9. Oracle Networking

Abstract: Networks have evolved from simple terminal-based systems to complex multi-tiered systems. Modern networks can be comprised of many computers on multiple operating systems using a wide variety of protocols and communicating across wide geographic areas. One need look no further than the explosion of the Internet to see how networking has matured and what a profound impact networks are having on the way we work and communicate with one another. While networks have become increasingly complex, they also have become easier to use and manage. For instance, we all take advantage of the Internet without knowing or caring about the components that make this communication possible because the complexity of this huge network is completely hidden from us. The experienced Oracle database administrator has seen this maturation process in the Oracle network architecture as well. From the first version of SQL*Net to the latest releases of Oracle Net, Oracle has evolved its network strategy and infrastructure to meet the demands of the rapidly changing landscape of network communications. This lesson highlights the areas that database administrators (DBAs) need to consider when implementing an Oracle network strategy. It also looks at the responsibilities the database administrator has when managing an Oracle network. The lesson then explores the most common types of network configurations and introduces the features of Oracle Net—the connectivity management software that is the backbone of the Oracle network architecture. It will also explore the Oracle network architecture and summarize the Oracle network infrastructure.

Contents

Objective:

- Explain solutions included with Oracle9i for managing complex networks.
- Describe Oracle networking add-on solutions.
- Explain the key components of the Oracle Net layered architecture.
- Explain Oracle Net Services role in client server connections.
- Describe how web client connections are established through Oracle networking products.

1. Network Design Considerations

There are many factors involved in making network design decisions. First and foremost is the design of the Oracle network architecture itself. It is flexible and configurable, and it has the scalability to accommodate a range of network sizes. Also, when you are working with an Oracle network, there are a variety of network configurations to choose from. The sections that follow summarize the areas that the DBA needs to consider when designing the Oracle network infrastructure.

1.1. Network Complexity Issues

The complexity of the network plays an important role in many of your network design decisions. Consider the following questions to determine network complexity:

- How many clients will the network need to support?
- What type of work will the clients be doing?
- What are the locations of the clients? In complex networks, clients may be geographically dispersed over a wide area.

- What types of clients are going to be supported? Will these be PC-based clients or terminal-based clients? Will these be thin clients that will do little processing or fat clients that will do the majority of the application processing?
- What is the projected growth of the network?
- Where will the processing take place? Will there be any middle-tier servers involved, such as an application server or transaction server?
- What types of network protocols will be used to communicate between the clients and servers?
- Will Oracle servers have to communicate with other Oracle servers in the enterprise?
- Will the network involve multiple operating systems?
- Are there any special networking requirements for the applications that will be used? This is especially important to consider when you are dealing with third-party applications.

1.2. Network Security Issues

Network security has become even more critical as companies expose their systems to larger and larger numbers of users through internets and intranets. Consider the following questions to determine the security of a network:

- Does the organization have any special requirements for secure network connections? What kinds of information will be sent across the Oracle network?
- Can you ensure secure connections across a network without risk of information tampering? This may involve sending the data in a format that makes it tamperproof and also ensures that the data cannot be captured and read by parties other than the client and the intended Oracle server.
- Is there a need to centralize the authorizations an individual has to each of the Oracle servers? In large organizations with many Oracle services, this can be a management and administration issue.

1.3. Interfacing Existing Systems with New Systems

The following issues should be considered when existing computer systems must communicate with Oracle server networks:

- Does the application that needs to perform the communication require a seamless, real-time interface?
- Does the existing system use a non-Oracle database such as DB2 or Sybase?
- Will information be transferred from the existing system to the Oracle server on a periodic basis? If so, what is the frequency and what transport mechanisms should be used? Will the Oracle server need to send information back to the existing system?

- Do applications need to gather data from multiple sources, including Oracle and non-Oracle databases, simultaneously?
- What are the applications involved that require this interface?
- Will these network requirements necessitate design changes to existing systems?

2. Network Responsibilities for the DBA

The database administrator has many design issues to consider and plays an important role when implementing a network of Oracle servers in the enterprise. Here are some of the key responsibilities of the DBA in the Oracle network implementation process:

- Understand the network configuration options available and know which options should be used based on the requirements of the organization.
- Understand the underlying network architecture of the organization in order to make informed design decisions.
- Work closely with the network engineers to ensure consistent and reliable connections to the Oracle servers.
- Understand the tools available for configuring and managing the network.
- Troubleshoot connection problems on the client, middle tier, and server.
- Ensure secure connections and use the available network configurations, when necessary, to attain higher degrees of security for sensitive data transmissions.
- Stay abreast of trends in the industry and changes to the Oracle architecture that may have an impact on network design decisions.

3. Network Configurations

There are three basic types of network configurations to select from when you are designing an Oracle infrastructure. The simplest type is the single-tier architecture. This has been around for years and is characterized by the use of terminals for serial connections to the Oracle server. The other types of network configurations are the two-tier, or client/server, architecture and the most recently introduced n-tier architecture. Let’s take a look at each of these configuration alternatives.

3.1. Single-Tier Architecture

Single-tier architecture was the standard for many years before the birth of the PC. Applications utilizing single-tier architecture are sometimes referred to as *green-screen applications* because most of the terminals using them, such as the IBM 3270 terminal, had green screens. Single-tier architecture is commonly associated with mainframe-type applications. This architecture is still in use today for many mission-critical applications, such as Order Processing and Fulfillment and Inventory Control, because it is the simplest architecture to configure and administer. Because the terminals are directly connected to the host computer, the complexities of network protocols and multiple operating systems don't exist.

When a single-tier architecture is being used, users interact with the database using terminals. These terminals are non-graphical, character-based devices. Figure 1 shows an example of the single-tier architecture. In this type of architecture, client terminals are directly connected to larger server systems such as mainframes. All of the intelligence exists on the mainframe, and all processing takes place there. Simple serial connections also exist on the mainframe. Although no complex network architecture is necessary, a single-tier architecture is somewhat limiting in terms of scalability and flexibility. Because all of the processing must take place on the server, the server can become the bottleneck to increasing performance.

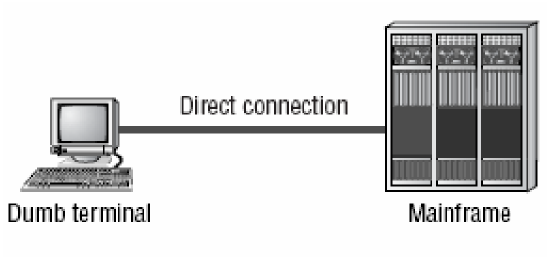

FIGURE 1. Single-tier architecture.

3.2. Two-Tier Architecture

Two-tier architecture gained popularity with the introduction of the PC and is commonly referred to as client/server computing. In a two-tier environment, clients connect to servers over a network using a network protocol, which is the agreed-upon method for the client to communicate with the server. TCP/IP is a very popular network protocol and has become the de facto standard of network computing. Whether TCP/IP or some other network protocol is chosen, both the client and the server must be able to understand the chosen protocol. Figure 2 shows an example of a two-tier architecture.

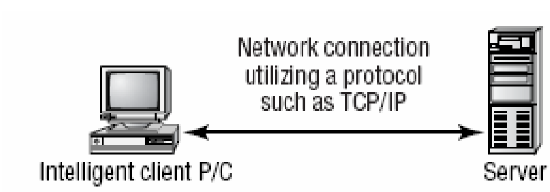

FIGURE 2. Two-tier architecture.

This architecture has definite benefits over single-tier architecture. First of all, client/server computing introduces the graphical user interface; this interface is easier to understand and learn, and it offers more flexibility than the traditional character-based interfaces of the single-tier architecture. Also, two-tier architecture allows the client computer to share the application processing load. To a certain degree, this reduces the processing requirements of the server.

The two-tier architecture does have some faults, even though at one time, it was thought to be the panacea of all networking architectures. Unfortunately, the main problem, that of scalability, persists. Notice that the term client/server computing contains a slash (/). The slash represents the invisible component of the two-tier architecture and the one that is often overlooked: the network!

When prototyping projects, many developers fail to consider the network component and soon find out that what worked well in a small environment may not scale effectively to larger, more complex systems. There was a great deal of redundancy in the two-tier architecture model because application software was required on every desktop. As a result of this scenario, many companies end up with bloated PCs and large servers that still do not provide adequate performance. What is needed is a more scalable model for network communications. That is what n-tier architecture provides.

3.3. N-Tier Architecture

N-tier architecture is the next logical step after two-tier architecture. Instead of dividing application processing work between a client and a server, you divide the work up among three or more machines. The n-tier architecture introduces *middleware* components, one or more computers that are situated between the client and the Oracle server, which can be used for a variety of tasks. Some of those tasks include the following:

- Moving data between machines that work with different network protocols.
- Serving as firewalls that can control client access to the servers.
- Offloading processing of the business logic from the clients and servers to the middle tier.
- Executing transactions and monitoring activity between clients and servers to balance the load among multiple servers.
- Acting as a gateway to bridge existing systems to new systems.

The Internet provides the ultimate n-tier architecture with the user's browser providing a consistent presentation interface. This common interface means less training of staff and also increases the potential reuse of client-side application components.

N-tier architecture makes it possible to take advantage of technologies such as networked computers. Such computers can make for economical, low-maintenance alternatives to the personal computer. Because much of the application processing can be done by application servers, the client computing requirements for these networked computers are greatly reduced. In addition, the processing of transactions can also be offloaded to transaction servers, which reduces the burden on the database servers.

The n-tier model is very scalable and divides the tasks of presentation, business logic and routing, and database processing among many machines, which means that this model accommodates large applications. In addition, the reduction of processing load on the database servers means that the servers can do more work with the same amount of resources. Also, the transaction servers can balance the flow of network transactions intelligently, and application servers can reduce the processing and memory requirements of the client (see Figure 3).

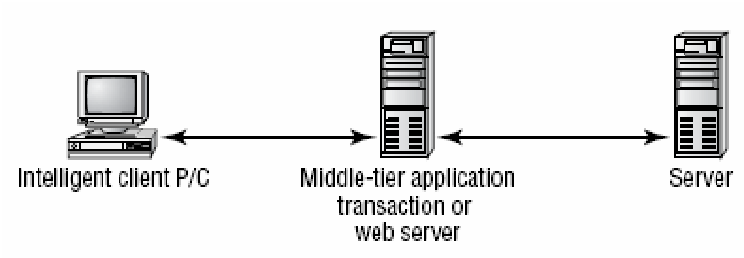

FIGURE 3. N-tier architecture.

4. Overview of Oracle Net Features

Oracle Net is the glue that bonds the Oracle network together. It is responsible for handling client-to-server and server-to-server communications, and it can be configured on the client, the middle-tier application, web servers, and the Oracle server.

Oracle Net also manages the flow of information in the Oracle network infrastructure. First, it is used to establish the initial connection to the Oracle server, and then it acts as the messenger, passing requests from the client back to the server or passing them between two Oracle servers. Basically, Oracle Net handles all negotiations between the client and server during the client connection. In the section entitled "The Oracle Net Stack Architecture" later on, we discuss the architectural design of Oracle Net.

In addition to functioning as an information manager, Oracle Net supports the use of middleware products such as Oracle9i Application Server (Oracle9i AS) and Oracle Connection Manager. These products allow n-tier architectures to be used in the enterprise, which increases the flexibility and performance of application designs.

4.1. Connectivity: Multi-Protocol Support

Oracle Net supports a wide range of industry-standard protocols including TCP/IP, IBM LU6.2, Named Pipes, and DECnet. (Unlike its predecessor Net8, Oracle Net no longer supports the Novell IPX/SPX protocol.) This support is handled transparently and allows Oracle Net to establish connectivity to a wide range of computers and a wide range of operating environments.

4.2. Connectivity: Multiple Operating Systems

Oracle Net can operate on many different operating system platforms, from Windows NT/2000, to all variants of Unix, to large mainframe-based operating systems. This range allows users to bridge existing systems to other Unix or PC-based systems, which increases the data access flexibility of the organization without making wholesale changes to the existing systems.

4.3. Connectivity: Java and Internet

With the introduction of Oracle8i, Oracle enabled connectivity to Oracle servers from applications using Java components such as Enterprise Java-Beans and Common Object Request Broker Architecture (CORBA), which is a standard for defining object interaction across a network. Oracle Net continues this trend by supporting standard connectivity solutions such as the Internet Inter-ORB Protocol (IIOP) and the General Inter-ORB Protocol (GIOP). These features allow clients to connect to applications interfacing with an Oracle database via a web browser. By utilizing features such as Secured Sockets Layer (SSL), client connections can obtain a greater degree of security across the Internet.

4.4. Directory Services: Directory Naming

Directory Naming allows for network names to be resolved through a centralized naming repository. The central repository takes the form of a Lightweight Directory Access Protocol (LDAP)–compliant server. LDAP is a protocol and language that defines a standard method of storage, identification, and retrieval of services. It provides a simplified way to manage directories of information, whether this information is about users in an organization or Oracle instances connected to a network. The LDAP server allows for a standard form of managing and resolving names in an Oracle environment. The quality of these services excels because LDAP provides a single industry standard interface to a directory service such as Oracle Internet Directory (OID). By utilizing Oracle Internet Directory, you ensure security and reliability of the directory information because information is stored in the Oracle database.

4.5. Directory Services: Oracle Internet Directory

The Oracle Internet Directory (OID) is an LDAP 3–compliant directory service, which provides the repository and infrastructure needed to enable a centralized naming solution using Directory Naming. OID can be used with both Oracle8i and 9i databases. In Oracle9i, the OID runs as an application.

The OID service can be run on a remote server and it can communicate with the Oracle server using Oracle Net. The OID is a scalable architecture, and it provides mechanisms for replicating service information among other Oracle servers.

OID also provides security in a number of ways. First of all, it can be integrated into a Secure Sockets Layer (SSL) environment to ensure user authentication. Also, an administrator can maintain policies that grant or deny access to services. These policies are defined for entities within the Oracle Internet Directory tree structure.

4.6. Scalability: Oracle Shared Server

Oracle Shared Server (formerly known as Multithreaded Server) is an optional configuration of the Oracle server that allows support for a larger number of concurrent connections without increasing physical resource requirements. This is accomplished by sharing resources among groups of users.

4.7. Scalability: Connection Manager

Oracle Connection Manager is a middleware solution that provides three additional scalability features:

Multiplexing Connection Manager can group together many client connections and send them as a single multiplexed network connection to the Oracle server. This reduces the total number of network connections the server has to manage.

Network access Connection Manager can be configured with rules that restrict access by IP address. This rules-based configuration can be set up to accept or reject client connection requests. Also, connections can be restricted by point of origin, destination server, or Oracle server.

Cross-protocol connectivity This feature allows clients and servers that use different network protocols to communicate. Connection Manager acts as a translator, providing two-way protocol conversion.

Oracle Connection Manager is controlled by a set of background processes that manage the communications between clients and servers. This option is not configured using the graphical Oracle Net Manager tool. Figure 4 provides an overview of the Connection Manager architecture.

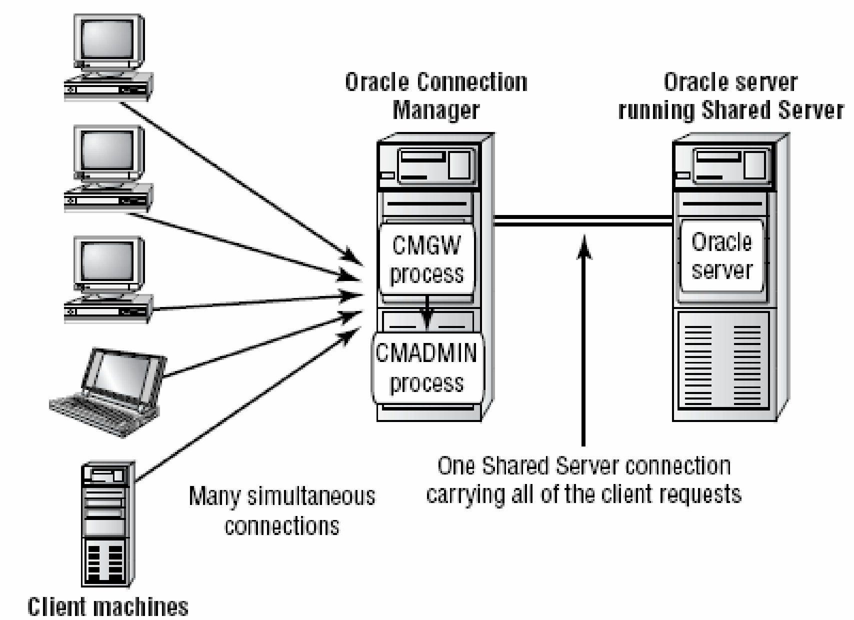

FIGURE 4. Connection Manager architecture.

4.8. Security: Advanced Security

The threat of data tampering is becoming an issue of increasing concern to many organizations as network systems continue to grow in number and complexity and as users gain increasing access to systems. Sensitive business transactions are being conducted with greater frequency and, in many cases, are not protected from unauthorized tampering or message interception.

Oracle Advanced Security not only provides the tools necessary to ensure secure transmissions of sensitive information, but it also provides mechanisms to confidently identify and authenticate users in the Oracle enterprise.

When configured on the client and the Oracle server, Oracle Advanced Security supports secured data transactions by encrypting and optionally checksumming the transmission of information that is sent in a transaction.

Oracle supports encryption and checksumming by taking advantage of industry-standard algorithms, such as RSA RC4, Standard DES and Triple DES, and MD5 checksumming. These security features ensure that data transmitted from the client has not been altered during transmission to the Oracle server.

Oracle Advanced Security also gives the database administrator the ability to authenticate users connecting to the Oracle servers. In fact, there are a number of authentication features for ensuring that users are really who they claim to be. These are offered in the form of token cards, which use a physical card and a user identifying PIN number to gain access to the system; the biometrics option, which uses fingerprint technology to authenticate user connection requests; Public Key; and certificate-based authentication.

Another feature of Oracle Advanced Security is the ability to have a single sign-on mechanism for clients. Single sign-on is accomplished with a centralized security server that allows the user to connect to any of the Oracle services in the enterprise using a single user ID and password. Oracle leverages the industry-standard features of Kerberos to enable these capabilities. (Kerberos is an authentication mechanism based on the sharing of secrets between two systems.) This greatly simplifies the privilege matrix that administrators must manage when they are dealing with large numbers of users and systems.

4.9. Security: Firewall Support

Firewalls have become an important security mechanism in corporate networks. *Firewalls* are generally a combination of hardware and software that are used to control network traffic and prevent intruders from compromising corporate network security. Firewalls fall into two broad categories:

> **IP-filtering firewalls** IP-filtering firewalls monitor the network packet traffic on IP networks and filter out packets that either originated or did not originate from specific groups of machines. The information contained in the IP packet header is interrogated to obtain this information. Vendors of this type of firewall include Network Associates and Axent Communications.
>
> **Proxy-based firewalls** Proxy-based firewalls prevent information from outside the firewall from flowing directly into the corporate network. Instead, the firewall acts as a gatekeeper, inspecting packets and sending only the appropriate information through to the corporate network. This prevents any direct communication between clients outside the firewall and applications inside the firewall. Check Point Software Technologies and Cisco are examples of vendors that market proxy-based firewalls.

Oracle works closely with the vendors of both types of product to ensure support of database traffic through these types of mechanism. Oracle supplies the Oracle Net Application Proxy Kit to the vendors of firewalls. This product can be incorporated into the firewall architecture to allow database packets to pass through the firewall and still maintain a high degree of security.

REAL WORLD SCENARIO

Know Thy Firewall

It is important to understand your network infrastructure, the network routes you are using to obtain database connections, and the type of firewall products you are using. I have had more than one situation in which firewalls have caused connectivity issues between a client and an Oracle server.

For instance, I remember what happened after a small patch was applied to a firewall when I was working as a DBA for one of my former employers. In this case, employees started experiencing intermittent disconnects from the Oracle database. It took many days of investigation and network tracing before we pinned down the exact problem. When we did, we contacted the firewall vendor and they sent us a new patch to apply that corrected the problem.

More recently, when I was working as a DBA for a large corporate client, the development staff started experiencing a similar problem. It turns out that the networking routes for the development staff had been modified to have connections routed through a new firewall. This firewall was configured to have a connection timeout after 20 minutes of inactivity, which was too short an amount of time for this department. As a result, we increased the timeout parameter to accommodate the development staff's needs.

These are examples of the types of network changes that a DBA needs to be aware of to avoid unnecessary downtime and to avoid wasting staff time and resources.

4.10. Accessibility: Heterogeneous Services

Heterogeneous Services provide the ability to communicate with non-Oracle databases and services. These services allow organizations to leverage and interact with their existing data stores without having to necessarily move the data to an Oracle server.

The suite of *Heterogeneous Services* is comprised of the *Oracle Transparent Gateway* and *Generic Connectivity*. These products allow Oracle to communicate with non-Oracle data sources in a seamless configuration. Heterogeneous Services also integrate existing systems with the Oracle environment, which allows you to leverage your investment in those systems. These services also allow for two-way communication and replication from Oracle data sources to non-Oracle data sources.

Transparent Gateway The Transparent Gateway product seamlessly extends the reach of Oracle to non-Oracle data stores, which allows you to treat non-Oracle data sources as if they were part of the Oracle environment. In fact, the user is not even aware that the data being accessed is coming from a non-Oracle source. This can significantly reduce the time and investment necessary to transition from existing systems to the Oracle environment. Transparent Gateway fully supports SQL and the Oracle transaction control features, and it currently supports access to more than 30 non-Oracle data sources.

Generic Connectivity Generic Connectivity provides a set of agents, which contain basic connectivity capabilities. It also provides a foundation so that you can custom build connectivity solutions using standard OLE DB, Microsoft's interface to data access. OLE DB requires an Open Database Connectivity (ODBC) driver to interface to the agents. You can also use ODBC as a stand-alone connection solution. For example, with the proper Oracle ODBC driver, you can access an Oracle database from programs such as Microsoft Excel. (These drivers can be obtained from Oracle or third-party vendors.) Because these drivers are generic in nature, they do not provide as robust an interface to external services as does the Transparent Gateway.

4.11. Accessibility: External Procedures

In some development efforts, it may be necessary to interface with procedures that reside outside of the database. These procedures are typically written in a third-generation language, such as C. Oracle Net provides the ability to invoke such external procedures from Oracle PL/SQL callouts. When a call is made, a process will be started that acts as an interface between Oracle and the external procedure. This callout process defaults to the name extproc. The listener is then responsible for supplying information, such as a library or procedure name and any parameters, to the called procedure. These programs are then loaded and executed under the control of the extproc process.

5. The Oracle Net Stack Architecture

The Oracle Net software is comprised of a series of programs that form a type of stack architecture. Each of these programs is responsible for handling various aspects of network communications, and each functions as a layer of the stack. This section discusses the architecture of the Oracle Net stack and defines the responsibilities of each portion. To successfully complete the OCP exam, you need to understand the structure and responsibilities of the Oracle Net stack. The structure and function of the Oracle Net stack is based on the Open Systems Interconnection (OSI) model.

5.1. The OSI Model

The *Open Systems Interconnection* (OSI) model is a widely accepted model that defines how data communications are carried out across a network.

There are seven layers in the OSI model, and each layer is responsible for some aspect of network communication. The upper layers of the model handle responsibilities such as communicating with the application and presenting data. The lower layers are responsible for transporting data across the network. The upper layers pass information, such as the destination of the data and how the data should be handled, to the lower layers. The lower layers communicate status information back to the upper layers. Table 1 shows the layers of the OSI model and the responsibilities each has in order for communications across a network to be executed. As you can see from this table, this layered approach allows for a separation of responsibilities. It also allows for the separation of the logical aspects of network communications, such as presentation and data management, from the physical aspects of communications, such as the physical transmission of bits across a network.

TABLE 1. The Layers of the OSI Model.

OSI Model Layer	Responsibilities
Application Layer	Interacts with the application. Accepts commands and returns data.
Presentation Layer	Settles data differences between client and server. Also responsible for data format.

Session Layer	Manages network traffic flow. Determines whether data is being sent or received.
Transport Layer	Handles interaction of the network processes on the source and destination. Error correction and detection occurs here.
Network Layer	Delivers data between nodes.
Data Link Layer	Maintains connection reliability and retransmission functionality.
Physical Layer	Transmits electrical signals across the network.

5.2. The Oracle Communications Stack

The OSI model is the foundation of the Oracle communications stack architecture. Each of the layers of the Oracle communications stack has characteristics and responsibilities that are patterned after the OSI model. Oracle interacts with the underlying network at the very highest levels of the OSI model. In essence, it is positioned above the underlying network infrastructure and communicates with the underlying network.

Oracle uses Oracle Net on the client and server to facilitate communications. The communications stack functions as a conduit to share and manage data between the client and server. The layers of Oracle communications stack are as follows:

- The application (client) layer
- The Oracle Call Interface (OCI) layer (client) or Oracle Program Interface (OPI) layer (server)
- The Two-Task Common (TTC) layer
- The Oracle Net Foundation layer
- The Oracle Protocol Adapters (OPA) layer
- The Network Specific Protocols layer
- The Network Program Interface (NPI for server-to-server communications only) layer

Figure 5 depicts the relationship of each of the layers of the stack on both the client and the server. The client process makes network calls that traverse down the Oracle Net client layers to the network protocol. The server receives the network request, processes it, and returns the results to the client.

The Application Layer (Client)

The application layer of the Oracle communications stack provides the same functionality as the Application Layer of the OSI model. This layer is responsible for interacting with the user, which involves providing the interface components, screen, and data control elements.

Interfaces such as forms or menus are examples of the application layer. This layer communicates with the Oracle Call Interface (OCI) layer.

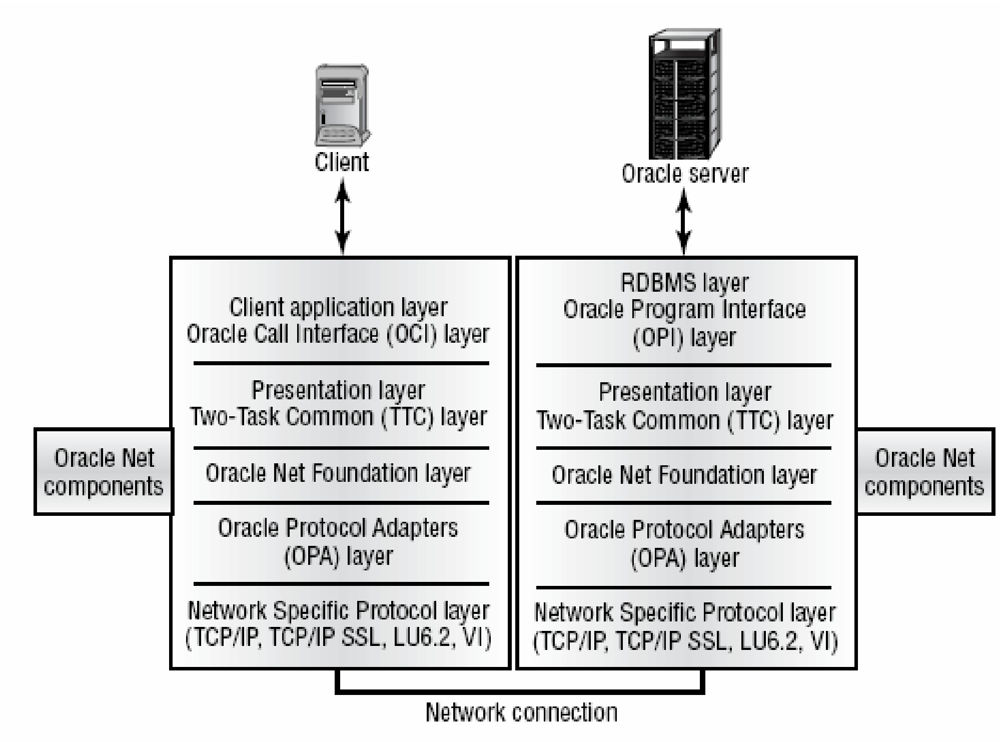

FIGURE 5. Oracle Net stack architecture.

The Oracle Call Interface (OCI) Layer (Client)

The *Oracle Call Interface* (OCI) layer is responsible for all of the SQL processing that occurs between a client and the Oracle server. The OCI layer exists on the client only. There is an analogous server component called the *Oracle Program Interface* (OPI) layer on the server. The OCI layer is responsible for opening and closing cursors, binding variables in the server's shared memory space, and fetching rows. Because the OCI is an open architecture, third-party products can write applications that interface directly with this layer of the communications stack. The OCI layer passes information directly to the Two-Task Common (TTC) layer.

The Two-Task Common (TTC) Layer

The *Two-Task Common* (TTC) *layer* is responsible for negotiating any datatype or character set differences between the client and the server. The Two-Task Common layer acts as a translator, converting values from one character set to another. The TTC will determine if any datatype differences are present when the connection is established. The TTC layer passes information to the Oracle Net Foundation layer. This layer shares some of the characteristics of the Presentation Layer from the OSI model.

The Oracle Net Foundation Layer

The *Oracle Net Foundation layer* (formerly known as the Transparent Network Substrate or TNS layer) is an integral component of the Oracle communications stack architecture, and it is analogous to the Session Layer in the OSI model. It is based on the Transparent Network Substrate (TNS), which allows Oracle Net to be a very flexible architecture, interfacing with a wide variety of network protocols. The TNS interface shields both the client and server from the complexities of network communications.

At this layer of the communications stack, Oracle Net interfaces with the other layers of the stack and their underlying protocols. It is this layer that provides the level of abstraction necessary to make Oracle Net a flexible and adaptable architecture, and it is this layer that compensates for differences in connectivity issues between machines and underlying protocols. This layer also handles interrupt messages and passes information directly to the Oracle Protocol Adapters (OPA) layer. The Oracle Net Foundation layer has several sublayers:

> **Network interface (NI) sublayer** The network interface sublayer provides a common interface on which the clients and servers can process functions. This layer of the stack is also responsible for handling any break requests.
>
> **Network routing (NR) sublayer** This is where Oracle Net keeps its network roadmap of how to get from the source or client to the destination or server.
>
> **Network naming (NN) sublayer** This layer takes network alias information and changes it into Oracle Net destination address information.
>
> **Network authentication (NA) sublayer** This layer is responsible for any negotiations necessary for authenticating connections.
>
> **Network session (NS) sublayer** The network session layer handles the bulk of activity in an Oracle network connection. This layer is responsible for such things as negotiating the initial connection request from the client. It is also responsible for managing the Oracle Net buffer contents and passing the buffer information between the client and the server. It also handles special features of connection management, such as buffer pooling and multiplexing, if these options are used.

The Oracle Protocol Adapters (OPA) Layer

The *Oracle Protocol Adapters* (OPA) *layer* interfaces with the underlying network. This becomes the entry point into the underlying network. This layer maps the Oracle Net Foundation layer functions to the analogous functions in the underlying protocol. There are different adapters for each protocol supported. This layer, in conjunction with the Network Specific Protocol layer, is analogous to the Network Layer in the OSI model.

The Network Specific Protocol Layer

This is the actual transport layer that carries the information from the client to the server. Some of the protocols supported by Oracle Net include TCP/IP, and DECnet. These protocols are not supplied with the Oracle software and must be in place to facilitate network communications.

The Oracle Program Interface (OPI) Layer (Server Only)

For every request made from the client, the Oracle Program Interface layer is responsible for sending the appropriate response. So when clients issue SQL statements requesting data from the database, the OPI interface fulfills that request. In server-to-server communication, the Network Program Interface (NPI) is used instead of the OPI.

The Network Program Interface (NPI) Layer (Server Only)

The *Network Program Interface* (NPI) *layer* is found only on the Oracle server. It is analogous to the OCI on the client. This layer is responsible for server-to-server communications, whereas the OPI is used in client-server communications and is used in distributed environments where databases communicate with each other across database links. This layer is analogous to the Presentation Layer of the OSI model, but from the server perspective.

5.3. Oracle Net and Java Support

To provide the ability to interface with Java code that can be written and deployed inside the Oracle server environment, Oracle Net supports the General Inter-ORB Protocol (GIOP). This protocol allows Object Request Brokers to talk to one another over the Internet. An Object Request Broker is a piece of software that handles the routing of object requests in a distributed network. The Internet Inter-ORB Protocol (IIOP) is a flavor of GIOP running under TCP/IP that supports the Secured Sockets Layer (SSL) for making secured network connections across the Internet. This type of connectivity would be used if an application were accessing Java procedures that were written and stored in the Oracle database. In this case, the Oracle9i Java Virtual Machine (JVM) would be running on the Oracle server, providing the Object Request Broker functionality.

The only portion of Oracle Net that is required is the Oracle Net Foundation layer. This streamlined communications stack allows for more efficient connectivity of Oracle servers when server-side Java procedures are being used. Figure 6 shows how the modifications streamline the Oracle communications stack.

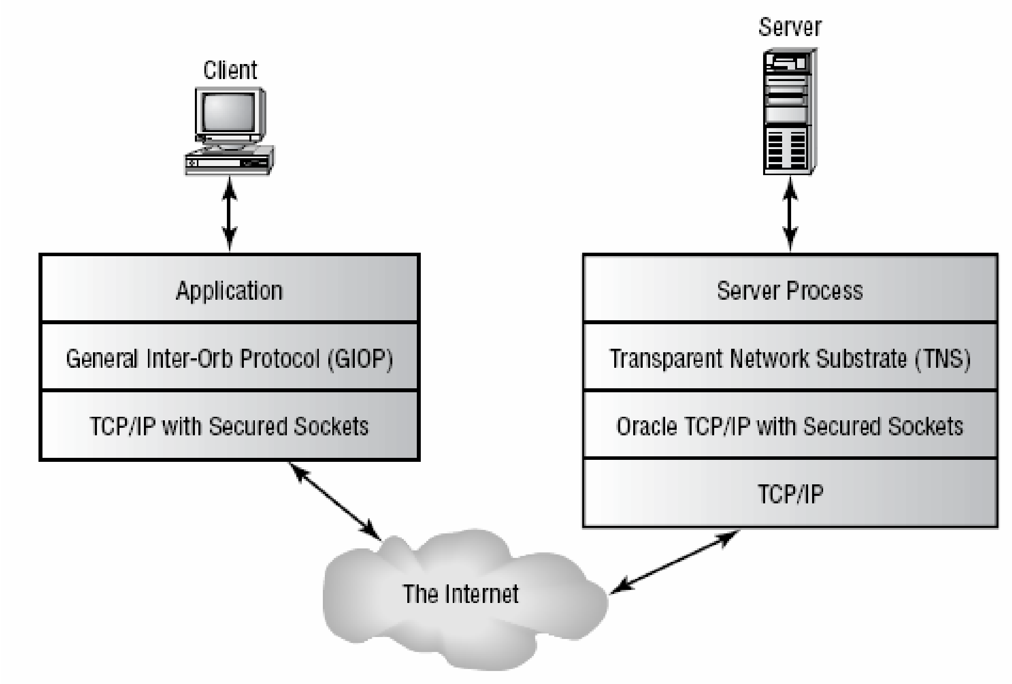

FIGURE 6. IIOP stack communications.

Oracle Net also provides robust connectivity for web-based applications. Connections to the Oracle server are available directly through Java applications, Java applets, or via an application server such as Oracle9iAS.

Internet Connectivity with Oracle Net

Connections initiated to an Oracle server via a web browser are much like client-server applications. The main difference is that the application server acts as the client, providing communications to and from the Oracle server. If a Java application resides on the web server, the Java Database Connectivity (JDBC) OCI driver is used to initiate communications to the Oracle server. (JDBC is an interface that allows Java programs to interact with data stored in tabular form, such as in an Oracle database.) If the connection is made via a Java applet, then the JDBC Thin Driver is used. The JDBC Thin Driver does not require Oracle client software to be installed, hence the term "thin driver." This driver provides the connectivity from the applet to the Oracle server.

Typically in this environment, an HTTP request is initiated at the client and sent to the application server. The application server forwards the request to the appropriate database service for processing. Oracle Net serves as the mechanism for communication between the application server and the database. HTTP is used to send the request to the application server and to receive the response from the application server. Figure 7 shows an example of web connections when using a Java application.

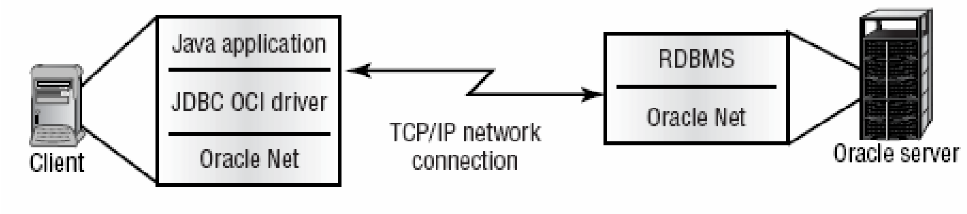

FIGURE 7. Web connectivity via a Java application.

Web Connections with No Application Server

A database can be configured to accept requests directly from a web environment without a middle-tier application server. This is because HTTP and IIOP can be configured on the Oracle server to accommodate these types of connections. Oracle supports the development and deployment of common Java objects, such as Enterprise JavaBeans (EJB), within the Oracle server itself. A client can make a request to the Oracle database and interface with these components directly. Oracle Net must be configured on the server to accept and process these types of requests. What makes this solution attractive is that no software needs to be deployed to the client; all the client needs is a web browser to interact with the database.

6. Summary

There are several key components that are necessary to understand in order to succeed when you are networking in an Oracle environment. The main responsibilities of the network administrator include determining the applications and type of connections that will be supported, the number of users and the locations from which they will be accessing the network, and the security issues involved in protecting sensitive information, such as single sign-on and data encryption.

In addition to being aware of their own responsibilities, the DBA needs to choose from the three basic types of network configurations when setting up their Oracle network: single-tier architecture, two-tier architecture, and n-tier architecture. Because systems have evolved from the simpler single-tier architecture to the more complex n-tier architecture, which can include connections through middle-tier servers and the Internet, database administrators will most likely find themselves choosing between the two architectures that Oracle Net is an integral part of: two-tier or n-tier architectures.

Oracle Net manages the flow of information from client computers to Oracle servers and forms the foundation of all networked computing in the Oracle environment. Oracle Net is comprised of a series of layers that make up the Oracle Net stack architecture. This architecture is based on the OSI model of networking and provides the basic building blocks of network interaction. Each layer in the Oracle Net stack is responsible for one or more networking tasks. Requests and responses are sent up and down the stack, which exists on both the client and the server.

In addition to the main network architecture that supports connections to an Oracle server, Oracle Net provides services that can be divided into five main categories: Connectivity, Directory Services, Scalability, Security, and Accessibility. Connectivity solutions include support for multiple protocols, multiple operating systems, and Java and Internet. Directory Services provide an infrastructure to resolve Oracle service names through a centralized naming repository. Scalability solutions include Connection Manager and Oracle Shared Server. Security options include Oracle Advanced Security, which provides an additional layer of security options and robust support for many varieties of firewalls. Accessibility support includes Heterogeneous Services and support for calling external procedures. Oracle Net also provides connectivity to Java stored procedures: the HTTP and IIOP protocols.

This lesson provides the foundation of knowledge that you will need to understand when you are designing an Oracle network infrastructure. The decisions you make about the network design have ramifications in terms of the scalability, security, and flexibility of your Oracle environment. When you understand the underlying network architecture and the network options available to you, you will be able to make informed choices when you are designing your Oracle network.

References

[1] Oracle9i DBA Fundamentals II.