LonWorks Terminology

- http://www.lonworks.org.cn

Acknowledged Service
A service of the LonTalk® protocol ensuring a message was received by the addressee(s). After a configurable number of retries, failures are logged in a status register in the node that can be accessed by network management tools.

Actuator
Any component that affects a physical variable of the system under control or indicates the values of system variables for human operators. Some examples are pumps, fans, heaters, alarm annunciators, and operator displays. This term may apply only to the component that converts electrical or pneumatic control signals into a physical force that causes a mechanical component such as a damper or valve to move, or it may apply to both components as a unit.

Ad Hoc Configuration Scenario
A tool is used to design and install the network on-site while it loads the network configuration information into each device as the device is defined and configured and connections are created. The network database is built simultaneously. This is different from the engineered system scenario in that information is incrementally loaded to the physical devices. This scenario has the advantage of offering the most flexibility by letting the installer make decisions on-site. It is most appropriate for simpler systems in which the details of the system to be installed are not known prior to commissioning. *See also Engineered System Scenario.*

Address Table
A table on a Neuron Chip that is limited to 15 entries and defines the groups to which the LonWorks Device belongs and the destinations to which it sends bound network variables and explicit messages. The address table entry also contains information such as the transport properties i.e. retry count, timer values, etc.

Since the output network variable refers to the address table when it sends a message, multiple network variables can share an address table entry if both network variables are bound to the same target device(s) and the transport properties are the same.

When a downstream device's network variable polls an upstream device for its value, the downstream device is the initiator of the update, therefore it needs to know where the information is coming from and it must use an address table entry.

The contents of the address table are filled in by the NSS when it creates connections.

Alias
A network variable alias is a copy of a network variable containing the same value as the primary network variable but using a different selector. Since it has a unique selector, it allows for complex connections to be created.

Application / Application Program
Every device must contain an application. The application may be in a device when it's purchased or it may be loaded into the device from application files (.APB and .NXE extensions) using a network management tool. The application determines how a device functions.

The software code in a LonWorks device that implements the "personality" of the device. Also referred to as the application or the application layer, it resides in ROM or is downloaded over the network into non-volatile RAM. The application program interfaces with the LonTalk firmware to communicate over the network. It may reside completely in the Neuron Chip, or it may be split between the Neuron and an attached host processor (a host-based device).

Application Configuration
A process by which the application program in each node is tailored to the desired functionality by selecting the appropriate configuration parameters. LonWorks Network Services (LNS®) provides a platform for manufacturers to create easy-to-use graphical configuration interfaces, called plug-ins, that are then automatically compatible with any other LNS-based network tool.

Application Device
An application device is a LonWorks device that runs a LonTalk layer 7 application. The layer 7 application may run on a Neuron Chip, in which case the device is called a Neuron Chip-hosted device. The layer 7 application

may run on another processor, in which case the device is called a host-based device. Neuron Chip application programs are developed with the LonBuilder Developer's Workbench or the NodeBuilder tool and then exported to the application device. Some Application Devices use a Neuron Chip as a network interface, and execute their application programs on a host processor. In this case, the Neuron Chip in the network interface executes the LonBuilder Microprocessor Interface Program (MIP) to communicate with a LonWorks network as well as the host processor.

Applicationless Device
A device state where the device has no application image. Program or hardware failure may also cause a device to become applicationless.

Authentication
A service provided by the LonTalk protocol used to ensure that a received message was sent by an authorized source.

Backbone Network
A high-speed network channel connecting several lower speed channels.

BACnet
A specification for a protocol created by ASHRAE and used by some manufacturers to gain a degree of upper-level interoperability.

Binding
Binding is a process that takes place during network design and installation. The device firmware is configured to know the logical address of the other devices or group of devices in the network expecting that network variable, and it assembles and sends the appropriate packets to these devices. Similarly, when the device firmware receives an updated value for an input network variable required by its application program, it passes the data to the application program. The binding process thus creates logical connections between an output network variable in one device and an input network variable in another device or group of devices. Connections may be thought of as “virtual wires”.

The process that defines connections between LonWorks Devices. Connections define the data that devices share with one another.

Bit Rate
The rate in bits at which the packet frame is transferred across the communication medium.

Bound Monitoring
Creates an event-driven update connection between the network variable being monitored and the browser. Therefore the browser will report a change in network variable value as the value changes. To turn on bound updates, right-click a network variable in the browser and select Properties. Open the Monitoring tab and select Bind this network variable to the browser for receiving network variable updates. Bound monitoring can also be accomplished by binding network variables to the local network service device's virtual functional block. (See also Polling).

Communications Channel
The communications media that connect LonWorks devices or the path between devices that exhibits various physical characteristics. Different transceivers may be able to interoperate on the same channel. Segments connected via a physical layer repeater are considered a single channel. LonWorks Routers are used to connect two channels.

Channel Segment
A portion of a channel. A single channel can be comprised of multiple segments connected by physical layer repeaters.

Channel Type
Channels are categorized by channel type, and every type of transceiver must identify the channel type or types that it supports. The choice of channel type affects transmission speed and distance as well as the network ology.

Client
A task requesting service from a server. See also Remote Client.

Client ID
A unique identifier assigned to a client when the client is created. The NSS uses client IDs to track the source of each service invocation.

Client-Server Architecture
An architecture where a device (client) makes a request to another device or object (server) that delivers it.

Commissioning a Device
The process of using a network installation tool to download the network configuration data and application configuration data to a device. For devices whose application programs are not contained in ROM, the network tool also downloads the application program into non-volatile RAM memory in the device. Devices are usually either commissioned and tested one at a time or commissioned in off-line mode, then brought on-line and tested one at a time.

Communication Protocol
Rules and procedures governing transfer of information between devices on a network. The abbreviated term protocol is often used. The protocol defines the format of the message being transmitted between devices and defines the actions expected when one device sends a message to another. The protocol normally takes the form of embedded software or firmware code in each device on the network. The LonTalk protocol is defined in the US by the ANSI/CEA 709-1 standard and also defined in the EU by the EN 14908-1 standard.

Configuration Network Variable
A special class of network variable used to store network-modifiable application configuration data. Configuration network variables are always inputs. For Neuron Chip-hosted Application Devices, the contents of configuration network variables can be stored in the device's on-chip EEPROM, or off-chip EEPROM, flash, or NVRAM. For host applications, it is the responsibility of the host to store configuration values.

Configuration Properties
Applications may contain network variables and configuration properties. These are defined in the device template. Configuration properties are data structures specified by the LonMark guidelines that provide standards for documentation and for the network message formats used to download the customization data to the device by network tools. Configuration properties within a device are set during installation, operation, and maintenance to determine how the data is manipulated within the device. The application reads the values from the network variables and configuration properties and performs functions upon them. For example, an application may allow an arithmetic function (add, subtract, multiply, or divide) to be performed on two values received from two network variables. The function to be performed could be determined by a configuration property.

Configuration properties are used to configure the operation of a device or LonMark object. Configuration properties may be implemented using a special class of network variables called a configuration network variable, or they may be implemented as configuration parameters stored in a data block that is read and written using the LonTalk file transfer protocol or direct memory read/write.

Configured Device
A device state where the device has both an application image and a network image. This indicates that the device is ready for network operation.

Connection
The implicit addressing established during binding. A connection links one or more logical outputs (network variables or message tags) to one or more logical inputs.

Destination Address
The logical address contained in every LonTalk® packet of the node or group of nodes designated to receive the packet. The destination address can be the unique Neuron ID, the logical node address, a group address, or a broadcast address.

Device
Each device includes one or more processors that provide its intelligence and implement the protocol. Each device also includes a component called a transceiver to provide its electrical interface to the communications channel.

A device publishes information as appropriate to the application that it is running. The applications are not synchronized, and it is possible that multiple devices may all try to talk at the same time. Meaningful transfer of information between devices on a network, therefore, requires organization in the form of a set of rules and procedures.

Sensors, actuators, and controllers are examples of devices. *See also LonWorks Device.*

Device Discovery
Devices can be automatically discovered and commissioned using the Discover Devices command. This command allows the network management tool to search for devices on the network and match them to uncommissioned

devices defined within the network. This allows the rapid commissioning of networks that have been designed off-site as described in Engineered System Scenario.

Device Name
The name given to the device file when saving it. It is recommended that a descriptive name is supplied for each LonWorks device.

Device State
The state of the target device. The device state is displayed in the NodeBuilder device window and the Neuron C debugger status bar. A device may be in the following states: bypass, flush, offline, online, preemption, and sleep.

Device Template
A template used by network management tools that contains all the attributes of a given device type (functional blocks, network variables, configuration properties, etc. Different device templates cannot share the same program ID.

Disabling Devices
Disabling a device disables all LonMark objects on the device. A device must be Online to be Enabled or Disabled.

Disabling FBs
The functional block will send its configured output network variable defaults. The device containing the functional block must be Online to use this command. To be disabled, a device must contain a LonMark compliant Node Object functional block. If Disable is selected and the device does not contain a Node Object, the device is put into the Offline state.

Device Interface File (XIF File)
A file produced by the LonBuilder Developer's Workbench or the NodeBuilder tool that documents a device's external interface (XIF). After converting the text version of the external interface file or XIF to binary using the utility, the host application can import external interface file definitions into the NSS using a set of services. The text version has an extension of XIF, the binary file has an extension of XFB.

The XIF file should not be be modified manually but it can be viewed using an application such as Notepad. Since the XIF file was originally generated from the application and matches the XIF within the device, changing the XIF will result in unpredictable behavior.

This file tells network-management tools how to communicate and recognize a device. It can be obtained from the device manufacturer’s website, it is included with the shipped product, it can be uploaded from the device (though without all of the file content), or it can be obtained from the LonMark site for each certified and published device. It contains Program ID, NVs, CPs, functional blocks, etc. If the XIF is version 4 or later, it will contain default configuration properties.

Domain
A logical collection of devices on one or more channels. Communications can only take place among devices configured in the same domain.

Domain ID
The level of the LonTalk addressing hierarchy of domain/subnet/node. The domain ID can be 0, 1, 3, or 6 bytes long. The zero length domain is reserved for the use of the LNS architecture and cannot be used as the system's domain.

Downlink
Data transfer from the host toward the network and the NSI.

Download
An installation process in which data – such as the application program, network configuration, and/or application configuration – is transferred over the network to a device by a network management tool.

Downstream Device
The device receiving a network variable update.

Dynamic Data Exchange (DDE)
A standard Microsoft Windows® protocol that defines a mechanism for Windows applications to share information with one another. When applications share information with each other using DDE, they are said to be holding a DDE conversation. Each conversation has a well-defined beginning, middle, and end. To begin a conversation,

one application, known as the client or destination application asks another application, known as the server or source application to open a communications channel.

Once a conversation is established, the client can send and receive data from the server on the DDE channel. Note that the destination application is the one that establishes the conversation, irrespective of which way the data actually flows.

EEPROM
Electrically erasable programmable read-only memory. There are a limited number of write actions to EEPROM for a given controller.

Enabling Devices
Enabling a device enables all LonMark objects on the device. A device must be Online to be Enabled or Disabled.

Enabling FBs
Activates the functional block. This command requires LonMark-compatible support for this operation in the device's application. The device containing the functional block must be Online to use this command.

Engineered System Scenario
The engineered system installation scenario allows the network to be designed without being connected to the physical network. Using Device Templates, devices can be added to one or multiple subsystems, configuration can be performed using plugins and binding can be created. Once the physical network is attached, these devices are associated with the physical devices through a simple commissioning step and all the configuration information is loaded into the devices and routers.

This is a common method for network design when the devices, configuration and programming is known before hand.

Event
The mechanism that the NSS uses to inform a LNS host application of specific network activity, such as the arrival of a service pin message, or a change of network addresses. The LNS host application subscribes to and ss event notification using services.

Explicit Addressing
A form of messaging in which the device's application manually constructs messages and manually assigns an address to them.

Explicit Message
Low-level messages that Application Devices use to communicate with one another. Each message contains a message code that identifies the type of message. Application devices use the codes to determine the action to take when the message is received. When using explicit messages, the device is responsible for building, sending, and responding to messages.

Fan-In Connection
A connection where the outputs on multiple devices are directed to a single input on another device.

Fan-Out Connection
A connection where the output on a single device is directed to an input on multiple other devices.

FB (See Functional Block)
Firmware
Firmware is programming inserted into programmable read-only memory (programmable ROM), thus becoming a permanent part of a computing device. Firmware is created and tested like software (using microcode simulation). When ready, it can be distributed like other software and, using a special user interface, installed in the programmable read-only memory by the user.

FMSI (Facility Master System Integrator)
An umbrella position within standard construction divisions to help oversee the specification and implementation of DDC systems. The FMSI is usually accountable for assuring Interoperability between sub-systems and different buildings, for providing a common Graphical User Interface and for assuring product from multiple bidder and vendors meets the intent of a specification as well as the letter of a specification and acts as a technical go-between for th various involved sub-trades (Controls, Electrical, Mechanical etc.).

Free ology

A connection scheme for the communication bus that removes traditional transmission line restrictions of trunks and drops of specified lengths and at specified distances, and terminations at both ends. Free ology allows wire to be strung from any point to any other, in bus, daisy chained, star, ring, or loop ologies, or combinations thereof. It only requires one termination anywhere in the network. This can reduce the cost of wiring by a factor of two or more.

Full Client (see Remote Full Client or Local Client)

Functional Block

Applications in devices are divided into one or more functional blocks. A functional block is a collection of network variables and configuration properties, which are used together to perform one task. To define multiple functional blocks within a device, the device must contain a LonMark Node Object functional block, as defined by the LonMark Application-Layer Interoperability Guidelines.

Gateway Device

A LonWorks device that allows proprietary legacy control systems to be interfaced to LonWorks systems. A gateway device has a physical interface appropriate to the foreign system device or communication bus. Its application program interfaces to the proprietary communication protocol for the foreign system, translates between the two protocols as required, and converts the proprietary command-based messages of the foreign system to SNVTs used by the information-based LonWorks applications.

Group

A logical collection of devices within a domain. Unlike a subnet, devices are grouped together without regard for their physical location in the domain. The number of groups to which a device may belong is determined by the number of available address table entries on it. This number is set by the Neuron application, but may not exceed 15. Groups and group membership are defined by the NSS during binding.

Group Addressing

A logical addressing mode in the LonTalk® protocol that allows a message to be sent simultaneously to a pre-configured group of devices. Each group has an 8-bit group ID. Each domain can have up to 256 groups defined.

Group ID

A number used to identify a group. Each group is assigned a unique ID from 0 to 255 by the NSS.

Group Member Number

Within groups that use acknowledged message service, each member of the group is assigned a group member number. Devices use their member number to determine if reminder messages indicate that their acknowledgment or response was already received.

HMI (See Human-Machine Interface)

Host

A device implementing layer 7 of the LonTalk protocol. A host may be based on the Neuron Chip, in which case it is called a Neuron Chip Hosted Device. A host may be based on another processor, in which case it is called a host-based device. A host-based device uses the Neuron Chip as a network interface to talk to the LonWorks network.

Host-Based Device

A devices in which layer 7 of the LonTalk protocol runs on a processor other than the Neuron Chip.

Host Network Variables

A variable managed by the computer that contains the LNS Network Interface. Bound updates to host NVs allow the computer to collect network data with greater efficiency than polling.

Host Processor

The application processor for a host-based device; typically a micro-controller, microprocessor, or computer.

Hub

The center of a connection, specified by node handle and network variable index or message tag index. Each connection is defined in terms of a hub and a set of items that connect to the hub. The hub must be either the only input or the only output in the connection. For example, if the hub is an output network variable, all the other members in the connection must be input network variables.

Human-Machine Interface (HMI)

An HMI is used to provide an interface for operator input and output. For example, in a LonWorks network that controls a building's heating system, the HMI could include real-time temperature readings from each room, as well

as an override switch to allow the operator to manually control the heaters and fans. It is not designed to replace high-end HMI tools.

Implicit Addressing
A form of messaging in which the Neuron Chip firmware builds and sends network variable update and explicit messages using information contained in tables in its EEPROM. Implicit addressing is established during binding.

I/O Interface
An electrical interface from a LonWorks device – such as voltage, current, or contact closure – to a non- LonWorks sensor or actuator. The I/O interface can be digital (on/off), analog, or a communication protocol.

Infrastructure Devices
Generally includes the devices of the Infrastructure Subsystem, i.e. routers, repeaters, terminators, power supplies and the LNS® Network Interface.

Infrastructure Subsystem
The top-level subsystem that is the central place for managing routers, other subsystems, channels and the LNS Network Interface.

Input Network Variable
A network variable that provides information to the device from some other devices on the network.

Interoperability
A condition that ensures that multiple devices (from the same or different manufacturers) can be integrated into a single network without requiring custom device or tool development.

There are a number of benefits to using interoperable devices:

- Project engineers can use the best-of-breed systems
- Interoperable products give manufacturers a chance to compete in systems that would otherwise be closed
- Engineering teams can build to a standard specification
- Building, Factory and Plant managers can monitor values using standard tools regardless of the company that manufactured the devices.

Layer
An aspect of a drawing that can be turned on or off. Each drawing contains three layers: physical, logical and data whose default setting is “View” or being displayed. The physical layer shows all devices, routers, channels, and subsystems on a page. The logical layer contains all functional blocks, message tags, network variables, unmonitored connections, and subsystems on a page. The data layer contains all monitored connections on a page. Click on a given button in the toolbar to show or hide the associated layer. Hiding layers that aren’t required for viewing allows the user to reduce the clutter in a drawing.

LCA (see LonWorks Component Architecture)

LCA Data Server API
A standard LCA component that provides high performance monitoring and control. Using the data server, client applications can observe the values of network variables and explicit messages and can change the values of network variables or send explicit messages to effect the operation of the network. The data server supports both bound and unbound monitoring (using polling) and, optionally, filters redundant updates so that only changes in a variable’s value are reported to the application. To simplify client applications, the data server optionally converts raw network data into formatted text strings, which can be directly displayed.

LCA Field Compiler API
An optional LCA component, which can be used to build tools that can be used to reprogram devices in the field. This API consists of Dynamic Link Libraries for a Neuron C compiler, assembler, linker, exporter, and debugger.

LCA Object Server ActiveX Control
An ActiveX control that converts LNS objects (e.g., devices, routers, channels) managed by the NSS for Windows engine into standard OLE objects. The object server also provides a network tool kernel that enables sharing of information and objects between multiple tools and components. The kernel also provides an extensible host database that contains host-specific data not managed by the NSS as well as application-specific data.

Legacy I/O Device

A sensor or actuator, which cannot directly attach to a LonWorks network.

Lightweight Client (see Remote Lightweight Client)

Link-Powered Device
A device that is powered by a central power supply connected to the network. This power supply is typically shared by several devices on the network, eliminating the need for a power supply at each device. The power is supplied over the same medium as the communication signals.

LNS® (LonWorks Network Services)
Echelon Corporation's LNS provides directory, installation, management, monitoring, and control services required for open LonWorks networks. LNS is a platform that allows multiple LNS applications to interoperate on the same personal computer (PC) or on multiple PCs on the same network. *See also LonWorks Network Services.*

LNS is a client-server operating system with a single LNS Server that supports many interoperating client applications. The LNS Server can run as a standalone application on a PC attached to the network, or it can run on the same PC as the Network Management tool. Clients on other PCs (called remote clients) can log into the LNS Server to access the shared LNS database.

Since API networks operate as a peer-to-peer network and use a different database structure (vs. client-server), the networks are incompatible. This implies that API products cannot be commissioned using an LNS tool and visa versa.

Network Management tools use the client-server capabilities of LNS to allow multiple Network Management tools running on different PCs to simultaneously access the same LNS Server. This capability allows multiple users to work at the same time on a single network.

LNS Client
Any application that uses the services of the LNS Server.

LNS Host Application
A host application that uses an NSI as its network interface. An LNS host application can make use of the services, events, and properties provided by an NSS to perform network installation, configuration, maintenance, repair, monitoring, and control. A LNS host application can also implement its own application-specific services, events, and properties and, through the LNS architecture, make these available to other LNS host applications.

Load Status
The application state of the target device. The load status is indicated in the NodeBuilder Device Window and the Neuron C debugger status bar. A device may be in one of the following load states: applicationless, configured, or unconfigured.

Local Client or Local Full Client
A Network Management tool running on the same PC as the LNS Server. The simplest Local Client configuration, a local application is where the Network Management Tool and LNS Server PC is directly connected to the LonWorks Channel.

Local IP Client
The other configuration for a local client is as a Local IP client.

LonMark®
A distinctive logo applied to LonWorks devices that have been certified to the interoperability guidelines of LonMark International.

LonMark Affiliate
A Regional or National chapter affiliated with LonMark International.

LonMark Certified
A LonMark certified device has a program ID type 8 (LonMark certified) or 9 (non- LonMark certified). In these cases each functional block will correspond to a LonMark object. A non-LonMark certified device will contain just one functional block, known as a virtual functional block, which represents all of the network variables and configuration properties on the device.

LonMark Device
A LonWorks device that has been certified to meet the interoperability standards of LonMark International.

LonMark Functional Profile

A LonMark functional-block definition or template. Such "Profiles" are designed for specific application areas, such as HVAC or lighting systems. An example is the VAV Controller functional profile, which takes room temperature value from the network and implements a PID control algorithm to drive a damper actuator to regulate room temperature. LonMark International forms task groups of interested members to design, approve, and publish functional profiles in numerous functional areas, such as HVAC, security, lighting, and semiconductor manufacturing systems.

LonMark International (LMI)
An independent organization of companies, organizations and individuals committed to the development, manufacture, and use of interoperable LonWorks products and networks.

LonMark Functional Block (formerly: LonMark Object)
A collection of network variables, configuration properties, and associated behavior defined as part of the LonMark interoperability program. LonMark objects define standard formats and semantics for how information is exchanged between devices on a network.

LonTalk® Firmware
A program implementation of the LonTalk protocol residing in ROM in the processor chip of every LonWorks device. A portion of non-volatile RAM in the device is reserved for modifiable configuration parameters to make tradeoffs in performance, security, and reliability for a particular application.

LonTalk Protocol
The protocol used on LonWorks networks to standardize communication. It defines a standard way for devices to exchange information.

LonTalk Router (see LonWorks Router)

LonWorks Component Architecture (LCA)
Echelon Corporation's LNS application programming interface for Microsoft Windows hosts. LCA is an architecture for implementing LonWorks network tools using multiple cooperating software components. LCA provides an open standard with a standard network tool kernel, so that tools may be constructed from software components from multiple vendors. LCA defines a standard Windows OLE service interface for invoking network services and a standard application interface for invoking LCA software components.

LonWorks Control Device
A LonWorks device that senses and/or controls the variables in the system being controlled. It can have any combination of embedded sensors and actuators, or input-output interfaces to external legacy sensors and actuators. The application program in the device can both send and receive values over the network and perform data processing (e.g. linearization, scaling) of the sensed variables and control logic such as PID loop control, data logging, and scheduling.

LonWorks Control Network
Network of intelligent devices (such as sensors, actuators, and controllers) that communicate with each other using a common protocol over one or more communications channels.

LonWorks Device
A device that communicates on a LonWorks network. A LonWorks device may be an application device or a router. LonWorks devices are commonly called devices or nodes in LonWorks documentation. Each LonWorks device includes local processing and input/output (I/O) hardware to process input data from sensors, execute a control task, and control actuators. Each device also includes the capability to communicate with other devices using the LonTalk protocol in firmware. The LonTalk protocol is a complete 7-layer communications protocol that ensures that devices can interoperate using an efficient and reliable communications standard.

Each LonWorks device contains an application program and hardware similar to the following:

- A Neuron Chip or other processor that can run the protocol.
- A transceiver.
- Application electronics to connect the Neuron Chip to I/O devices such as sensors, actuators, displays, and keypads.
- An optional host processor. If a host processor is used, the application executes on the host processor and the Neuron Chip or other processor is used as a network interface.

A LonWorks Router is a special type of LonWorks device.

LonWorks Network Services (LNS®) Architecture
The foundation for interoperable LonWorks installation, maintenance, monitoring, and control tools. Using the services provided by the LNS architecture, tools from multiple vendors can work together to install, maintain, monitor, and control LonWorks networks.

LonWorks Node (see LonWorks Device)

LonWorks Router
An active LonWorks device, which physically connects two LonWorks channels. Each router side can receive a packet, make a decision as to whether the packet needs to be transmitted, and transmit the packet on the other side's channel, if required. The router necessarily injects some delay in the packet transmission.

A router can be configured to be one of the following:

- Repeater: all packets are forwarded.
- Permanent Repeater: all packets are forwarded. Subnets can span permanent repeaters.
- Bridge: all packets in a given domain are forwarded.
- Permanent Bridge all packets in a given domain are forwarded. Subnets can span permanent bridges.
- Learning Router: packets are routed only for a given domain. The router starts as a bridge and reduces forwarding as it learns the topology. Learning routers are vulnerable to failures if configured devices are incorrectly moved within the topology.
- Configured Router: packets are routed only for a given domain. Configured routers forward packets based on configured tables. This is the most reliable and efficient form of router.

Each router side can be addressed by its Neuron ID or by a subnet/node address. The side of the router, which can communicate with the network manager, is referred to as the near side, and the other side as the far side.

LonWorks Technology
LonWorks technology consists of the tools, modules, and ICs required to build intelligent device and to install them in control networks. Each LonWorks device includes local processing and input/output (I/O) hardware to process input data from sensors, execute a control task, and control actuators. Each device also includes the capability to communicate with other devices using the LonTalk protocol in firmware.

Media (see Physical Medium)

Media-Independent
A LonWorks network uses a networking protocol, LonTalk®, to communicate among devices. Therefore, it doesn't care whether the media is twisted pair, power line, fiber, or any other type of physical connection among the network's nodes. In fact, the media can be mixed and matched depending on what saves money or the physical constraints of the installation, therefore the network is said to be media independent.

Message Code
A 1-byte field in a LonTalk message that identifies the type of message. The following table lists the message types supported by the LonTalk protocol and the message codes used for each type, ex. Application Message, Foreign Frame Message, etc.

Message Service Types
The three main types of messaging services are Acknowledged, Unacknowledged, or Repeated messaging. The browser can also be set to use Priority Messaging and, if so, the priority slot that will be used. When using acknowledged or repeated messaging, the user can specify the number of times the browser sends a message. Unacknowledged messaging sends a message one time. The default message service type depends on the device manufacturer, but it is typically acknowledged.

Message Tag
Logical input and output ports that nodes use to send and receive explicit messages. A node always contains a msg_in tag and may contain declared message tags as well. Declared message tags are bi-directional (the node can both send and receive messages with them). The msg_in message tag can only be used to receive messages.

Generally nodes use network variables to communicate with one another since they are interoperable and produce more efficient code.

Message Tag Index

A number used to identify a message tag. Message tag indices are assigned by the Neuron C compiler in the order in which the variables are declared. The first message tag in a program is index 0, the second is index 1, and so on.

MMI
An acronym for Man-Machine Interface, which has been, replaced with the acronym HMI or Human-Machine Interface.

Monitor Sets
Groups of monitor points that the network management tool uses to monitor and control network variables.

Network Address
A device's logical (domain/subnet/node) address. This address is assigned at installation time by the NSS.

Network Configuration
The process of converting a network design into the data elements required by the LonTalk protocol in each node of the network. This includes:

- Assigning domain ID and logical addresses to all devices and groups of devices.
- Binding network variables to create logical connections between devices.
- Configuring the various LonTalk protocol parameters in each node for the desired features and performance, including channel bit rate, acknowledgement, authentication, priority service, etc.

Network Configuration Tool
Software applications which are used to facilitate the network configuration process. Functional network design is as simple as dragging the devices' application functional blocks onto the drawing and connecting inputs and outputs to determine which functional blocks use what network variables.

Network Database
A database used by a network installation tool to allocate and track network resources. The installation tool uses the network database to ensure that resources are allocated correctly and efficiently and so that damaged devices can always be replaced. The network database can also be used by user interface applications to ensure that the user interface names match the installed devices.

Network Driver
Software that runs on the host that manages the interface with the NSI or network interface. This driver isolates the host application from the hardware and software implementation of the interface.

Network Image
A device's network address and connection information. For Neuron® Chip hosted devices, the node's network image is stored in EEPROM on the Neuron Chip. For host applications, all of the network image except the network variable configuration table is stored in EEPROM on the Neuron Chip.

Network Interface Device
Referred to as the Network Adapter and formerly known as a Network Services Interface [NSI]. It is a LonWorks device that has a physical interface to an external host computer such as a PC or a hand-held maintenance tool. The device application program provides communication protocols and an API (application programming interface) to allow host-based applications such as network tools to access the LonWorks network. For optimum performance when attached to LonWorks networks, use an LNS® Fast Network Interface (also known as a VNI).

Network Management
The management of functions, services, events, and properties in an integrated LonWorks network.

Network Management Tool
A software application which is used to facilitate one or more network management tasks, such as network design, configuration, installation, documentation, maintenance, modification, monitoring, or supervisory control.

Network Merge
Initially a network can be installed as a number of independent sub-networks, each with an independent LNS Server, and later the sub-networks can be merged into one network.

When multiple systems are being installed, the installers for each system want to install and debug their installation without interference from other systems. When each system has been correctly installed, they can be merged into a single network.

Network Recovery
The recovery process uses the Database Recovery Wizard to create a network design and the LNS database from an existing physical network. This feature is useful to create a network design if a backup copy does not exist or is unusable. The wizard retrieves as much information as possible from the network devices, given a network domain ID and, optionally, a database description file, and builds a complete LNS database. Even though it can rebuild the entire network database, database recovery is not a replacement for backing up a network database.

Network Services Interface (NSI)
The component in Echelon Corporation's LNS architecture that provides the physical connection to the LonWorks network, manages transactions with the NSS, and provides transparent remote access to the NSS.

Network Services Server (NSS)
The component in Echelon Corporation's LNS architecture that processes network services, maintains the network database, and enables and coordinates multiple points of access. Note that the NSS-10 module combines elements of both the NSI and NSS, but does not support multiple points of access.

Network Variable
Applications may contain network variables and configuration properties. These are defined in the device template. A network variable is any data item (temperature, a switch value, or an actuator position setting) that a particular device application program expects to get from other devices on the network (an input network variable) or expects to make available to other devices on the network (an output network variable). Network variables allow a device to send and receive data over the network to and from other devices.

Every network variable represents a path through which data may flow into or out of a device via the network. All network variables are defined as either input or output; this determines whether the network variable handles data going into or out of the device.

Every network variable and configuration property has a type, which determines data interpretation, i.e. the content and structure of the data. LonMark International defines Standard Configuration Property Types, or SCPTs (pronounce skip'-its), and Standard Network Variable Types, or SNVTs (pronounced sni'-vits), which contain many common data types. For example, SNVT_temp_f is a network variable type for network variables containing temperature as a floating-point number, and SCPT_location is a configuration property type for configuration properties containing the device location as a text string. See the LonMark Resource Files for descriptions of all SNVTs and SCPTs. Applications may use non-standard types called user network variable types (UNVTs) and user configuration property types (UCPTs). These types are defined in user resource files. Additional resource files may be provided by device manufacturers. See Echelon Corporation's LNS Resource File Catalog Utility User's Guide and the LonMark Resource File Developer's Guide in the LNS Utilities and LonMark Reference help file for additional information on using these files.

High-level objects that application devices use to communicate with one another. The types, functions, and number of network variables in each node are determined by the application code within the device. Network variables make it easy to develop networked control applications by eliminating all of the low-level and tedious work of building and sending downlink messages, and receiving and responding to uplink messages.

Network Variable Index
A number used to identify a network variable. Network variables indices are assigned by the Neuron C compiler in the order in which the variables are declared. The first network variable declared is index 0, the second index 1 and so on. The NSS uses the network variable index to refer to the network variables on a node. Neuron Chip hosted devices can declare a maximum of 62 network variables (indices 0 to 61). The NSS for Windows supports host-based devices with up to 4096 network variables (indices 0 to 4095). The NSS-10 module supports host-based nodes with up to 255 network variables (indices 0 to 254). In an array of network variables, each element has a separate index.

Network Variable Selector
A 14 bit number used to identify connected network variables. Network variable selectors are assigned by the NSS during binding.

Network Variable Types
A network variable's type defines its structure and contents. A network variable type can be either a SNVT or a user-defined type.

Neuron Chip

A family of VLSI components that implements the LonTalk protocol. The Neuron Chip is manufactured by multiple vendors and can manage I/O devices and execute user-written application code, or alternatively it can be used to create a network interface to a host processor.

Neuron Chip Firmware

Firmware required to operate a Neuron Chip and implement the LonTalk protocol. This firmware is contained in the VERxxx subdirectories of the LonWorks IMAGES directory (default C:\LONWORKS\IMAGES) in files with the name S*.NX*. A custom system image including the Neuron Chip firmware with extensions for the LTM-10 hardware is included in the LTMSYS.NX file in the VER122 subdirectory of the IMAGES directory. This file does not include the MIP image included in the LTM-10 module.

Neuron ID

The Neuron ID is a hardware address that will change if the hardware changes. Each LonWorks device has a unique 48 bit Neuron ID that was burnt into the Neuron chip when it was manufactured, resulting in approximately 300 trillion different combinations. The Neuron ID is broadcast through the network when a device is pinned so that a logical address (Subnet/Node ID) can be assigned. The network management tool must have the Neuron ID to commission a device. The Neuron ID does not contain information about the address of a device. A device's Neuron ID and Subnet/Node ID can be found in the Identifiers tab of the device's properties. See also Subnet/Node ID.

Node

Another name for a network device / LonWorks device. This term is used in the LonWorks Network Services Architecture.

Node Address

A unique 15-bit logical identifier for each node in a domain. The node address consists of two parts: a 7-bit subnet address and an 8-bit Node ID. The Node ID is unique within the subnet.

Node ID

The third part of the LonTalk addressing hierarchy of domain/subnet/node. At installation time, each device is assigned a unique node ID within its subnet by the NSS.

Node Object

A LonMark functional block that monitors the status of all LonMark functional blocks in a device and makes the status information available for monitoring by network-management tools. It is responsible for disabling, overriding and testing the device.

NSI (see Network Services Interface)

NSS (see Network Services Server)

NV (see Network Variable)

Offline

This causes the application to stop running. In this state, the device will still respond to Online, Wink, and Test commands. An Offline device will still receive network variable updates, but the application will not process these values, and the device will NOT send network variable updates. If the device is reset while Offline, it will go Online after the reset.

Object

The items managed by the NSS. The NSS treats the network as a collection of objects. Objects include nodes, programs, connections, network variables, message tags, and the system.

OffNet

When a LNS® network design is unattached to a physical network or attached and the management mode is set to OffNet, changes to application and network configuration properties are saved in the database and propagated across the network. After being OnNet, network variables, such as temperature or enable/disable, can be changed on the network while working in OffNet mode (after devices have been commissioned and functional blocks added, connected, and configured).

Online

The device's state when its application is executing. A device must be Online to be Enabled or Disabled.

OnNet
When a LNS network design is attached to a network and the management mode is set to OnNet, any changes made are propagated across the network immediately (after devices have been commissioned or functional blocks added, connected, or configured).

When working OnNet, the LNS Server interacts with the physical network and changes devices as the devices are changed in the network management tool. When working OffNet, the LNS Server can browse and test the devices on the physical network, but it will not make changes to the configuration of any devices.

Output Network Variable
A network variable that provides information from a device to other devices on the network.

OSI
OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementors so that their products will consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, most products involved in telecommunication make an attempt to describe themselves in relation to this model. It is also valuable as a single reference view of communication that furnishes everyone a common ground for education and discussion.

Developed by representatives of major computer and telecommunication companies beginning in 1983, OSI was originally intended to be a detailed specification of interfaces. Instead, the committee decided to establish a common reference model for which others could develop detailed interfaces that in turn could become standards. OSI was officially adopted as an international standard by the International Organization of Standards (ISO). Currently, it is Recommendation X.200 of the ITU-TS.

The main idea in OSI is that the process of communication between two end points in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. Each communicating user or program is at a computer equipped with these seven layers of function. So, in a given message between users, there will be a flow of data through each layer at one end down through the layers in that computer and, at the other end, when the message arrives, another flow of data up through the layers in the receiving computer ultimately to the end user or program. The actual programming and hardware that furnishes these seven layers of function is usually a combination of the computer operating system, applications (such as the Web browser), TCP/IP or alternative transport and network protocols, and the software and hardware that enable a signal to be put on one of the lines attached to the computer.

OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host computer. Messages intended for this computer pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host. The seven layers are:

Layer 7: The Application layer...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is not the application itself, although some applications may perform application layer functions.)

Layer 6: The Presentation layer...This is a layer, usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a popup window with the newly arrived text). Sometimes called the syntax layer.

Layer 5: The Session layer...This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with session and connection coordination.

Layer 4: The Transport layer...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures complete data transfer.

Layer 3: The Network layer...This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does routing and forwarding.

Layer 2: The Data-link layer...This layer provides synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.

Layer 1: The Physical layer...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier.

Override Off (FB)
Takes the functional block out of override. The functional block will now function normally. This command requires LonMark-compatible support for this operation in the device's application. The device containing the functional block must be Online to use this command.

Override On (FB)
Puts the FB in override mode. Now the output network variables output their configured override value, even if part of the network is not yet operating and there is no input to the functional block. The override values are set using the functional block's configuration properties. This command requires LonMark -compatible support for this operation in the device's application. See the documentation for the functional block being managed for more information on how to use override. The device containing the functional block must be Online to use this command.

Passive Configuration Tool
Can be any hardware or software that allows for configuring of a single device or set of devices without modifying the network. Plug-Ins used for network-management tools and platforms are the most common PCT.

PCMCIA
A slot format for PC Cards of Type I, Type II, and Type III usually found only on notebook PCs. "PCMCIA" is a term that is no longer used. There are Type II PC Card interfaces available for connecting a PC to a LonWorks network.

PDF (Portable Document Format)
A file format used by LonMark to represent easily download-able paper-like digital versions of our profiles, types, and marketing information. It was created by Adobe Systems Incorporated and later, version 1.4 PDF/Archive (PDF/A-1), was standardized internationally as ISO 19005-1:2005 and corrected as ISO 19005-1:2005/Cor 1:2007. Other formats include PDF/E (ISO 24517), PDF/X (ISO 15930), and the ISO DIS 32000 for PDF version 1.7. LonMark may use any of these PDF formats in documentation made available to members and non-members.

Physical-Layer Repeater
A hardware device that connects two segments of a channel. Unlike a LonWorks router, a physical layer repeater has no intelligence, so it cannot selectively forward packets to increase network capacity, and unlike a router, it forwards damaged packets.

Physical Medium
A communication environment that carries the modulated signals from sources to destinations in a network. LonWorks networks support many media types, including twisted pair, power line, fiber optic cable, radio frequencies, infrared, and coax.

Ping Interval
The ping interval determines how often a device is pinged by the LNS Server to ensure it is still operating and in communication with the network. Set the ping interval based on the expected attachment of the device to the network. If you expect that the device will never move on the network, select Never. Set the interval to 15 minutes for a device you expect will move rarely, to 2 minutes for a device you expect will move fairly often, and to 1 minute for a device you expect will move very often. The default ping interval is Never.

PL-20 Channel
A power line LonWorks channel type defined by ANSI/CEA-709.2 and EN14908-3.

Pre-Engineered Configuration
Configuration information is collected into a database at system design time by the network configuration tool, then is downloaded to the physical nodes later at network installation time.

Poll
An explicit request to a node for the value of one of its network variables.

Polling
A method of monitoring Network Variables. Periodically, as specified by the polling rate, the browser will request and display the value of a network variable. This capability is useful when the network variable value changes rapidly. If the network variable value changes rarely, polling causes unnecessary network traffic. The solution in this situation is to either reduce the polling rate or use bound monitoring. See also Bound Monitoring.

Priority
A mechanism provided by the LonTalk® protocol to allow devices priority access to a channel.

Private Media
A communications media that isn't shared with other parties and uses 1 domain. See also Shared Media.

Program ID
The Program IDs can be displayed as either ASCII text, hex digits, or, if it's a LonMark device, the standard program ID component details. Similar to the Neuron ID, the Program ID is hard coded into a device (in the Neuron Chip's EEPROM) and cannot be changed by the user. Nodes with the same Program ID must have the same external interface.

When a device is defined in the network management tool, the following steps are performed:
1. It looks up the device template name using a Program ID field. If the Program ID field is empty, it uses the TmplName user cell specified in the device shape (see Additional Device User Cells). If a device template with that name is found, it is associated with the device shape.

If the device template specified in the TmplName or ProgramID user cell is not found, the tool looks for the external interface file specified in the XifName user cell. If the external interface file is found, the tool looks up the device template by the program ID specified in the external interface file. If a device template with a matching program ID is found, the user is given the option of changing the name of the device template to the name specified by the device shape, or using the existing device template name. In either case, the existing device template is associated with the device shape.

2. If the device template is not found by the name specified in the TmplName user cell or the program ID specified in the external interface file but the external interface file is found, a new device template is created by importing the external interface file. The new device template is given the name specified in the device shape, and is associated with the device shape.

3. If the device template is not found by the name specified in the TmplName user cell or the program ID specified in the external interface file, and the external interface file is not found, the user is prompted to select an existing device template or to define a new device template by importing an external interface file. The resulting device template is associated with the device shape.

Protocol
A communication scheme defined by (i) services, (ii) data types handled by the services, and (iii) a state transition scheme for each device receiving or providing the protocol services. See also Communication Protocol.

Protocol Analyzer
A tool that can read every packet on a LonWorks channel. A protocol analyzer is different from a device containing the complete LonTalk protocol stack in that it can receive every packet on the network, not just packets that are addressed to it.

Protocol analyzers allow users to observe, analyze, and diagnose the behavior of installed LonWorks networks.

RAM
Random access memory, which has read-write capabilities. This type of memory is volatile implying it loses contents on power loss.

Remote Client
The name of the PC running the network management tool when it is not the LNS® Server.

Remote Full Client
A network management tool communicating with LNS server through a LonWorks or LonWorks IP channel. When OnNet, a remote full client can monitor and control the network without routing the requests through the LNS Server. It can also make configuration changes. There are a number of ways a remote full client can be connected to the LNS Server; the simplest is by connecting directly to a LonWorks network using an LNS network interface.

Remote Lightweight Client
A network management tool communicating with LNS server through an IP channel. The combination is called an LNS/IP channel. When OnNet, a remote lightweight client can monitor and control the network. It canNOT make configuration changes.

Remote Operation
Describes using the network management tool on a PC other than the LNS Server.

Reset
Resets the device by sending the Reset network management command to the device to stop execution, terminate all incoming and outgoing messages, set all temporary settings to their initial values, and start the application again using the original conditions. If the device was offline, it will be put online.

ROM
Read-only (non-volatile) memory. This type of memory maintains its contents after a power loss.

Router
Multiple channels can be connected using routers. Routers are used to manage network message traffic, extend the physical size of a channel (both length and number of devices attached), and to connect channels that use different media (transceiver types) together. Unlike other devices, routers are always attached to two channels. LonWorks /IP routers may be used to connect LonWorks networks to the Internet or IP networks. See also LonWorks Router.

SCPT (see Standard Configuration Property Type)

Segment (see Channel Segment)

Selector
It associates a network variable in an FB with the connection, thus determining which network variable is the intended recipient of the update. The following restrictions apply to selector values:
1. Network variables within a selection can only share one selector.
2. Multiple input network variables on a node cannot share a selector.
3. Each network variable can only have one selector.
4. An alias, required to resolve selector conflicts, uses a selector.

Self-Documentation
A mechanism that a device can use to provide descriptive information. Self-documentation can be provided for the device's program and network variables. A provider of user-defined services and events may also support self-documentation for itself, its services, events, objects, and properties. When possible, the NSS makes self-documentation information available to the host application through properties.

Self-Identification
A mechanism that a device can use to document the types of network variables it contains (identified by SNVT ID). When available, the NSS and the network management tool automatically import this information to determine a device's external interface.

Sensor
Any component that is used to determine the condition or value of a physical system variable, or to accept commands from a human operator.

Shared Media
A shared media system is one where multiple, independently managed networks share one (or more) physical channels. For example, in a building, if multiple systems use the power wiring of the building as a channel, they are shared media systems. Systems using shared media must follow pre-established rules to ensure that they don't interfere with one another during system configuration or repair. See also Private Media.

SLTA (Serial LonTalk Adapter)
A serial NSI interface with built-in twisted pair transceiver that connects to any host with an EIA-232 (formerly RS232) port. It can also connect to the host remotely using a Hayes compatible modem. For example, Echelon Corporation's "SLTA-10" is for use with a remote application or for portable hosts that do not contain a Type II PC slot or USB slot.

A network interface that provides an EIA-232 (formally RS-232) interface to connect a host processor to a LonWorks network.

SNVT (see Standard Network Variable Type)

SNVT ID
A code used to identify the type of SNVT used by a network variable. A value of 0 indicates that the variable is not a SNVT. Also sometimes called a SNVT index.

SOAP (once known as: Simple Object Access Protocol)

SOAP is a simple XML based protocol to let applications exchange information over HTTP. It provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

Source Address
The logical node address of the transmitting node, contained in every packet transmitted over a LonWorks network.

Standard Configuration Property Type (SCPT)
A method of storing application configuration data using files that are read and written using the LonWorks File Transfer Protocol (LW-FTP); via direct memory-read/write of the file data (DM-R/W); or through config network variables using typical network-management tools. SCPTs are defined by LonMark International to ensure that LonMark devices can be configured without a proprietary configuration tool. There are hundreds of them defined by LonMark.

Standard Network Variable Type (SNVT)
SNVTs facilitate interoperability by providing a well-defined interface for communication between devices made by different manufacturers. See the Echelon or LonMark website for a current list and documentation.

Stencils
Stencils are used to simplify finding and reusing shapes. Stencils contain shapes that may be reused in the drawings. The shapes contained on a stencil are called master shapes.

Subnet
A logical collection containing up to 127 devices within a domain. Up to 255 subnets can be defined within a single domain. All devices in a subnet must be on the same segment. Subnets cannot cross non-permanent type routers.

Subnet/Node ID
The logical address assigned to a device after it has been commissioned. In a network with one subnet, the LNS Network Interface is assigned a Subnet/Node ID of 1/127, whereas all other devices Subnet/Node IDs begin with an address of 1/1 and increase sequentially to 1/2, 1/3, etc. the network management tool automatically assigns these values.

The subnet portion of the ID is used to route packets. Packets will only be exposed to other channels (subnets) as required, i.e. the source channel, the destination channel and all channels between the source and destination. The node ID portion of the address is used to identify a device on a subnet.

The address is hardware independent. When a device is replaced, the new device will use the same Subnet/Node ID. A device's Subnet/Node ID and Neuron ID can be found in the Identifiers tab of the device's properties. See also Neuron ID.

Subsystems
Subsystems contain devices, routers, and functional blocks. Subsystems allow the placement of devices, routers, and functional blocks onto separate folders for organizational purposes.

Subsystems may also be placed in other subsystems, allowing the creation of a subsystem hierarchy for large networks. For example, a network may consist of HVAC, lighting, security, and operator subsystems. These may be further divided into subsystems for each floor, and each floor divided into subsystems for each room.

Supernode
A subsystem with its own network variable interface. Supernodes may be used to organize groups of devices into logical units and to hide complex subsystem details, exposing only the desired network variables. This reduces errors and decreases the time required for network engineering and commissioning. A supernode's network variable interface may contain any network variable on any device found within the supernode or its nested subsystems.

Target
The destination of a connection, specified by node handle and network variable or message tag index. Each connection is defined in terms of a hub and a set of targets that connect to the hub. For network variable connections, the hub must be either the only input or the only output in the connection. For example, if the hub is an output network variable, all the targets in the connection must be input network variables.

Terminator
A device comprised of a capacitor and a resistive element providing electrical termination for signals on a given channel type. Almost all networks require a specific type of terminator depending on the channel type, ex. twisted pair, and the network topology, i.e. free or bus.

Thin Client
A low-cost, centrally managed computer devoid of CD-ROM players, diskette drives, and expansion slots. The term derives from the fact that small computers in networks tend to be clients, not servers. Since the idea is to limit the capabilities of these computers to only essential applications, they tend to be purchased and remain "thin" in terms of the client applications they include.

The term "thin client" seems to be used as a synonym for both the NetPC and the network computer (NC), which are somewhat different concepts. The Net PC is based on Intel microprocessors and Windows software (Intel was a leader in defining the Net PC specification). The network computer (NC) is a concept backed by Oracle and Sun Microsystems that may or may not use Intel microprocessors and uses a Java-based operating system. The increased numbers of thin clients in today's workplace and educational facilities reflects a corporate and institutional need for low-cost computers dedicated to Internet use.

TP/FT-10
The free topology twisted pair LonWorks channel type, 78 kbps bit rate.

TP/XF-1250
A bus twisted pair LonWorks channel type, 1250 kbps bit rate.

TP/XF-78
A bus twisted pair LonWorks channel type, 78 kbps bit rate.

Transaction
A mechanism to group a series of service invocations into a single operation. Transactions are used to make sure that either the entire series of service invocations take effect, or that none of them do. An LNS® host application can explicitly manage transactions or it can let the NSS implicitly start and commit transactions as needed.

Transceiver
The device that physically connects a Neuron® Chip to its channel. The transceiver implements layer 1 of the LonTalk® protocol. There are Network Interface Transceivers (TP-XF1250, TP-FT10), Channel Transceivers (TP-XF1250, TP-FT10) and Device Transceivers (TP-XF1250, TP-FT10, TP-XF78).

Transceiver ID
A transceiver ID is a number between 0 and 31 that represents a different type of transceiver. Transceiver IDs are reported by routers and NSIs as a function of the type of transceiver attached. Note that LonWorks routers do not support transceiver ID. For example the STDXCVR.TXT file in C:\LonWorks\types shows TP/XF-1250 with nXcvrId = 3 and TP/XF-78 with nXcvrId = 1 and the XcvrCount = 24. Also, a transceiver ID of 30 is reserved to indicate a custom transceiver. Transceiver IDs are distinct from transceiver types.

Transceiver Type
A transceiver type is a number that refers to an entry in the standard transceiver type file (C:\LonWorks\types\STDXCVR.TYP). An entry in this file may reference a transceiver ID. However, some entries may have no transceiver ID.

Typeless Network Variable
A network variable for which there is no SNVT type or length information available. Typeless network variables can be bound to any other network variable type; it is the responsibility of the installation tool's application program to prevent nonsensical connections from being formed that contain typeless network variables.

UCPT (see User Configuration Property Type)

Unconfigured Device
A device state where the device has an application image, but no network image. The device must be configured before it can operate on the network.

Unconfiguring devices
Performed by pressing and holding the service pin down for 15-20 seconds until the power led flashes briefly (p 5-22). This is also known as decommissioning a device and can be done using the device’s Manage shortcut menu.

Uplink
Data transfer from the network and the NSI toward the host.

Upstream Device
The device sending a network variable update.

User Configuration Property Type (UCPT)
A non-standard data structure used for configuration of the application program in a LonMark device. UCPTs can be used only when there is no appropriate Standard Configuration Property Type (SCPT) defined. LonMark certified devices have UCPTs documented in resource files according to a standard format, in order to allow the devices to be configured without the need for proprietary configuration tools.

User Profiles
1) Can only be enabled by the Administrator of the network. The Administrator will set the following values for each user: UI setting, Access control (subsystems), Privileges (object in a subsystem) and Actions (ex. Read, Modify, Commission, etc.). This allows different users to log on to a network and only make changes according to their user name. All passwords are case sensitive and cannot be retrieved if they have been forgotten.

2) Manufacturer-defined functional profiles created to ease installation and configuration by the end user: User Functional Profile Templates (UFPTs).

VNI
For optimum performance when attached to LonWorks networks, use an LNS Fast Network Interface (also known as a VNI).

Wink
Causes the device to generate an application dependant audible or visible response such as flashing the power LED. This command will ONLY have an effect if the device supports the Wink function. This can be useful for identification and testing purposes.

XIF (see Device Interface File)

XML (Extensible Markup Language)
A flexible way to create standard information formats and share both the format and the data on the World Wide Web.