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.

Ad Hoc Configuration Scenario

In this scenario, the network integration tool is used to design and install the network on-site. 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.

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 created when a connection is made.

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 the alias network variable has a unique selector, it allows for complex connections to be created.

ANSI/EIA/CEA–709.1 (EN14908–1) protocol

Also known as the LonTalk protocol.

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 the network integration tool. The application determines how a device functions.

A Glossary of LonWorks Terms

Application Configuration

A process by which the application program in a device 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.

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.

Binding

Binding is a process that takes place during network design and installation to define connections between LonWorks devices. 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”.

Bit Rate

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

Bus Topology

Devices in a bus topology network are connected in a daisy-chain arrangement. The bus allows for easier troubleshooting and simplifies network schematics for expansion and faultfinding. Bus topology networks are terminated at each end. See also free topology.

Channel / 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 topology.

Client

A task requesting service from a server. See 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.

See Remote Client.

Commissioning a Device

The process of using a network management 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 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 LonWorks protocol is defined by the ANSI/EIA 709-1 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 similar logical inputs.

Connections are considered to intersect when they share one or more network variables.

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 LonWorks Device.

Device Interface

See External Interface File.

Device Name

The name given to a device when adding it to the network. It is recommended that a descriptive name is supplied for each LonWorks device.

Device State

A device may be in either the offline or online state.

The device state is displayed in the Network Integrator subsystem view and device properties Status tab.

Device Template

A template used by the network management tool that contains all the attributes of a given device type (functional blocks, network variables, configuration properties, and so on).

Disabling a Device

Disabling a device disables all LonMark objects on the device. A device must be Online to be enabled or disabled.

Disabling Functional Blocks

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.

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 top 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 – are 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.

EEPROM

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

Enabling a device

Enabling a device enables all LonMark objects on the device. A device must be Online to be enabled or disabled.

Enabling Functional Blocks

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. After the network is attached, the network design is associated with the physical devices through a simple commissioning step and the configuration information, defined during network design, is loaded into the devices and routers.

Event

The mechanism that the NSS uses to inform a LNS host application of network happenings, such as the arrival of a service pin message, or a change of network addresses. The LNS host application subscribes to and stops 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.

External Interface File

The interface defined by the network inputs and outputs to the functional blocks on a device is called the *device interface* (it is also called the *application-layer interface* or the *external interface*). A network tool may upload the device interface definition from the device, or it may read the device interface definition from a standalone file called the *device interface (XIF) file*. In open multi-vendor networks, the design of the device interface is vital to providing interoperability and easy integration. Standardization of the device interfaces is an important element of designing for interoperability.

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.

Free Topology

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 topology allows wire to be strung from any point to any other, in bus, daisy chained, star, ring, or loop topologies, 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. Circon recommends always using bus topology.

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.

A functional block performs a task by receiving configuration and operational data inputs, processing the data, and sending operational data outputs. A functional block may receive inputs from the network, from hardware attached to the device, or from other functional blocks on a device. A functional block may send outputs to the network, to hardware attached to the device, or to other functional blocks on the device.

To define multiple functional blocks within a device, the device must declare LonMark objects as defined by the LonMark Application Layer Interoperability Guidelines. A LonMark object is a functional block that is documented in accordance with the LonMark guidelines. The device does not have to be LonMark -certified, but it must conform to the LonMark application layer guidelines and it must have a program ID that identifies the application as a LonMark -certified (type 8) or LonMark -compliant (type 9) application.

Devices that do not comply with the LonMark guidelines are limited to a single functional block that contains all the input and output network variables for the device.

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. Groups are identified by a one byte group ID.

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.

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 device 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 network variables allow the computer to collect network data with greater efficiency than polling.

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.

i.LON Internet Server

A family of products from Echelon, designed to provide an IP and Web server interface to LonWorks networks.

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 such as a PCLTA-21, SLTA-10 or i.LON.

Infrastructure Subsystem

The top-level subsystem in an LNS network database that is the central place for managing routers, other subsystems, channels and the LNS Network Interface.

Input Network Variable

See network variable.

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.

Data Server API

A standard component that provides high performance monitoring and control. Using the data server, client applications, such as Network Integrator 3, 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.

Legacy I/O Device

A sensor or actuator that cannot attach directly 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

See LonWorks Network Services.

LNS Client

Any application that uses the services of the LNS Server. Network Integrator and LNS plug-ins are LNS clients.

LNS device credit

Each time Network Integrator is used to commission a device when it is OnNet, one LNS device credit is used. No credits are used when commissioning routers, network service devices, or the LNS Server device. Network Integrator uses no additional credits for replacing devices that have been commissioned by Network Integrator or recommissioning devices that are already in the database even if the application or program IDs change in the devices. No LNS device credits are used when designing a network using the engineered system installation scenario; they are deducted when the physical devices are commissioned. LNS device credits are returned for deleting devices commissioned with Network Integrator. See also LonMaker credits.

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 a device. A device may be in one of the following load states: applicationless, configured, or unconfigured.

Local Client or Local Full Client

Network Integrator running on the same PC as the LNS Server. The simplest Local Client configuration, a local application, is where Network Integrator 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. The i.LON Internet Server, when used as a router, allows a LonWorks/IP channel to be connected to a LonWorks channel. Using the i.LON server, a local application can communicate with a LonWorks network using an IP network card.

LonMaker Integration Tool

An LNS Turbo-based LonWorks network management tool from Echelon Corporation.

LonMaker Credits

Each time the LonMaker tool is used to commission a device when it is OnNet, one LonMaker credit is used. No credits are used when commissioning routers, network service devices, or the LNS Server device. The LonMaker tool uses no additional credits for replacing devices that have been commissioned by the LonMaker tool or recommissioning devices that are already in the database even if the application or program IDs change in the devices. No LonMaker credits are used when designing a network using the engineered system installation scenario; they are deducted when the physical devices are commissioned. LonMaker credits are returned for deleting devices commissioned with the LonMaker tool. See also LNS credits.

A Glossary of LonWorks Terms

LONMARK

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

LONMARK–Certified

A LonMark certified device has passed the LonMark certification tests and can use a program ID type 8.

LONMARK Device

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

LONMARK Functional Profile

A LonMark object 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. Complete documentation on all LonMark objects can be found on the LonMark Association web site.

LONMARK International

An independent organization of LonWorks developers, system integrators, and end-users that define standards to ensure interoperability between LonWorks devices from multiple manufacturers.

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 most LonWorks devices. 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

A device publishes information as instructed by the application that it is running. The applications on different devices are not synchronized, and it is possible that multiple devices may all try to communicate 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. These rules and procedures are defined by the ANSI/EIA/CEA-709.1 (EN14908-1) (LonTalk) protocol. The protocol defines the format of the messages being transmitted between devices and defines the actions expected when one device sends a message to another. The protocol implementation normally takes the form of embedded software or firmware code in each device on the network.

LonTalk Router

See LonWorks Router.

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 the following hardware:

- A Neuron Chip.
- 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 is used as a network interface.

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. Network Integrator runs on the LNS® network operating system.

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. Two development tools are available for LonWorks devices.

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.

Network Address

A node'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

A software application, such as Network Integrator, which is used to facilitate the network configuration process.

Network Database

A network database contains network and device configuration information for a single LonWorks network.

The LNS global database, manage by the LNS server, contains the network databases. There is a network database for each network managed by the LNS server.

A network database is used by a network management tool, such as Network Integrator or Echelon's LonMaker, to allocate and track network resources. The network management 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, such as Visual Integrator or a Web server, to monitor the network.

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, the entire network image except the network variable configuration table is stored in EEPROM on the Neuron Chip.

Network Integrator

An LNS Turbo-based LonWorks network management tool from Circon Systems Corporation.

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 example, the Echelon PCLTA-21 LonTalk Adapter is a network interface device packaged on a standard PC PCI adapter card. It plugs into the PCI bus internal to the PC, providing access to the network for network management tools compatible with LNS such as Network Integrator. 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, such as Network Integrator and Echelon's LonMaker for Windows Integration Tool, 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. Examples of installations that may use this technique include the following:

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.

The database merge process has the following limitations:

- Server support only - The database merge does not support merging of networks from a remote client (full or lightweight).
- Single direction only - There is no provision for backing out a merged database after the process has begun, although it may be restored to a previously backed-up database and drawing.

- Single root subsystem - The database merge utility only supports networks that have a single root subsystem. If the network has more than one root subsystem, the pre-merge utility will fail. All but one of the root subsystems must be removed or relocated before retrying the pre-merge utility.

Network Resynchronization

Resynchronization provides a convenient method for resynchronizing physical devices with the LNS network database — for example, to recommission all devices on the network. Resynchronization always uses the information in the LNS network database to resolve any inconsistencies.

Network Services Interface (NSI)

The component in the LNS architecture that provides the physical connection to the LonWorks network, manages transactions with the NSS, and provides transparent remote access to the NSS.

Local NSI connections use PCC-10, PCLTA-10 or PCLTA-20 network interfaces.

Remote NSI network connections would use either an SLTA-10 or i.LON 1000 interface.

Network Services Server (NSS)

The component in the 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 “skipits), and Standard Network Variable Types, or SNVTs (pronounced “snivits), 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 LNS Utilities and LonMark Reference help file for a list and description 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.

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. See also Selector.

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 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.

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. Network Integrator 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 object that monitors the status of all LonMark objects in a node and makes the status information available for monitoring by network management tools. When more than one object is defined on a node, it is the FB that manages the other FBs on the node. It is responsible for disabling, overriding and testing devices.

NSI

See Network Services Interface.

NSS

See Network Services Server.

NV

See Network Variable.

Offline

The device's state when the application is not 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 when it is 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 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 when Network Integrator is set to OnNet. After being OnNet, network variables, such as temperature or enable/disable, can be changed on the network when the network management tool is 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 in the online state to be enabled or disabled.

OnNet

When a 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 design. 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

See Network Variable.

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 implementers 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 them 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.

PCC–10

A type II PC (formerly PCMCIA) card NSI that includes an integral FTT-10 transceiver. Other transceiver types can be connected to the PCC-10 via external transceiver "pods". The PCC-10 is the best NSI to use with laptop, notebook, or embedded PCs.

PCLTA–10

A 1/2 size ISA card NSI. Unlike the PCNSI, it includes a twisted pair transceiver onboard, eliminating the need to attach a separate SMX transceiver assembly. The PCLTA-10 also supports the Windows plug-and-play standard. The PCLTA-10 is obsolete. See PCLTA-20/21.

PCLTA–20/21

A PCI card NSI. Unlike the PCNSI, it includes a twisted pair transceiver onboard, eliminating the need to attach a separate SMX transceiver assembly. The PCLTA-20/21 also supports the Windows plug-and-play standard. The PCLTA-20/21 is the best NSI to use on a desktop PC host that attaches to a twisted pair channel.

Peer–To–Peer

A control strategy in which independent intelligent devices share information directly with each other and make their own control decisions without the need or delay of using an intermediate, central, or master controller. Because of the enhanced system reliability introduced by eliminating the master (a single point of failure) and the reduced installation and configuration cost inherent in peer-to-peer designs, LonWorks networks often use a peer-to-peer control strategy.

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 supports 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 or PLT–22

A power line LonWorks channel type.

Plug–Ins

Easy-to-use LNS-compatible graphical software for configuring an application program in a LonWorks device. Compatibility with LonWorks Network Services (LNS) automatically assures compatibility with other LNS-based network tools. For example, the applications in the Circon controllers all have LNS-based plug-ins for configuration. After defining and performing network configuration of one of these devices using the network management tool, the user can right-click on the device icon, select Configure from the shortcut menu and the configuration plug-in is immediately launched from within the network management tool.

Pre–Engineered Configuration

Configuration information is collected into a database at system design time by the network configuration tool, and 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 one domain. See also Shared Media.

Program ID

See standard program ID.

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 has read-write capabilities. This type of memory is volatile implying it loses contents on power loss.

Remote Client

A computer that is running Network Integrator but is not running the LNS Server. Network Integrator gains access to the LNS Server as a remote client.

Remote Full Client

A network integration 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 integration 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 can also make configuration changes.

Remote Operation

Describes using the network integration tool on a PC other than the PC where the LNS Server resides.

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.

Resource Catalog

The resource catalog is used as file system for the location of the various device resource files contained on a PC. Network management/monitoring tools can use the catalog to easily keep track of which device files are in which directory on the PC.

Resource Files

Resource files are used to publish definitions for both standard and manufacturer defined resources, for a particular type of LonMark-certified device as identified by its SPID. Standard resource files are maintained and provided by LonMark.

Standard resources include *standard functional profiles* (also called *LonMark profiles*), *standard network variable types (SNVTs)*, *standard configuration property types (SCPTs)*, and *standard enumeration types*.

Manufacturer-defined resources include *user functional profiles*, *user network variable types (UNVTs)*, *user configuration property types (UCPTs)*, and *user enumeration types*.

Resource files are grouped into *resource file sets*, where each set defines functional profiles, network variable types, configuration property types, enumeration types, strings, and formats for specified device types. The range of device types that a resource file set applies to is called the *scope* of the resource file set. For example, the scope may specify that the resource file set applies to an individual device type or to all device types.

ROM

Read-Only Memory. A type of memory that 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, such as the Echelon i.LON Internet Server may be used to connect LonWorks networks to the Internet or IP networks.

See LonWorks Router for details about types of routers.

SCPT

See Standard Configuration Property Type.

Segment

See Channel Segment.

Selector

A network variable selector is a 14-bit number used to identify connected network variables (i.e. network variables that are part of the connection). Devices may use different names to refer to a network variable, or network variables may be located at different offsets within each device's memory, resulting in a different network variable index within each device. However, each connection in the system is assigned a selector value, and all network variables in a given connection must use the same selector, unless network variable aliases are used. Intersecting connections must also use the same network variable selector. Connections are considered to intersect when they share one or more network variables.

LNS can also **share** network variable selectors among unicast connections if those connections are between disjoint devices. Network variable selectors that are shared between unrelated connections come from the pool of sharable selectors available on the system.

However, LNS will need to assign an **exclusive** network variable selector to sets of intersecting connections, and to each connection that contains multiple targets. The selectors used to form this type of connection are called exclusive selectors.

This does not mean that all network variable selectors in a network need to be unique. Any network variable selector can be used multiple times in the same network, provided that the devices using the same selector do not share connections or network addresses, so that the selector is unambiguous to all of the devices.

This does not also mean that only a single selector may apply to a given network variable. The LNS Server supports network variable aliases transparently. Network variable aliases allow LNS to map multiple selector values to a single input or output network variable. The number of network variable aliases on each device is defined by the device manufacturer, and cannot be changed by an LNS application, but the LNS Server will take advantage of available aliases, if necessary.

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 integration tool automatically import this information to determine a device's external interface.

Sensor

A component 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.

ShortStack

ShortStack is technology from Echelon used to LonWorks-network-enable any microprocessor-based device. The device's application uses the ShortStack API and the hardware includes the ShortStack Micro Server. Circon's APC-300 uses ShortStack technology.

SLTA–10

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. The SLTA-10 is the best NSI to use for remote application or for portable hosts that do not contain a type II PC 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.

Source Address

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

Standard Configuration Property Type (SCPT)

Configuration properties or CPs provide a standard mechanism used by network tools to download and store configuration data to LONWORKS devices using files that are read and written using the LonTalk File Transfer Protocol.

LonMark International defines a standard set of configuration property types called Standard Configuration Property Types (SCPTs) to ensure that LonMark devices can be configured without a proprietary configuration tool.

SCPTs are defined for a wide range of configuration properties that are used in many kinds of functional profiles: dead-bands and offsets, device default values, minimum and maximum limits, gain settings, and delay times. To enhance interoperability SCPTs are used wherever applicable.

The document *LonMark SNVT and SCPT Master List* provides a complete list of all the standard configuration property types (SCPTs) as defined by the LonMark International. Visit the LonMark Web site to obtain this document.

Standard Network Variable Type (SNVT)

SNVTs facilitate interoperability by providing a well-defined interface for communication between devices made by different manufacturers. See the LonMark website for a current list and documentation. Currently there are approximately 200 different SNVTs.

Standard Program ID (SPID)

The *standard program ID (SPID)* is an 8-byte number within the read-only data structure of a device as defined by the ANSI/EIA/CEA-709.1 (EN14908-1) protocol. It uniquely identifies the device interface for a device. It is used by network tools to associate a device with a device interface definition. This speeds up the commissioning process by allowing a network management tool to obtain the device interface definition without uploading the entire definition from every device.

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 integration 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. Used by the network management tool 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.

Target

The destination of a connection, specified by a 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, 78Kbps bit rate.

TP/XF–1250

A bus twisted pair LonWorks channel type, 1.25Mbps bit rate.

TP/XF–78

A bus twisted pair LonWorks channel type, 78Kbps 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 network management tool to prevent nonsensical connections from being formed that contain typeless network variables.

U10 USB Network Interface

A LonWorks network interface for TP/FT-10 networks, in a USB form factor. The driver is included in Echelon's OpenLDV.

UCPT

See User-Defined 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 a Device

Performed by pressing and holding the service pin down for 15-20 seconds until the power led flashes briefly. This is also known as decommissioning a device.

UNVT

See User-Defined Network Variable Type.

Uplink

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

Upstream Device

The device sending a network variable update.

User–Defined 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. See also Standard Configuration Property Type (SCPT).

User–Defined Network Variable Type (UNVT)

A non-standard network variable provided by a manufacturer with their device application software. See also Standard Network Variable Type (SNVT).

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 dependent 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 External Interface File.