to the DCN LAN. The disadvantage of the in-band solution is that the management connection is dependent on the Service Network. The failure of one or more links in the Service Network can bring down the in-band management connectivity. To protect against the impact of circuit failure, redundant circuits may be used.

The primary focus of this book is to examine the configurations of NEs and how NEs connect to the rest of the DCN. A more detailed view of that part of the DCN is shown in Figure 1-2.

In this view, two rings of NEs are connected via two GNEs to a NOC LAN switch. Two types of interfaces are relevant to an NE: LAN (Ethernet) and Data Communication Channel (DCC). In this figure, both GNEs have one LAN interface and two DCC interfaces, and all ENEs have only DCC interfaces.

What, precisely, is a GNE in networking terms? A GNE is an NE that has an active LAN interface and that can forward or relay traffic between the LAN interface and DCC interfaces. Note the word *can*. It means that a GNE is capable of performing such a function but does not always do so. For example, a standalone GNE without any DCC interfaces is still a GNE by this definition. An ENE is simply an NE that only forwards traffic between DCC interfaces. An ENE relies on a GNE to communicate with OSS in the NOC.

FIGURE 1-2
Detailed view of the NE connectivities



NOTE

The word *forward* means that the GNE is receiving traffic on one side and forwards the traffic to the other side. The word *relay* means that the GNE is performing a proxy function, relaying traffic between OSS and ENEs.

As you can see, the preceding GNE and ENE definitions are based solely on the actual role an NE is performing at a given time with regard to traffic flows. They are not necessarily a reflection of an NE's architectural position in the DCN or the intended role in traffic flows. You will see these definitions in action throughout the book. Based on these definitions, the following observations can be made:

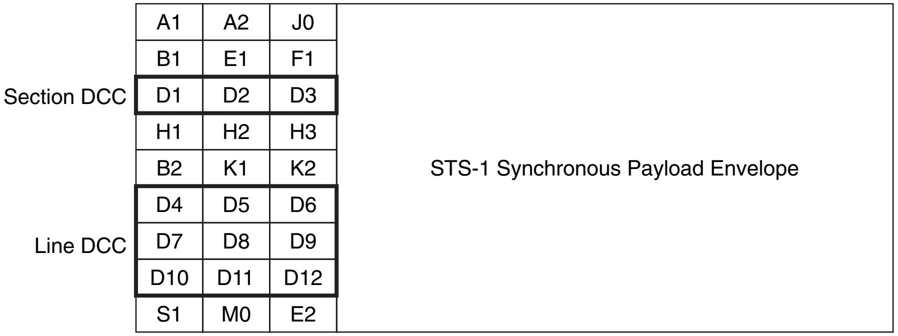
- A GNE stops being a GNE when its LAN interface is down.
- An NE with an active LAN interface that is not capable of forwarding or relaying traffic is not a GNE.
- An ENE can become a GNE when its LAN interface is activated.

The important point to keep in mind here is the status of the LAN interface and its impact on the GNE status. This point should become clearer to you as you progress.

DCC was introduced previously without any definition. DCC is simply an out-of-band path between a pair of SONET/SDH NEs. It is out-of-band because it does not use the SONET/SDH payload capacity, which is for client traffic. DCC is made of SONET/SDH overhead bytes and is provided as a clear channel for NE management purposes.

NOTE
Some may consider DCC an in-band channel because it shares with client data on the same physical link. From the service point of view, however, DCC is really out-of-band because that channel is not being used by the client data.

Figure 1-3 shows SONET Section and Line overhead bytes (SDH has similar overhead bytes), with special emphasis on DCC bytes. There are 3 bytes in Section DCC (SDCC) and 9 bytes in Line DCC (LDCC) in each STS-1 frame. The Synchronous Payload Envelope is used to transport client data.



A1	A2	J0
B1	E1	F1
D1	D2	D3
H1	H2	H3
B2	K1	K2
D4	D5	D6
D7	D8	D9
D10	D11	D12
S1	M0	E2

FIGURE 1-3
SONET DCC bytes

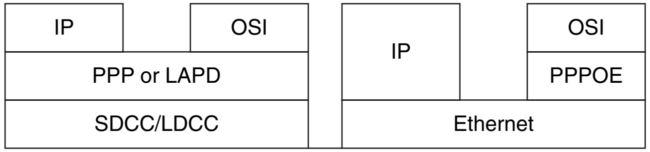
Because a SONET frame is transmitted every 125 microseconds, SDCC is rated at 192 kbps and LDCC at 576 kbps. In ONS, DCC bytes are terminated in an IO card and the content is extracted and passed to the shelf controller. Here are a few points to keep in mind about the use of DCC in ONS:

- Both SDCC and LDCC can be enabled on the same port, but it is not recommended; after LDCC is operational, SDCC should be removed.
- A finite number of DCC terminations are supported on a particular type of controller card (there are limits in terms of the total DCC terminations, SDCC terminations, and LDCC terminations), and every LDCC uses the space of three SDCCs.

- In a 1+1 protection, DCC can be provisioned only on the working port.
- There was a conflict between LDCC and BLSR in earlier releases (resolved in release 7.x).
- ONS can communicate over DCC with other vendors using IP, but some link information is not exchanged.
- General Communication Channel (GCC) is another form of DCC that is defined in ITU G.709 to allow communications between transponders.

To provide for end-to-end communication, proper networking protocols are implemented on each device. Two protocol stacks are supported in newer software releases of ONS: IP and OSI. Here, IP refers to IP version 4. Figure 1-4 shows an implementation of the protocol stacks on a GNE.

FIGURE 1-4
ONS protocol stacks on a GNE



Point-to-Point Protocol (PPP) is the Layer 2 protocol for IP over DCC. OSI uses Link Access Procedure D channel (LAPD) as the layer over DCC. On the LAN side, OSI uses PPP over Ethernet (PPPOE) as the Layer 2. Because this book is exclusively about IP management, no further discussion of the OSI protocol stack will be provided.

Basic IP Configurations

An ONS 15454 shelf has three LAN ports:

- One backplane port on the ANSI shelf for SONET or front-mount electrical connection (FMEC) port on the SDH shelf

- One RJ-45 port on the active TCC
- One RJ-45 port on the standby TCC

How these LAN ports are wired together depends on the wiring mode. All three ports are connected by repeaters except in TCC2P, where a LAN switch is used instead. Unlike other TCCs, TCC2P adds a secure mode operation (secure mode is discussed in Chapter 8). However, secure mode is off by default on TCC2P. When secure mode is off, TCC2P behaves like prior versions of TCC with regard to LAN port wiring. All ports are connected via repeaters, hence the repeater mode. Unless specifically indicated (as in Chapter 8), TCC is assumed to operate in repeater mode throughout this book.

In repeater mode, a MAC address/IP address pair (one MAC address and one IP address) is used for the entire shelf (a minor exception is the multi-shelf configuration, discussed next). The MAC address on ONS 15454 resides on the Auxiliary Interface Protection (AIP) board on the ANSI shelf and on the FMEC on the SDH shelf. One IP address is assigned to the entire shelf. The shelf comes with a factory default address. For example, the default IP address for ONS 15454 is 192.1.0.2.

Before the multi-shelf feature, introduced in release 7.0, an NE was a node housed in a single shelf. With the multi-shelf feature, a node can include one to eight shelves. Although the basic MAC and IP addressing and LAN port wiring method (repeater mode) are still true, a node controller (a master shelf) uses an autogenerated private IP address to communicate with other slave shelves in the same node. However, such intershelf communications do not affect the internode communications. For the purposes of NE to OSS communication, the multi-shelf feature has no impact. The multi-shelf configuration is discussed in detail in a case study.

There are three IP parameters when configuring an ONS node:

- IP address
- Subnet mask
- Default router

NOTE
The subnet of 192.168.190.0/24 cannot be used for node addressing because it is reserved for internal communications within the shelf.

There are many ways to assign addresses and subnets, which will have different implications for routing and traffic forwarding. Chapter 2, “Node Addressing Schemes,” walks through this subject at great length.

The default router field provides the router’s address on the attached LAN. The GNE uses this address to reach unknown networks in the DCN. In Chapter 4, “Static Routing,” you will learn different options to set the default router field and their implications for routing.

GNEs exchange Open Shortest Path First (OSPF) routes with ENEs, and they learn each other’s address dynamically. By default, the OSPF area for DCC links is 0 (the backbone area). Chapter 5, “Dynamic Routing,” introduces you to the details of OSPF routing on ONS.

Interfaces and Ports

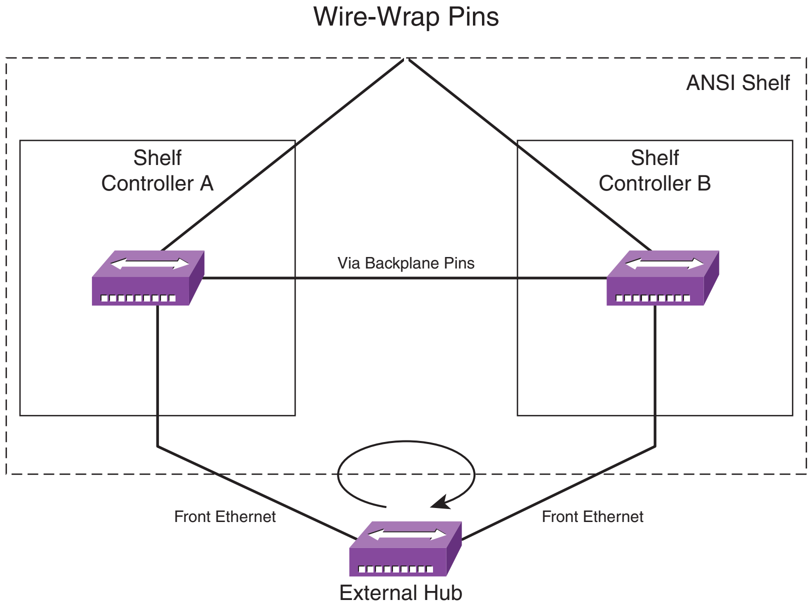
You learned previously that the LAN interface status affects the status of an NE on whether it is a GNE or an ENE. You also learned that an ONS can have multiple LAN ports on the same shelf. For example, an ONS 15454 shelf has three LAN ports. So how does the LAN interface relate to LAN ports?

An interface is a Layer 3 concept, and it can process IP packets; examples of this are a LAN interface or a DCC interface. A port is a physical concept. On an ONS, a LAN interface can have multiple physical LAN ports. Activating any of the LAN ports activates the LAN interface. The first active LAN port brings up the LAN interface. Additional active LAN ports do not change the status of the LAN interface. This is also true on the reverse. The LAN interface is up as long as there is at least one active LAN port. In contrast, a DCC interface is associated with exactly one SONET/SDH port.

In newer software releases, individual LAN ports can be disabled via software. You can disable both front LAN ports, the backplane port, or all LAN ports. Disabling a LAN port shuts down that port, and the port LED is off. This feature allows a quick change to the LAN interface or port status remotely. For example, disabling all the LAN ports of a GNE essentially changes the GNE into an ENE.

When multiple LAN ports are active and connected to the same repeater or hub domain, a repeater loop is formed. Figure 1-5 shows an example of a repeater loop.

FIGURE 1-5
Example of a repeater loop with an ONS 15454 shelf



In this example, the backplane port (via wire-wrap pins) is not active (the two connections from the shelf controllers to the wire-wrap port on the backplane are connected by a relay; when the backplane port is connected to an external device such as a hub, only one connection is active), but the two front LAN ports are connected to the same hub. As pointed out previously, each shelf controller is wired with a repeater, and the two repeaters are automatically connected over the backplane when two controllers are installed. As a result, Ethernet frames are repeated over and over, forming a repeater loop.

A repeater loop causes the frames to go on forever, until collisions occur or the entire domain is saturated. Symptoms of a repeater loop are LEDs flashing continuously, large numbers of collisions, and extremely slow response. To make matters worse, all other non-ONS devices connected to the same repeater domain are affected as well. Thus, a repeater loop should be avoided at all costs.

There are two common ways to avoid a repeater loop:

1. Connect only one of the three ports, such as the backplane port, to the external hub.
2. Use a LAN switch that supports spanning tree protocol (STP) if redundant LAN ports are needed.

One thing to remember with STP is that it has various timers and steps to go through when a port goes through status changes. If the forwarding port is disconnected, the other port goes through Blocking, Listening, Learning, and Forwarding stages. During each STP convergence, there may be about a 30-second period with no traffic being forwarded; in other words, there is no connectivity to the node during this period.

Table 1-2 shows all the common interfaces on an ONS 15454. There are three LAN interface names, depending on the controller type and configurations. The pdcc interface is sequentially named as pdcc0, pdcc1, and so on. The loopback interface is a unique interface that is automatically created by the software. A packet sent to this interface causes the packet to be looped back to the controller itself.

TABLE 1-2 Common Interface Names and Types on ONS 15454

Name	Type	Description
cpm0	LAN	An Ethernet (LAN) interface on controller cards older than TCC2.
motfcc0	LAN	An Ethernet (LAN) interface on TCC2 and TCC2P (it becomes the private interface when TCC2P is in secure mode).
motscc0	LAN	The front (public) interface on TCC2P in secure mode (not active when LAN ports are in repeater mode).
pdcc	DCC	A point-to-point unnumbered interface. An unnumbered interface is an IP interface that has no IP address assigned.
lo0	Loopback	A software interface internal to the controller to represent itself.

ONS Management Tools

A variety of tools exist to manage the ONS nodes. This section introduces Cisco Transport Manager (CTM), Cisco Transport Controller (CTC), a command-line management tool, and Simple Network Management Protocol (SNMP).

CTM is an element management system (EMS) for ONS and other Cisco devices. CTC is a craft management tool for ONS, which is designed to locally manage one or more NEs. Although both CTM and CTC can provide OAM&P, CTM offers enterprise-level capacity, high availability, and many other automated features. It is out of scope here to compare or contrast the two management tools. For the purposes of this book, I make no effort to distinguish the two unless it is specifically called for.

NOTE
ONS also provides a command-line shell interface, though it is really a debug tool rather than a management tool. The shell interface is a VxWorks interface, which provides low-level commands in almost every aspect of the node functionality. Because the shell interface is so powerful, it is not an interface designed for regular users of the equipment. Cisco strongly discourages its use unless directed by qualified personnel.

Beyond the graphical user interface (GUI) management tools mentioned previously, there is a popular command-line–based tool to provide limited management capability: the Transactional Language 1 (TL1) interface. The TL1 interface on ONS is an industry-standard interface that is primarily used for information retrieval and some parameter editing and creation. It is well suited for performing large numbers of tasks repetitively and efficiently because commands can be easily incorporated into a script.

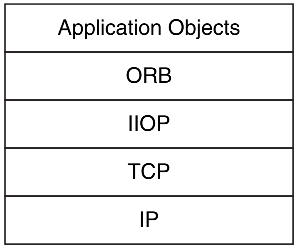
Because CTC is the most common management tool for ONS, the next few paragraphs provide an overview of CTC. CTC is a Java-based application that is downloaded dynamically from the NEs. Each software release works only with one or more Java Runtime Environments (JRE); therefore, the proper JRE version is important for CTC to load. Table 1-3 shows some of the ONS 15454 releases and JRE version compatibility.

TABLE 1-3 ONS 15454 Release and JRE Compatibility

JRE Version	Compatible ONS Releases
1.3	3.0–4.6
1.4	4.6–7.x
1.5	7.x or later

CTC communicates with NEs using Common Object Request Broker Architecture (CORBA) messages. CORBA is an open and vendor-independent architecture defined by Object Management Group (OMG), an industry consortium for middleware. CORBA is a client/server protocol—the client is CTC, and the server runs on the NE. Peers communicate on the same layer. Figure 1-6 shows a protocol stack for CORBA over TCP/IP.

FIGURE 1-6
CORBA protocol stack



To communicate down to the transport layer, General Inter-ORB Protocol (GIOP) is defined (ORB stands for Object Request Broker). The most important specialized mapping of GIOP is Internet Inter-ORB Protocol (IIOP), which passes requests or receives replies through the Internet's transport layer using the Transmission Control Protocol (TCP).

The following paragraphs describe the underlying mechanisms of CTC-NE communications. For a client to make a request of a program somewhere in a network, it must have an address for the program. This address is known as the Interoperable Object Reference (IOR). Using IIOP, part of the address is based on the server's port number and IP address. To establish a connection, a CORBA client uses the addressing information contained in IORs of the target object to contact a CORBA server, not the TCP/IP header information. This has connectivity implications when the underlying IP network uses address translation (more on this topic in Chapter 8).

The CTC-NE communication is a complicated process that involves many TCP ports and a variety of downloaded files. The following is a high-level process for CTC of recent releases. Actual files and steps may vary from release to release.

Assuming a fresh CTC is launched via a web browser to an NE, called the launch NE, the following general steps occur using CTC on a Windows-based workstation:

Step 1. The browser contacts the NE for the launch page. It then downloads and starts the launcher applet, and the browser window status shows Applet cerent.launcher.CtcLauncherApplet started. At this point, all CTC INI files are updated.

Step 2. The browser downloads the launcher.jar file (JAR stands for Java Archive). After downloading starts, a Java popup status bar is shown. At this point, the browser window refreshes, and the browser is not used any more.

Step 3. When the downloading is complete, a file named ctc-LAUNCHER*nnnn*.jar is created, where *nnnn* is a four-digit number.

Step 4. A file named ctc.bat is created, which is a batch file to load the launcher file in Step 3 using javaw; if the batch file exists, the old file is deleted. More on the content of the file will be discussed later.

Step 5. Download ORB JAR (with the filename ctc-ORB*nnnnn*.jar).

Step 6. Update the ctc.bat file with the locations of the launcher JAR file and the ORB JAR file.

Step 7. Several additional JAR files are downloaded.

Step 8. Start ctc.bat, and the CTC login window pops up. This batch file can be started manually any time to start the login window directly if the associated launcher file is present. It checks whether other JAR files are present. If not, they are downloaded. The ctc-log file is created. The user must enter the correct authentication information to continue.

Step 9. One or more JAR files are downloaded.

Step 10. The NE shelf view is displayed, and the CTC is fully functional.

If previous CTC cache exists, only a few steps are required:

Step 1. The browser starts the launcher applet. The browser window status indicates the following: Applet cerent.launcher.CtcLauncherApplet started.

Step 2. The ctc.bat is updated and started. The CTC login window pops up, and the ctc-log file is created. The user enters the correct authentication information to continue. The user can also start CTC without Step 1 by starting the ctc.bat file directly.

Step 3. The NE shelf view is displayed, and the CTC is fully functional.

There are two CTC INI files: CTC.INI and CTCLauncher.INI. The first file is a CTC preference file for GUI settings, and the second file is a launcher preference file. Clicking Delete CTC Cache in the browser deletes the content of this file and all the JAR files. Note that the cache is deleted only if no CTC session is active on the workstation.

Listing 1-1 shows a sample CTCLauncher.INI file for release 7.0. Here are a few points worth noting:

- The CTC version is indicated (7.00).
- The JRE version is indicated (1.4).
- The launcher JAR and its path are indicated.

LISTING 1-1 Sample CTCLauncher.INI File for Release 7.0

```
#CTC Preferences File
#Tue Sep 19 11:50:51 CDT 2006
ctc.launcher.CTC07.00.jre=C\:\\PROGRA~1\\Java\\J2RE14~1.2_1
ctc.launcher.ctc-jar-launcher=C\:\\Documents and Settings\\ranzhang\\Local Settings\\Temp\\
ctc-LAUNCHER28628.jar
```

Listing 1-2 shows a sample ctc.bat file for release 7.0. Here are a few points worth noting:

- The Java application launcher (or Java Virtual Machine) is javaw, which is the same as java, except javaw provides no console or online help.