

# NetWare Protocols

---

## Background

NetWare is a *network operating system* (NOS) and related support services environment created by Novell, Inc. and introduced to the market in the early 1980s. Then, networks were small and predominantly homogeneous; local-area network (LAN) workgroup communication was new; and the idea of a personal computer (PC) was just becoming popular.

Much of NetWare's networking technology was derived from *Xerox Network Systems* (XNS), a networking system created by Xerox Corporation in the late 1970s. For more information on XNS, see Chapter 22, "Xerox Network Systems."

By the early 1990s, NetWare's NOS market share had risen to between 50 and 75 percent. With over 500,000 NetWare networks installed worldwide and an accelerating movement to connect networks to other networks, NetWare and its supporting protocols often coexist on the same physical channel with many other popular protocols, including TCP/IP, DECnet, and AppleTalk.

## Technology Basics

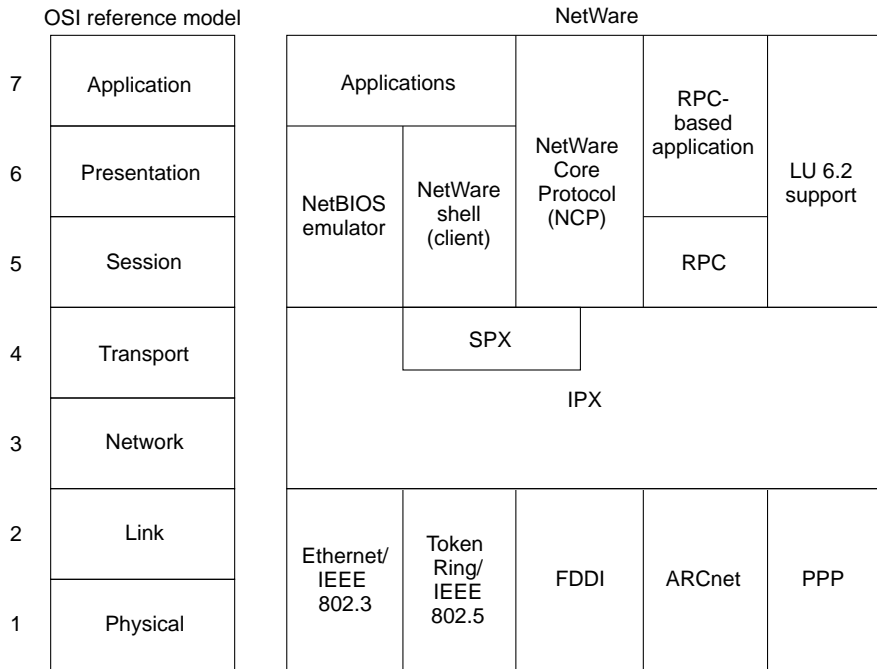
As a NOS environment, NetWare specifies the upper five layers of the OSI reference model. It provides file and printer sharing, support for various applications such as electronic mail transfer and database access, and other services. Like other NOSs such as the *Network File System* (NFS) from Sun Microsystems, Inc. and *LAN Manager* from Microsoft Corporation, NetWare is based on a *client-server architecture*. In such architectures, *clients* (sometimes called workstations) request certain services such as file and printer access from *servers*.

Originally, NetWare clients were small PCs, while servers were slightly more powerful PCs. As NetWare became more popular, it was ported to other computing platforms. Currently, NetWare clients and servers can be represented by virtually any kind of computer system, from PCs to mainframes.

A primary characteristic of the client-server system is that remote access is transparent to the user. This is accomplished through *remote procedure calls*, a process by which a local computer program running on a client sends a procedure call to a remote server. The server executes the remote procedure call and returns the requested information to the local computer client.

Figure 19-1 illustrates a simplified view of NetWare's best-known protocols and their relationship to the OSI reference model. With appropriate drivers, NetWare can run on any media-access protocol. The figure lists those media-access protocols currently supported with NetWare drivers.

Figure 19-1 NetWare and the OSI Reference Model



S1346a

## Media Access

NetWare runs on *Ethernet/IEEE 802.3*, *Token Ring/IEEE 802.5*, *Fiber Distributed Data Interface (FDDI)*, and *ARCnet*. For information on Ethernet/IEEE 802.3, see Chapter 5, “Ethernet/IEEE 802.3.” For information on Token Ring/IEEE 802.5, see Chapter 6, “Token Ring/IEEE 802.5.” For information on FDDI, see Chapter 7, “Fiber Distributed Data Interface.” NetWare also works over synchronous wide-area network (WAN) links using the *Point-to-Point Protocol (PPP)*. PPP is discussed in more detail in Chapter 9, “Point-to-Point Protocol.”

Attached Resource Computer Network (ARCnet) is a simple network system that supports all three primary cable types (*twisted-pair*, *coaxial*, and *fiber-optic*) and two topologies (*bus* and *star*). It was developed by Datapoint Corporation and introduced in 1977. Although ARCnet has not attained the popularity enjoyed by Ethernet and Token Ring, its low cost and flexibility have resulted in many loyal supporters.

## Network Layer

*Internet Packet Exchange (IPX)* is Novell’s original network-layer protocol. When a device to be communicated with is located on a different network, IPX routes the information to the destination through any intermediate networks. Figure 19-2 shows the IPX packet format.

**Figure 19-2 IPX Packet Format**

Checksum	
Packet length	
Transport control	Packet type
Destination network	
Destination node	
Destination socket	
Source network	
Source node	
Source socket	
Upper-layer data	

S1347

The fields of the IPX packet are as follows:

- *Checksum*—A 16-bit field that is set to 1s.
- *Packet length*—A 16-bit field that specifies the length, in bytes, of the complete IPX datagram. IPX packets can be any length up to the media maximum transmission unit (MTU) size. There is no packet fragmentation.
- *Transport control*—An 8-bit field that indicates the number of routers the packet has passed through. When the value of this field reaches 15, the packet is discarded under the assumption that a routing loop might be occurring.
- *Packet type*—An 8-bit field that specifies the upper-layer protocol to receive the packet's information. Two common values for this field are 5, which specifies *Sequenced Packet Exchange (SPX)*, and 17, which specifies the *NetWare Core Protocol (NCP)*.
- *Destination network, destination node, and destination socket*—Specify destination information.
- *Source network, source node, and source socket*—Specify source information.
- *Upper-layer data*—Contains information for upper-layer processes.

Although IPX was derived from XNS, it has several unique features. From the standpoint of routing, the encapsulation mechanisms of these two protocols are the most important difference.

*Encapsulation* is the process of packaging upper-layer protocol information and data into a frame. For Ethernet, XNS uses standard Ethernet encapsulation, whereas IPX packets are encapsulated in Ethernet Version 2.0 or IEEE 802.3 without the IEEE 802.2 information that typically accompanies these frames. Figure 19-3 illustrates Ethernet, standard IEEE 802.3, and IPX encapsulation.

---

**Note** NetWare 4.0 supports encapsulation of IPX packets in standard IEEE 802.3 frames. It also supports SubNetwork Access Protocol (SNAP) encapsulation, which extends the IEEE 802.2 headers by providing a type code similar to that defined in the Ethernet specification.

---

**Figure 19-3 Ethernet, IEEE 802.3, and IPX Encapsulation Formats**

Ethernet	Standard IEEE 802.3	IPX
Destination address	Destination address	Destination address
Source address	Source address	Source address
Type	Length	Length
Upper-layer data	802.2 header	IPX data
	802.2 data	
CRC	CRC	CRC

S1348a

To route packets in an internetwork, IPX uses a dynamic routing protocol called the *Routing Information Protocol (RIP)*. Like XNS, RIP derived from work done at Xerox for the XNS protocol family. Today, RIP is the most commonly used *interior gateway protocol (IGP)* in the Internet community, a large international network environment providing connectivity to virtually every research institution, government agency, and university and many private businesses in the United States, as well as many foreign organizations. For more detailed information on RIP, see Chapter 23, “Routing Information Protocol.”

In addition to the difference in encapsulation mechanisms, Novell also added a protocol called the *Service Advertisement Protocol (SAP)* to its IPX protocol family. SAP allows nodes that provide services (such as file servers and print servers) to advertise their addresses and the services they provide.

Novell also supports IBM Logical Unit (LU) 6.2 *network addressable units (NAUs)*. LU 6.2 allows peer-to-peer connectivity across IBM communication environments. Using NetWare’s LU 6.2 capability, NetWare nodes can exchange information across an IBM network. NetWare packets are encapsulated within LU 6.2 packets for transit across the IBM network.

## Transport Layer

*Sequenced Packet Exchange (SPX)* is the most commonly used NetWare transport protocol. Novell derived this protocol from the XNS *Sequenced Packet Protocol (SPP)*. As with the *Transmission Control Protocol (TCP)* and many other transport protocols, SPX is a reliable, connection-oriented protocol that supplements the datagram service provided by Layer 3 protocols.

Novell also offers *Internet Protocol (IP)* support in the form of *User Datagram Protocol (UDP)/IP* encapsulation of other Novell packets, such as SPX/IPX packets. IPX datagrams are encapsulated inside UDP/IP headers for transport across an IP-based internetwork. For more information on UDP and the Internet protocols in general, see Chapter 18, “Internet Protocols.”

## Upper-Layer Protocols

NetWare supports a wide variety of upper-layer protocols, but several are somewhat more popular than others. The *NetWare shell* runs in clients (often called workstations in the NetWare community) and intercepts application I/O calls to determine whether they require network access for satisfaction. If so, the NetWare shell packages the requests and sends them to lower-layer software for processing and network transmission. If not, they are simply passed to local I/O resources. Client

applications are unaware of any network access required for completion of application calls. *NetWare Remote Procedure Call* (NetWare RPC) is another more general redirection mechanism supported by Novell.

The *NetWare Core Protocol* (NCP) is a series of server routines designed to satisfy application requests coming from, for example, the NetWare shell. Services provided by NCP include file access, printer access, name management, accounting, security, and file synchronization.

NetWare also supports the *Network Basic Input/Output System* (NetBIOS) session-layer interface specification from IBM and Microsoft. NetWare's NetBIOS emulation software allows programs written to the industry-standard NetBIOS interface to run within the NetWare system.

NetWare application-layer services include *NetWare Message Handling Service* (NetWare MHS), *Btrieve*, *NetWare Loadable Modules* (NLMs), and various IBM connectivity features. NetWare MHS is a message delivery system that provides electronic mail transport. *Btrieve* is Novell's implementation of the binary tree (btree) database access mechanism. NLMs are implemented as add-on modules that attach into the NetWare system. NLMs for alternate protocol stacks, communication services, database services, and many other services are currently available from Novell and third parties.

