

Automatización de edificios

# PROTOCOLO COMUNICACIONES LONWORKS

# Sistemas de control en edificios

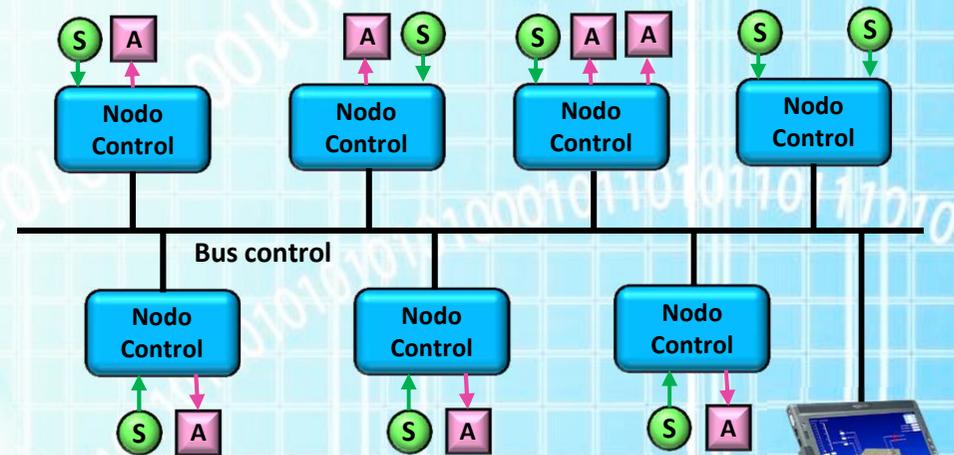
En otros temas, hemos visto que existen dos formas de sistemas de control de edificios según su gestión, el **centralizado** y el **distribuido**.

En el caso del **modelo centralizado** todas las **decisiones se toman en un único sistema**, en el modelo **distribuido**, las decisiones se toman en **multitud de pequeños sistemas (nodos)**, lo que hace necesario que existan **comunicaciones entre nodos**.



El control central de procesos recibe datos, toma todas las decisiones y ordena las acciones correspondientes.

Si el sistema central falla, el sistema deja de funcionar.



Los procesos están distribuidos por todos los nodos.

Cada 'nodo' tiene su propio programa de control.

Si un 'nodo' falla el sistema no cae.

Los nodos se comunican entre si por el 'bus'.

# Sistemas de control en edificios

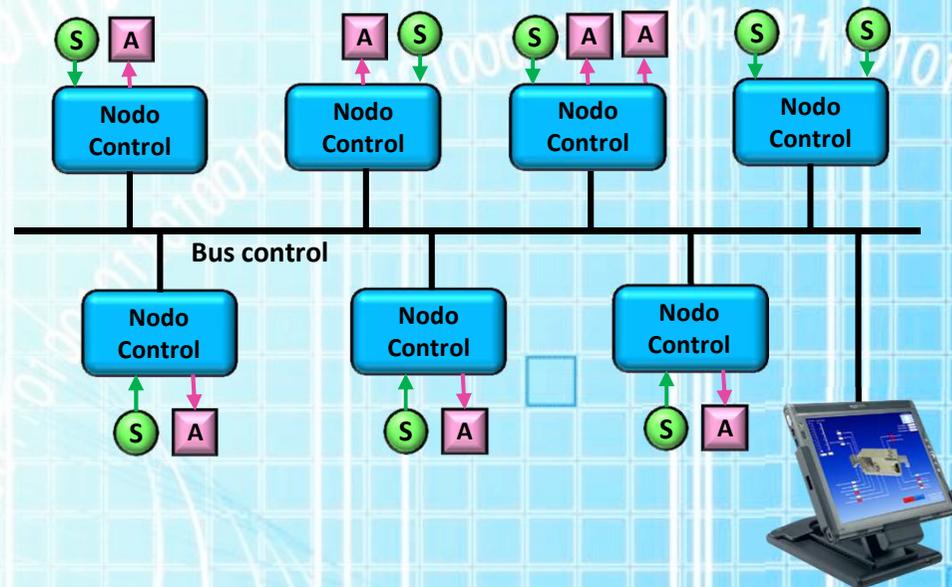
## Modelo distribuido

Este sistema presenta algunos **problemas relacionados con su organización**:

- Existen **multitud de 'procesadores'** que resuelven problemas en su entorno, pero:
  - ✓ Pueden **necesitar información de otros nodos** o procesadores.
  - ✓ Puede que tengan que **enviar información a otros nodos** o procesadores
  - ✓ Las **comunicaciones tienen que estar reguladas**, ¿Quién habla primero?, ¿hay choques?
  - ✓ ¿Cómo se **identifica cada procesador o nodo**?
  - ✓ ¿Cómo se sabe **de donde viene un dato**?, ¿qué **tipo de dato** es?, ¿cómo se **decodifica**?

### Otras cuestiones

- ¿Se puede **añadir un nuevo nodo** de **forma sencilla, sin cambiarlo todo**?
- ¿Se puede **cambiar la configuración del conjunto de forma sencilla**?
- El conjunto, ¿se comporta como **un único sistema** coordinado?
- ¿Se puede **sustituir un nodo** de forma simple?
- ¿Se puede **actualizar el firmware** de uno o varios elementos **de forma sencilla**?
- Si **algún nodo cae**, ¿**el sistema se paraliza** o sigue funcionando?

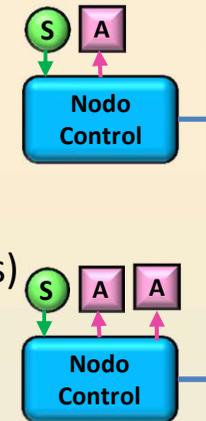


# Sistemas de control en edificios

## Modelo distribuido: Protocolo comunicaciones

Para **resolver las necesidades de un modelo distribuido** es preciso un sistema información y de comunicaciones o **'protocolo'** que resuelva de forma **eficiente** los problemas planteados:

- **Cada nodo tiene información propia (sensores, procesos, etc).**
  - Estos **datos se deberán transmitir al nodo o nodos que lo precisen**
    - ✓ **Variables con información:** hora temperatura, humedad, etc.
    - ✓ Los datos deberán tener un **formato estándar.**
    - ✓ Cada **dato debe identificarse de otros similares** (p.ej. varias temperaturas)
    - ✓ Debe haber **seguridad en la 'entrega'** del dato en destino.
  - Estos **datos pueden solicitarse** desde un **sistema HMI o Scada**
- **Cada nodo puede precisar recibir datos de otros nodos para tomar decisiones**
  - Para operar sobre actuadores pueden ser **necesarios datos externos en otros nodos**
    - ✓ **Variables con información:** hora temperatura, humedad, etc.
    - ✓ Los datos deberán tener un **formato estándar.**
    - ✓ Cada **dato debe identificarse de otros similares** (p.ej. varias temperaturas)
    - ✓ Se debe comunicar al origen el **acuse de recibo en algunos datos.**
  - El sistema **HMI o Scada** puede **modificar datos o desear actuar sobre las salidas.**



**Se hace necesario un bus (enlace físico) y un protocolo de comunicaciones**

# Sistemas de control en edificios

## Modelo distribuido: Protocolo comunicaciones

También se ha mencionado que una de las **ventajas del sistema distribuido** es que es más flexible para su **crecimiento y modificación**:

- **Se desea ampliar la instalación con nodos o sensores nuevos**
  - Deben de poderse **añadir nodos sin alterar la instalación existente**, esto implica:
    - ✓ Necesidad de un sistema lo más simple posible de **gestión de nodos**.
    - ✓ Posibilidad de **modificar y añadir nuevas comunicaciones** de forma simple.
    - ✓ Que **no sea necesario modificar la instalación física** existente ni detenerla.
    - ✓ En caso de **sensores/actuadores nuevos poder incorporarlos al sistema / protocolo**.
  - Incorporar los **nuevos nodos al HMI o Scada**.
- **Es necesario reemplazar nodos averiados o dar de baja alguno de ellos:**
  - Herramientas de **sustitución de un nodo averiado** fáciles de usar:
    - ✓ La **configuración del nodo averiado debe pasar al nodo nuevo**.
    - ✓ El **nodo nuevo debe comunicarse con los nodos que lo hacía el averiado**.
    - ✓ Estas **operaciones deben realizarse sin parar el sistema**.
  - Se podrá **dar de baja un nodo no necesario** sin alterar al resto.

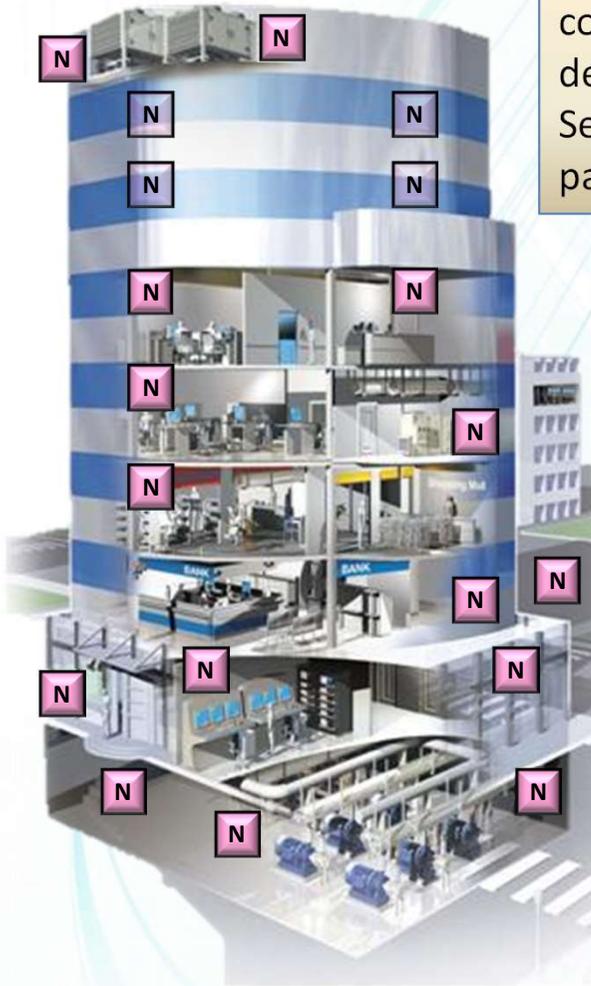
**Es necesario el uso de herramientas / software para estas labores**

# Sistemas de control en edificios

## Modelo distribuido: Protocolo comunicaciones

Desde una **visión global**, todos los nodos existentes deberán comportarse como una **'única entidad'** que podrá ser **monitorizada** desde al menos un equipo Scada o HMI.

Se puede dar el caso de sistemas sencillos que no usen un equipo para monitorización, por ejemplo el control de una piscina.



# Sistemas de control en edificios

## Modelo distribuido: bus y protocolo

En un sistema distribuido, es muy importante los **tipos de bus soportados** para llevar a cabo las **comunicaciones** y la **posibilidad de incorporar pasarelas** para poder mezclarlos, podemos destacar:

- **Par simple trenzado** (bus de dos hilos).
- Par trenzado con alimentación incluida.
- **Power Line** ( usando la red eléctrica ), también conocidos como ‘de corrientes portadoras’.
- **TCP/IP** por redes de datos convencionales (ethernet, fibra óptica, etc.)
- **Inalámbricos**
- Estándares **serie** como RS-485, RS-422 o RS-232 (el más usado es RS-485)
- etc.

Otra característica muy importante es que el sistema sea **‘abierto’ o ‘propietario’**, en los sistemas **‘abiertos’** podemos encontrar **productos de cientos de fabricantes**, que cubran prácticamente todas las necesidades para cualquier circunstancia en domótica / inmótica, además si el sistema es **‘abierto’ podremos mezclar los productos de los fabricantes sin ningún problema.**

Actualmente los sistemas abiertos más importantes que cumplen estas características son:

- ✓ **KNX**
- ✓ **LonWorks**
- ✓ **BacNet (con un gran crecimiento)**

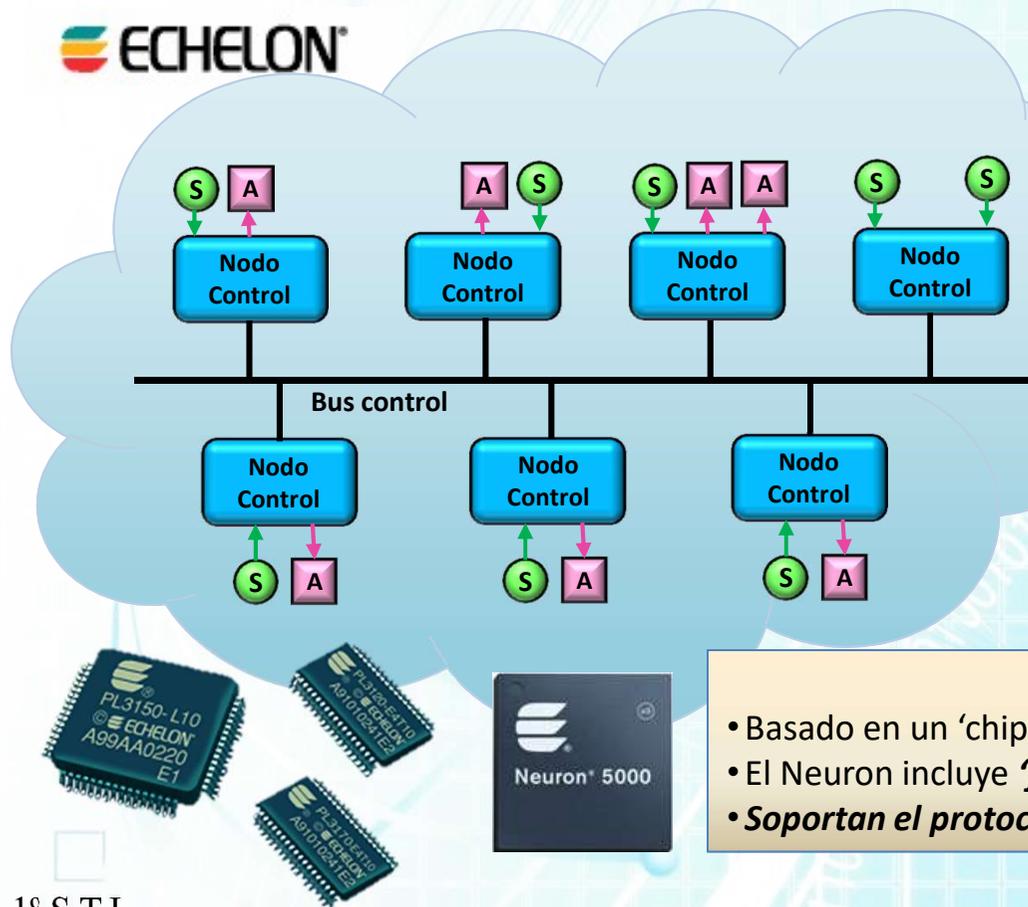


Nosotros nos vamos a centrar en este capítulo en el sistema LonWorks

# Sistemas LonWorks

## Modelo

El sistema **Lonworks** aparece en 1988 creado por la empresa americana **Echelon**, hay más de 4000 fabricantes de dispositivos que usan esta tecnología para construir sistemas distribuidos usados en una gran cantidad de ámbitos: edificios inteligentes, fábricas, viviendas, sistemas de transporte, aviones, trenes, vehículos militares, etc. El protocolo de comunicaciones se denomina **LonTalk**.



- Dispositivos inteligentes comunicados entre sí.
- No hay una categoría, todos son 'iguales'.
- No hay un organizador o supervisor.

### Requisitos del sistema: bus + protocolo

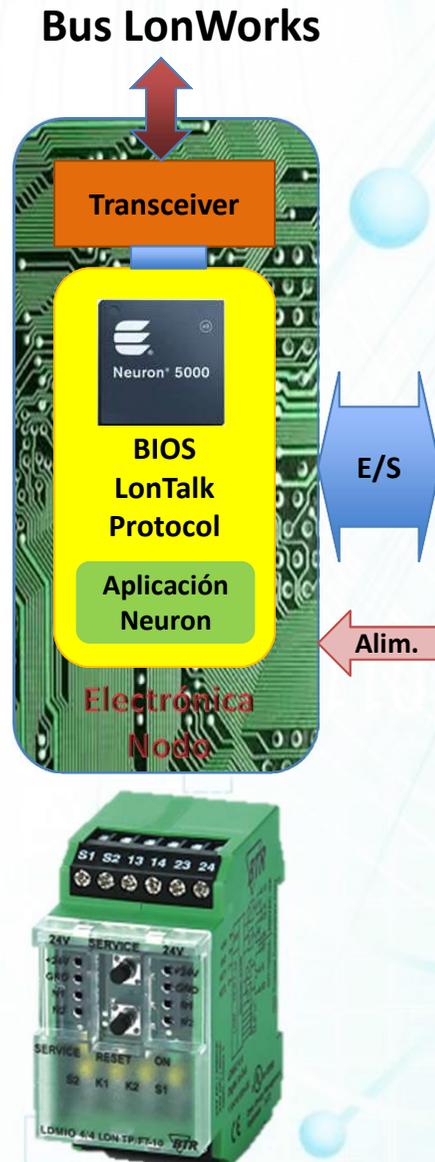
- Comunicaciones fiables entre los nodos.
- Mensajes cortos y precisos.
- Funcionalidad 'peer-to-peer', de igual a igual.
- Dispositivos de bajos costo.
- Sistema abierto a la industria.
- Soporte de diferentes buses físicos.

### Hardware LonWorks

- Basado en un 'chip' llamado **NeuronChip o Neuron**.
- El Neuron incluye **'firmware' / BIOS** con el protocolo implementado.
- **Soportan el protocolo LonTalk** sin que el usuario tenga que conocerlo.

# Sistemas LonWorks

## Elementos: Nodo



### Nodo

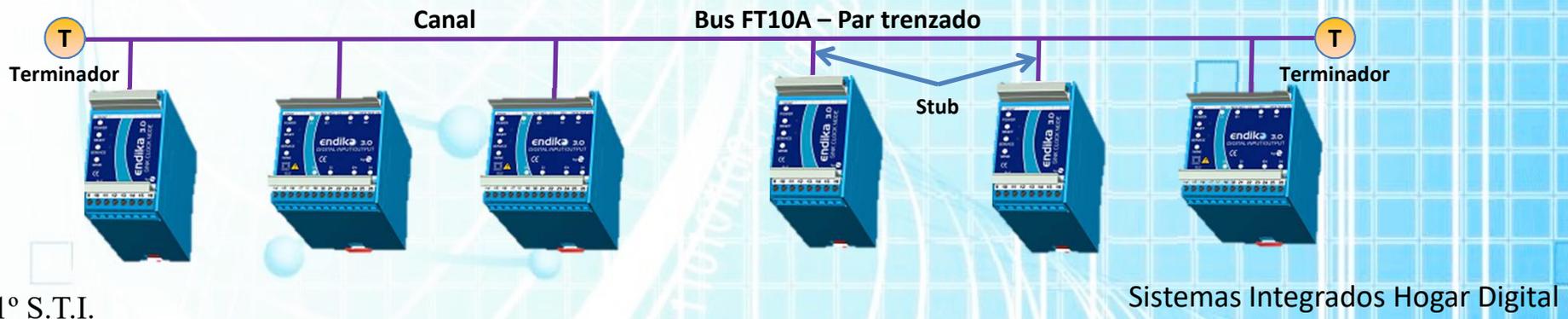
- El **nodo** incorpora un núcleo, el **Neuron**, encargado de ejecutar **labores programadas**.
- Existen **miles de nodos compatibles con LonWorks** programados para **multitud de funciones**.
- Los nodos **se pueden reprogramar** usando un lenguaje de programación llamado **Neuron C**.
- El nodo **incorpora circuitos electrónicos extras** para conectarse a las **entradas / salidas del nodo**.
- El **'transceiver'** es el elemento que **conecta el Neuron con el Bus** que usamos.
- Los nodos de **diferentes fabricantes** admiten **'interoperabilidad'**, es decir, pueden usarse juntos.
- Se **comunican entre sí** mediante mensajes que cumplen el **protocolo LonTalk (EIA709.1)**
- Se pueden usar diferentes tipos de buses en una misma instalación con las pasarelas adecuadas.
- Los nodos usan el **'bus' como medio físico de comunicación**, existiendo diferentes tipos:
  - **Par trenzado** TP/FT10 a 78,5 kb/s y TP/XF1250 a 1,25 Mb/s (el más usado es el **FTT10A**).
  - **Power Line** 'PL20' y 'PL30', usa corrientes portadoras sobre el tendido eléctrico (5 y 2 kb/s).
  - **Redes IP**, usado sobre todo como pasarela entre redes o conexión remota.
  - **Fibra óptica**, 'FO10', para enlaces a **distancias mayores** (hasta 3,2 Km).
  - **Radio, línea telefónica, coaxial, etc.** (menos usados).
- ❖ **Cada medio físico tiene sus limitaciones**
- Con la evolución han ido apareciendo **nuevos procesadores** para los nodos, los primeros en el mercado fueron el **3120 y el 3150**, actualmente se están sustituyendo por el **FT-5000, más pequeño, económico y veloz**, recientemente ha aparecido el **FT-6000** con mejoras sobre el FT-5000 aunque casi no está implementado en nodos.
- Cada **Neuron** incorpora **tres microprocesadores de 8 bits**, dos para las comunicaciones y uno para la aplicación del nodo, de esta forma se consigue **mejorar el rendimiento y aislar el protocolo** de la **aplicación** del nodo.
- Los nodos **vienen de fábrica con una aplicación 'tipo'** para el **hardware** que incorporan (**entradas analógicas, salidas, reloj, etc**), suelen instalarse en el edificio con el **programa 'estándar'** por parte de los **técnicos que realizan las instalación y puesta en marcha** del sistema.
- Cada nodo incluye **documentación completa** con **características y funcionalidad** de la aplicación.

# Sistemas Lonworks

## Características físicas, definiciones y restricciones

**Canal:** Es un *medio físico de comunicación* al cual están *conectados nodos Lonworks usando los transceivers adecuados* para ese tipo de canal. Cada *canal tiene limitaciones físicas de distancia, velocidad, número de dispositivos máximos conectables, etc.*

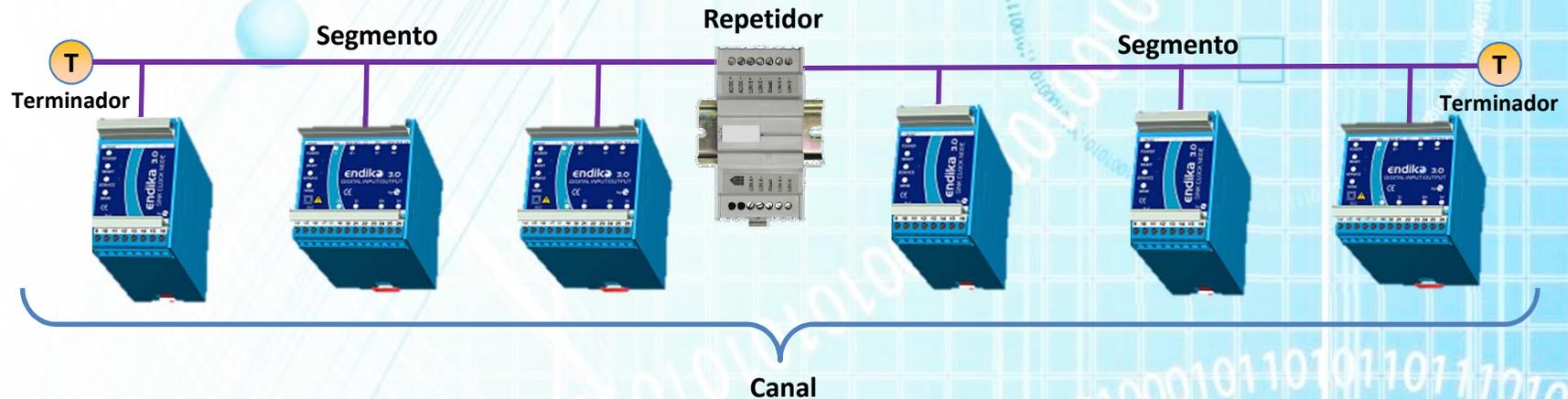
Tipo Canal	Medio	Velocidad	Máx. Distancia	Máx. Nº Nodos	Observaciones
TP/FT-10	Par trenzado Topología libre o bus (*)	78 kbps	500 m. Topología Libre 2700 m. Topología Bus	64	Es el más usado por la simplicidad de su bus como sistema de conexión entre nodos en un edificio.
TP/LPT-10	Par Trenzado Link Power Topología libre o bus	78 kbps	500 m. Topología Libre 2200 m. Topología Bus	128	Menos usado, permite llevar alimentación al nodo (limitada) para alimentar su electrónica.
TP/XF-1250	Par trenzado	1,25 Mbps	130 m.	64	Usado como 'troncal' para unir varios canales
TP/XF-78	Par trenzado	78 kbps	1400 m.	64	Primer tipo de bus Lon, tiende a desaparecer.
PL-20	Línea potencia	5,4 kbps Banda C 3,6 kbps Banda A	Depende del entorno	---	Se usa el tendido eléctrico como bus, no precisa tirada bus, usado en SmartCities.
IP-10	Lonworks sobre IP	10, 100 Mbps 1Gb	Determinado por red IP	---	Usado para conectar instalaciones entre sí o para acceso remoto desde Internet.
FO-20	Fibra óptica	1,25 Mbps	30 kilómetros	---	Para unir instalaciones a larga distancia.



# Sistemas Lonworks

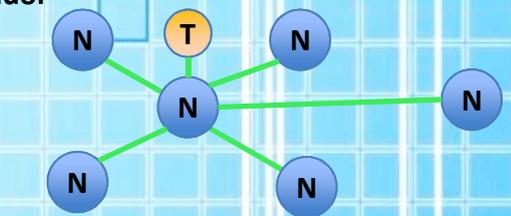
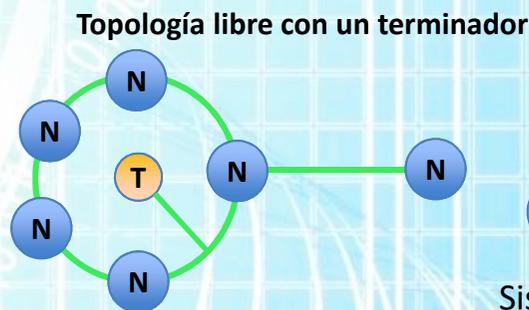
## Características físicas, definiciones y restricciones

**Segmento:** En caso de que un *Canal supere los límites físicos*, se puede *dividir en 'segmentos' intercalando un 'repetidor'*. Si ponemos un repetidor, *estamos en el mismo Canal*. El *repetidor 'amplifica' o 'restaura' las señales en los segmentos*.



**Terminador:** Los *buses de pares trenzados* suelen llevar un *terminador* en los extremos para *evitar 'ecos' de las señales eléctricas* en el bus, *depende de la topología* (forma de la red) como se *situarán los 'terminadores'*.

**Topología:** Forma en la que se *distribuye el bus por una red de nodos*. Cabe distinguir la *topología libre* y la *topología en 'bus'*. La libre, como su nombre indica permite un conexionado libre del bus, en la tipo 'bus', solo existe un bus único que recorre los nodos.

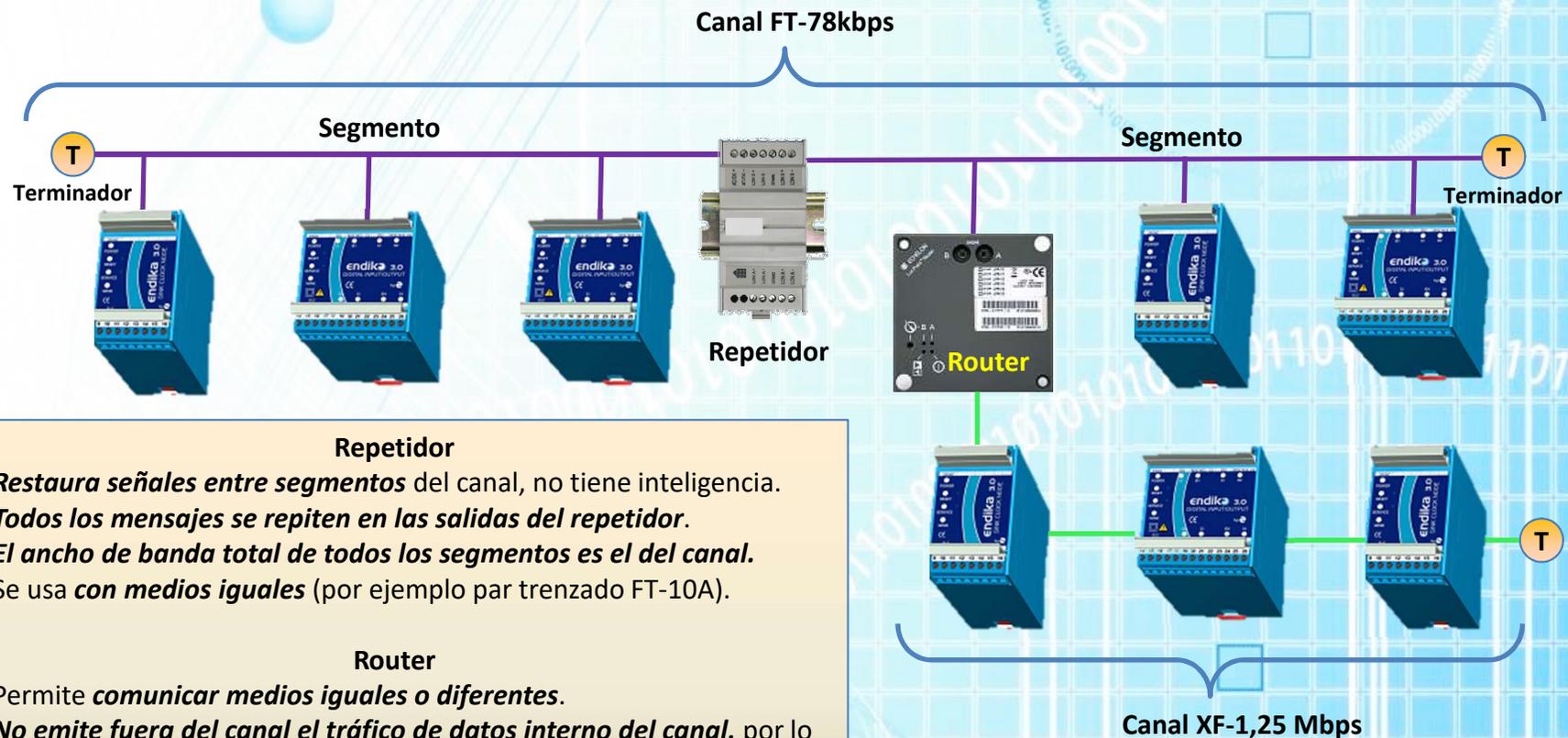


Sistemas Integrados Hogar Digital

# Sistemas Lonworks

## Características físicas, definiciones y restricciones

**Router:** Para *superar limitaciones físicas, aislar el tráfico entre canales o cambiar de medio físico* se pueden usar el 'router', cada *router tiene 'dos caras' con dos Neuronas que transmiten los mensajes entre canales*. Los routers se pueden configurar en diferentes *modos de funcionamiento* con herramientas externa.



### Repetidor

- *Restaura señales entre segmentos* del canal, no tiene inteligencia.
- *Todos los mensajes se repiten en las salidas del repetidor.*
- *El ancho de banda total de todos los segmentos es el del canal.*
- Se usa *con medios iguales* (por ejemplo par trenzado FT-10A).

### Router

- Permite *comunicar medios iguales o diferentes.*
- *No emite fuera del canal el tráfico de datos interno del canal*, por lo tanto, *no transmite los mensajes que no tendrían que salir, conservando cada canal su ancho de banda.*
- *Puede trabajar como repetidor, con tablas o en modo 'aprendizaje'.*

Dos buses diferentes conectados con un router

# Sistemas Lonworks

## Elemento: Interfaces

Tenemos una *instalación LonWorks*, sabemos que existen routers, repetidores, pero: **¿Cómo podemos conseguir conectarnos a un sistema de control desde un PC?**, **¿Cómo podemos conectar sistemas entre sí?**, ¿qué pasarelas podríamos usar?, estamos hablando de **Interfaces a PC o Interfaces a red IP que pueden hacer la doble función**, dar conectividad a un PC o enlazar redes.



# Sistemas Lonworks

## Características físicas, definiciones y restricciones

### Notas

Se han indicado **longitudes de cable máximas para canales** según el **medio usado**, estas longitudes **pueden ser inferiores si no se usan los materiales adecuados**. Se pueden consultar fabricantes y tipos recomendados en la información técnica de **Echelon** o en la asociación **LonMark**.

Por ejemplo:

Cable	Canal	Topología	Longitud
Belden 8471	TP/FT-10	Libre	500 m.
Belden 8471	TP/FT-10	Bus	2700 m.
Anixter 9D220150	TP/XF-1250	Bus	130 m.

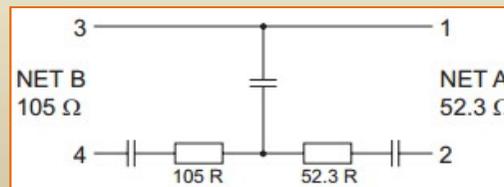
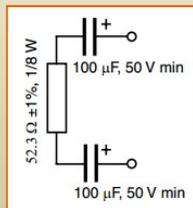
<http://www.echelon.com>

<http://www.lonmark.org>

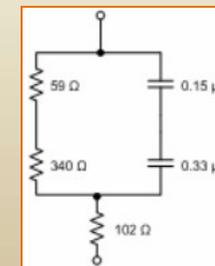
**LonMark** es una **asociación internacional de fabricantes e integradores de sistemas LonWorks** en la que se puede encontrar gran cantidad de **documentación técnica sobre Lonworks**, aparte de **enlaces a los productos de los fabricantes asociados**. En España existe una **Web de LonMark España** <http://www.lonmark.es>, pero para **documentación técnica es recomendable la Web principal**.

En la **Web de Echelon** está **toda la información técnica de sus productos, manuales, recomendaciones de diseño, ejemplos de aplicación, etc.** Para acceder a las descargas es necesario estar registrado, también existe un **forum sobre Lon en Echelon**.

Los **terminadores de red presentarán diferentes impedancias, según se use uno o dos**, si se usan **dos terminadores**, uno en cada extremo, su impedancia será de **105Ω** y en caso de ser **un solo terminador será de 52,3Ω**.



Ejemplos de esquemas de terminadores para bus LonWorks



# Sistemas Lonworks

## Comunicaciones y protocolo

### Direccionamiento

Cada **Neuron** tiene un **identificador único de 48 bits**, no existen dos chips Neuron en el mundo con el mismo NeuronID, este número **se graba en fábrica y es como el 'NIF' del chip**. Se usa básicamente en los **procesos de instalación y mantenimiento** en una instalación, se expresa en **hexadecimal**, por ejemplo: **45:5c:2b:87:ac:05**

Los **nodos se comunican mediante mensajes por el bus**, estos mensajes llevan la **información para que el sistema se coordine y funcione** de forma correcta, para determinar **donde deben dirigirse los mensajes** se pueden usar diferentes tipos de direccionamiento:

- **Dirección física por NeuronID ( identificador del chip):**
  - ✓ **Poco práctico en el uso de la red**, ya que si se cambia un nodo por avería, sería necesario cambiar el Neuron ID en todos los nodos que le envían datos.
  - ✓ Se usa **para la instalación del nodo en el sistema** o red LonWorks.
- **Dirección lógica a nodo:**
  - ✓ Usa como dirección el **conjunto: Dominio / Subred / Nodo (Domain / Subnet / Node)**
  - ✓ Se asigna en el **proceso de instalación de la red LonWorks**.
  - ✓ Es el **modo común de direccionamiento** en las redes instaladas.
  - ✓ Permite **sustituir un nodo** averiado sin tener que modificar ni tocar el resto de la instalación.
- **Dirección lógica a grupo:**
  - ✓ Se transmite el **mensaje a un grupo de nodos** predeterminado, un nodo puede pertenecer hasta a 15 grupos.
- **Broadcast:**
  - ✓ Se envía un **mensaje a todos los nodos** de un dominio o subnet.

❖ Una vez que se establezca la **'red' lógica de mensajes**, los **datos se moverán de un nodo a otro según sea preciso**.

❖ Si **existe un ordenador con un programa SCADA o HMI**, podrá **solicitar a los nodos mensajes con información** de estados, valores, etc. y **podrá enviar a los nodos valores** de configuración, cambios de estados, actuaciones, etc.

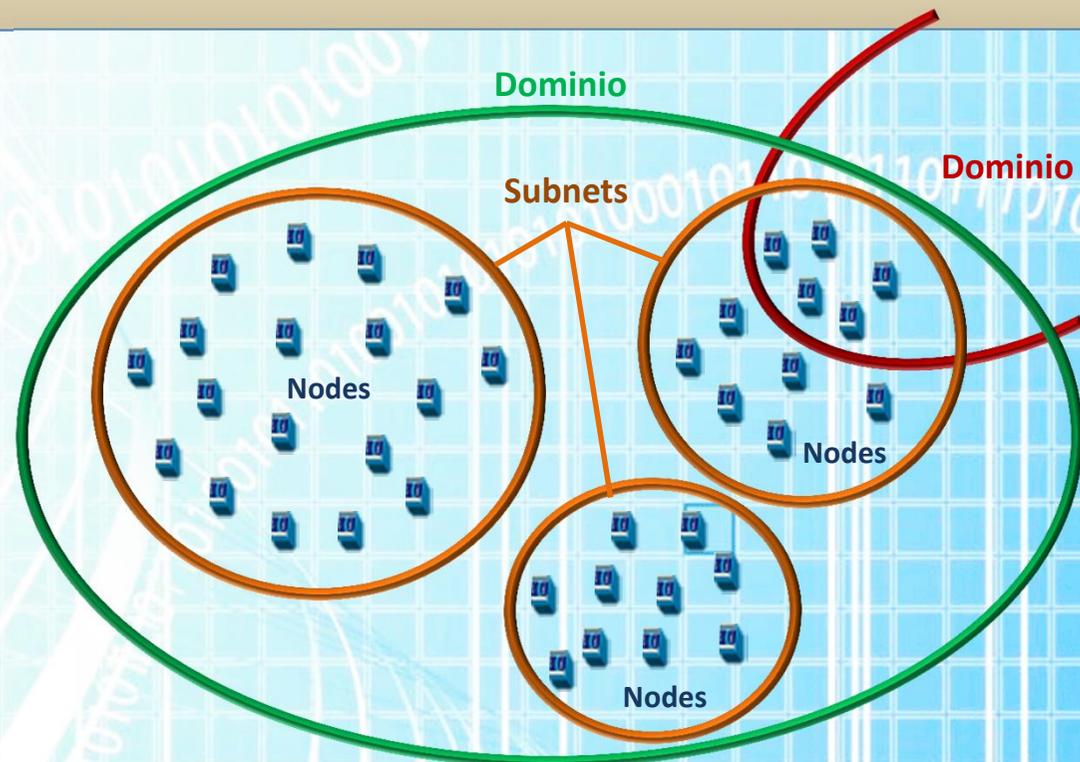
# Sistemas Lonworks

## Comunicaciones y protocolo

### Direccionamiento

- Hemos visto que cada **Neuron tiene un identificador único de 48 bits**, pero que no es práctico usarlo en una red instalada.
- También se ha comentado que el modo más común de direccionamiento es el lógico: **Dominio / Subred / Nodo**, este modo de **organización se asigna al nodo en el proceso de instalación** con las herramientas de software adecuadas.
- La **subred (subnet)** es un conjunto de **hasta 127 nodos** (nodos).
- En un **dominio (domain)** puede haber **hasta 255 subredes** (subnets).
- **Prácticamente no hay limitación de número de dominios** ( $2^{48}$ ).
- Con una multiplicación simple ( $127 * 255$ ) podemos observar que pueden existir hasta **32385 nodos en un dominio**.

- La **dirección física del Neuron** (NeuronID) suele venir **impresa en el nodo**, pero si no es así, podemos **'obligar' al Neuron a que la comunique** activando el **'Service Pin'**.
- La **dirección lógica de un nodo** es la referencia a **Domain / Subnet / Node**, por ejemplo: el **nodo** de la azotea que mide temperatura, humedad y luz exterior podría tener como **dirección lógica: domain=0, subnet=5 y node=7, no habrá otro nodo en la instalación con esa 'dirección'**.
- El **identificador de dominio** puede tener **diferentes tamaños: 0 bytes (null domain), 1 byte, 3 bytes o 6 bytes**.
- **No confundir el cableado de la instalación (red física) con la configuración lógica de la instalación (direcciones lógicas).**



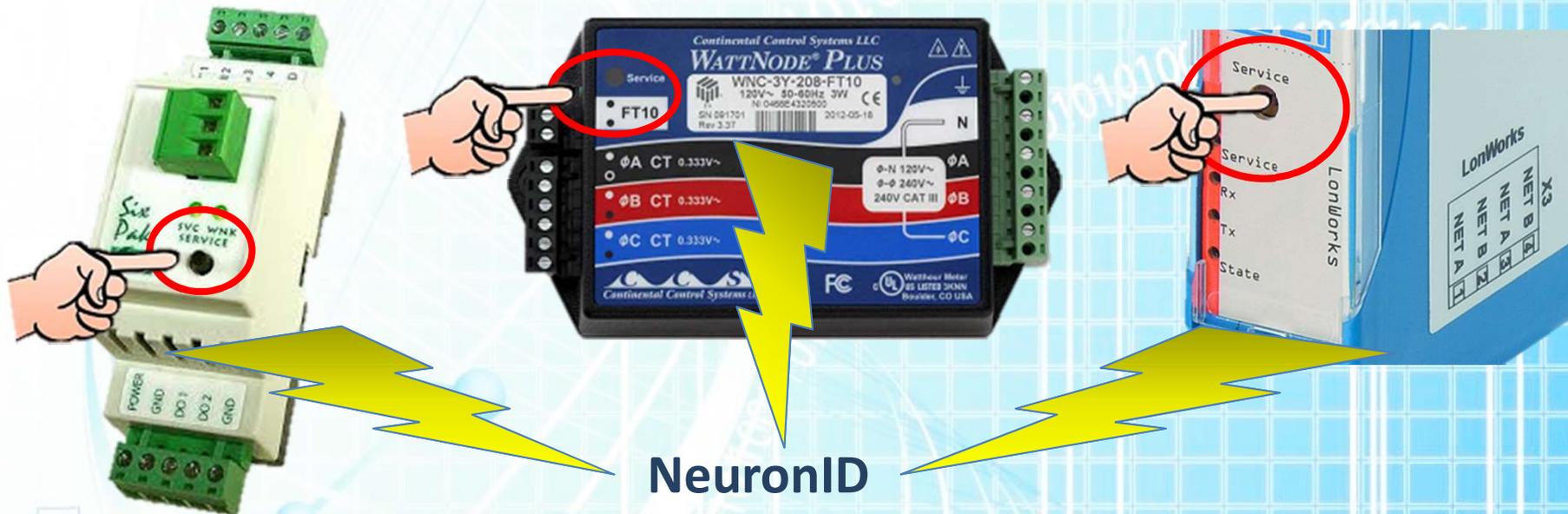
# Sistemas Lonworks

## Comunicaciones y protocolo

### Direccionamiento

- Hemos visto que cada **Neuron** tiene un **identificador único de 48 bits** y también que suele venir **impreso en alguna pegatina** en el nodo, pero de no ser así, **es posible obligar al Neuron a que lo publique**.
- **Todos los nodos Lonworks** tienen en algún lugar **acceso a un pulsador**, normalmente oculto o de difícil acceso, que al pulsarlo **provoca que el Neuron publique por el 'bus' su NeuronID**, esta operación es útil para **identificar un nodo en la red** cuando hay decenas o cientos de ellos y se van a usar **herramientas para configurarlo** y **asignarle un dominio, subnet, número de nodo y otros parámetros**.

**Nota:** Un nodo sin instalar no tiene asignado direccionamiento lógico (domain / subnet / node ).



El NeuronID se usa para 'comisionando' (integrar nodo en red y configurarlo), creando uniones (bindings) y para monitorizar su estado (\*).

# Sistemas Lonworks

## Comunicaciones y protocolo

### Tipos de servicio

- Las **comunicaciones de los mensajes** pueden realizarse con **diferentes tipos de 'servicio'** en cuanto a la **forma que se transmiten y se gestionan los 'acuses de recibo'** de la transmisión:
- **Mensaje con acuse de recibo (Acknowledged Messaging) :**  
Cuando **se envía un mensaje** (packet) a un nodo o a un grupo de nodos, se espera que **cada nodo** que ha recibido la información **'emita' un acuse de recibo**, si **no se produce el acuse de recibo**, se **volverá a repetir el envío** hasta un número de veces determinado (por defecto tres intentos). Esta forma de **'mensajería entre nodos' es la más común.**
- **Mensaje repetido (Repeated Messaging):**  
Un mensaje se **transmite repetidas veces** (para asegurarse en un mayor grado la recepción).
- **Mensaje sin acuse de recibo (Unacknowledged Messaging) :**  
Se **transmite un mensaje**, pero **no se espera acuse de recibo** por parte del receptor o de los receptores. Usado para pasar una **información a muchos nodos y no tener que gestionar multitud de acuses de recibo con gran tráfico.**

**Nota:** Se puede **combinar el servicio Repeated con Unacknowledged**, se transmite un **mensaje varias veces y no se espera respuesta** del receptor o los receptores, **por ejemplo un nodo 'reloj' transmite la fecha/hora de esta forma cada cinco minutos a 200 nodos y no hay que gestionar 200 acuses de recibo y también es muy difícil que un nodo no escuche alguna de las transmisiones repetidas.**

- 
- **Mensaje autenticado (Authenticated Messaging):**  
Permite **establecer una clave de autenticación de identidad** en las comunicaciones de la red para garantizar que el **nodo que emite un mensaje está 'autorizado'** a hacerlo, es necesario que la **clave sea la misma**, por defecto los mensajes no se envían con autenticación. No es común usar este modo **en instalaciones 'privadas', es decir sin acceso a extraños.**

# Sistemas Lonworks

## Comunicaciones y protocolo

### Mensajes en LonWorks

- Hemos comentado que los *nodos se intercambian mensajes entre sí*; casi siempre, estos mensajes suelen contener *datos 'técnicos'* del tipo *'temperatura', 'on/off', 'amperios de corriente', 'litros/segundo', 'peso', etc.*
- Para hacer posible la *comunicación correcta de estos datos* y asegurar que *TODOS los nodos transmiten y reciben esos datos* de la *misma forma* aunque los *fabricantes sean diferentes* se creó una *lista de datos estándar llamados SNVTs o 'variables de red'*.
- La *mayoría de los mensajes de una red LonWorks* llevan datos (de ahora en adelante **VARIABLES**) *del tipo SNVT (Standar Network Variable Type)*, veamos algunos ejemplos de estos tipos de variables:

Idx	Tipo	Descripción	Rango	Resolución
1	SNVT_amp	Intensidad de corriente (amperios)	-3276,8 a 3276.8 A	0,1 Amp
84	SNVT_time_stamp	Fecha y hora	1/1/1 al 31/12/3000	1 segundo
39	SNVT_temp	Temperatura en grados Celsius	De -274º a 6261,5º	0,1º
63	SNVT_temp_f	Temperatura en grados Celsius (en 'coma flotante')	-273,17º a 3,40*10 <sup>38</sup>	----
69	SNVT_flow	Caudal expresado en litros por segundo	0 a 65535 l/sg	1 l/sg.

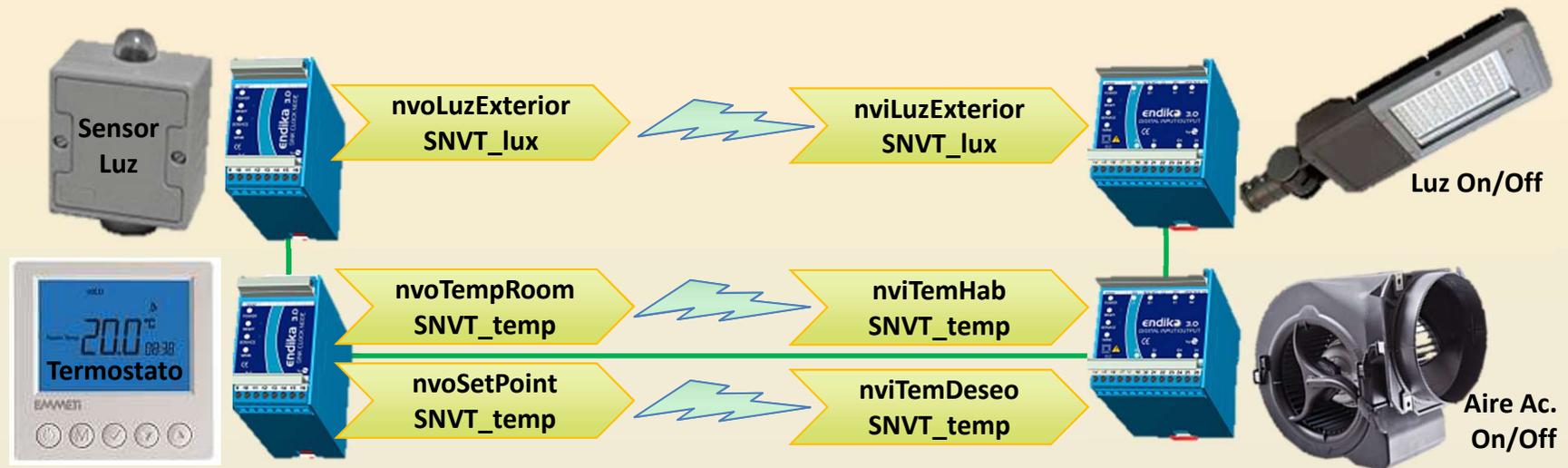
- El tipo *SNVT\_xxxx* nos indica un *formato para intercambio* de información, un *nodo puede tener asignadas varias variables* del mismo tipo, *para distinguir unas de otras usaremos nombres los nombres que le asigna el programador del nodo*, por ejemplo, en un nodo puede existir la variable *temp\_aguafria* y la variable *temp\_aguacal* y ambas ser del *tipo SNVT\_temp*.
- Los *mensajes y los nodos trabajan con el Idx (nº indice del tipo de variable)* pero para las *personas es más intuitivo* trabajar con su *nombre de tipo de variable*: tipo *84 es SNVT\_time\_stamp*.
- Para obtener *la lista completa de variables estándar definidas (SNVT)* para LonWorks nos conectaremos con la web de LonMark ([www.lonmark.org](http://www.lonmark.org)) y en *'recursos técnicos'* podremos encontrar *la lista más reciente de variables*.

# Sistemas Lonworks

## Comunicaciones y protocolo

### Variables de red Lonworks: SNVT

Veamos unos *ejemplos simples de comunicación usando variables SNVT*, tenemos *cuatro nodos con diferentes funciones*, el bus está representado por la línea verde:



El **nodo que mide la luz exterior** ofrece una variable llamada `nvoLuzExterior` del tipo `SNVT_lux`.

El **nodo** conectado al **termostato de habitación** ofrece **dos variables** que informan de la **temperatura de la habitación** (`nvoTempRoom`) y otra que informa de la **temperatura deseada** (`nvoSetPoint`), ambas son del **tipo** `SNVT_temp`.

El **nodo que controla la luminaria** tiene una **entrada de variable** llamada `nviLuzExterior` para que le **informen de la luz exterior** y **decidir** si se activa la luminaria.

El nodo que controla el A.A. de **habitación** tiene **dos variables para recibir la temperatura de la habitación y la temperatura deseada** (`nviTemHab` y `nviTemDeseo`), ambas del tipo `SNVT_temp`.

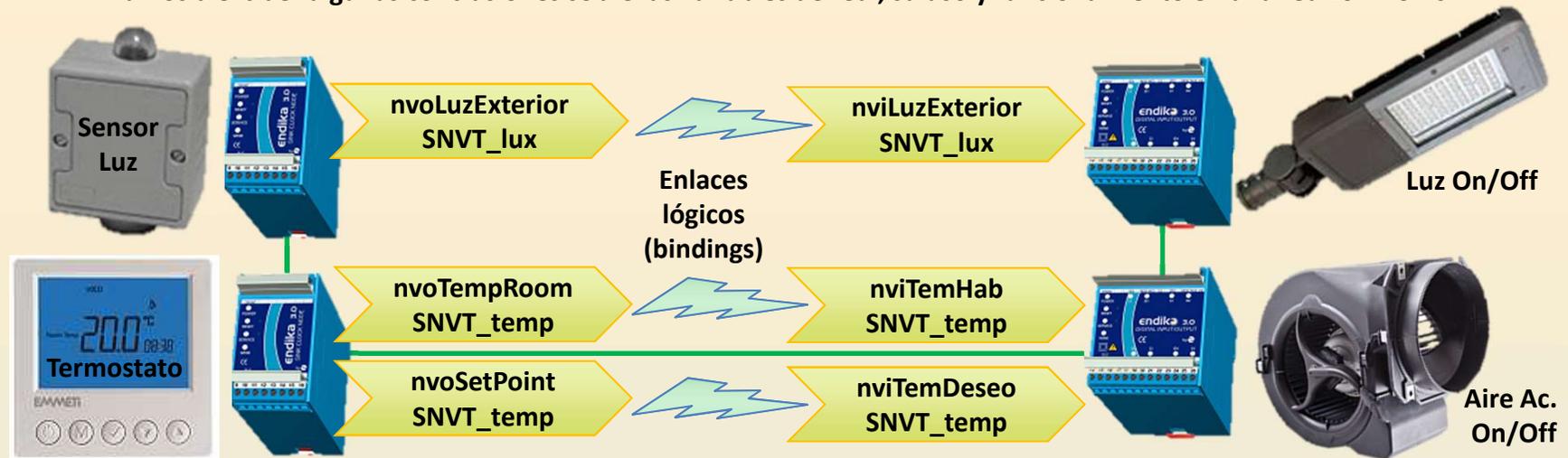
Mediante el **protocolo LonWorks**, se harán **llegar los valores de variables origen a las variables destino**.

# Sistemas Lonworks

## Comunicaciones y protocolo

### Variables de red LonWorks: SNVT

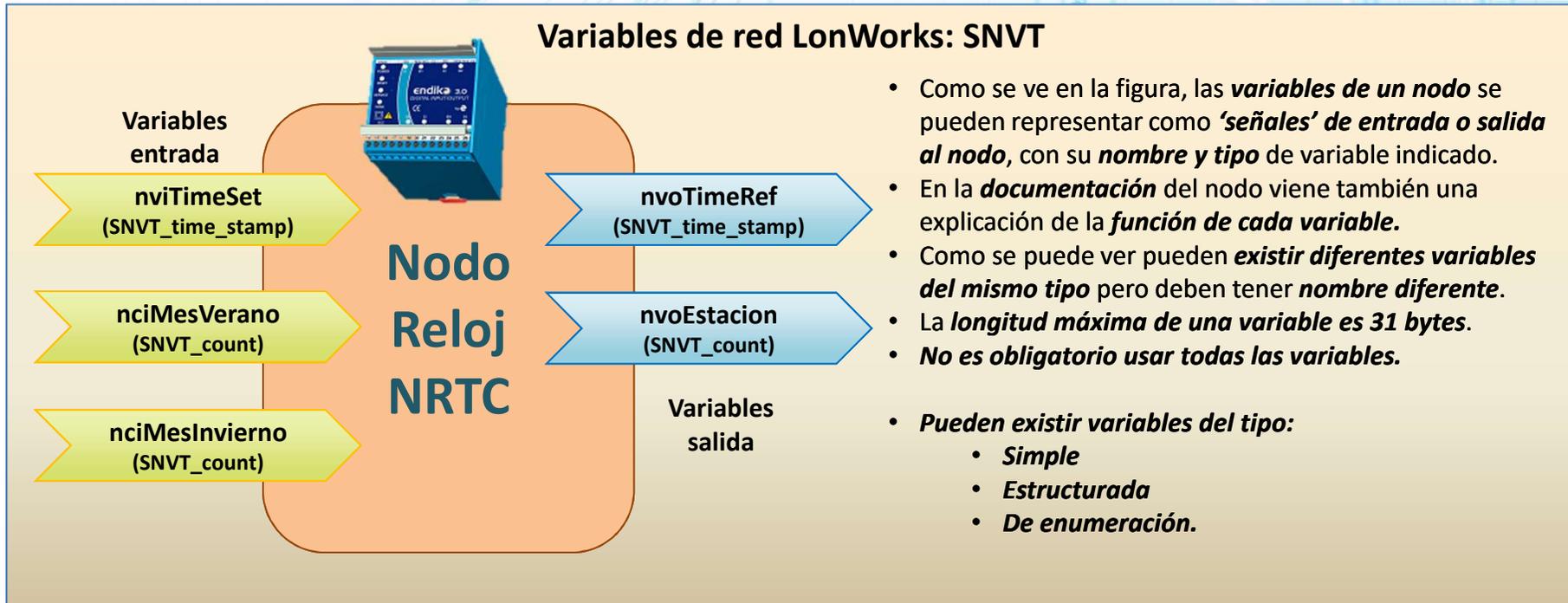
Vamos a extraer algunas conclusiones sobre las variables de red , su uso y funcionamiento en una red LonWorks



- Las **variables vienen ya con el nodo** y tienen un **nombre y un tipo**, el **nombre lo determina el fabricante** y el tipo es del estándar.
  - El **nombre de la variable y el tipo lo elige el que realiza el programa del nodo y suele estar relacionado con su función**.
- Existen **variables de salida** (entregan información en la red) y **variables de entrada** (esperan información desde la red).
  - Las variables en los nodos **se definen como de entrada o salida, nunca son de entrada/salida**.
- Las **comunicaciones se establecen entre variables de salida y variables de entrada del mismo tipo**.
  - **No** se pueden **comunicar variables de diferente tipo** (formato) por razones obvias.
  - **No** se puede comunicar una variable del tipo **SNVT\_temp** con una del tipo **SNVT\_temp\_f** ya que tienen formatos diferentes aunque ambas sean de temperatura.
- Es **necesario asociar, enlazar, determinar que variables se comunican entre si** (bindings).
  - En el **proceso de creación de la red** y mediante **herramientas software se definen los enlaces o bindings**, de esta forma se **determina que variables se conectarán entre si**, existe la posibilidad de **conexiones múltiples**, por ejemplo, la variable **nvoLuzExterior** del nodo de la imagen se podría **comunicar a 40 nodos diferentes que tengan una entrada del tipo SNVT\_lux**.
  - Los **enlaces o 'bindings'** se guardan en la memoria **EEPROM del Neuron** para que **no se pierdan al quitar la alimentación**.

# Sistemas Lonworks

## Comunicaciones y protocolo



### Variable simple

El **contenido de la variable expresa el valor del parámetro enviado**: Por ejemplo, una variable tipo **SNVT\_temp** expresa **temperatura en grados Celsius**. **nvoTemExt -> -10º C**

### Variable estructurada

El **contenido de la variables es una estructura con diferentes valores**: Por ejemplo, una variable tipo **SNVT\_time\_stamp** tiene diferentes valores: **year, month, day, hour, minute y second**. **nvoFHora: 25/11/2014 17:35:23**

**nvoFHora.Year=2014, nvoFHora.Month=11, nvoFHora.Day=25, nvoFHora.Hour=17, nvoFHora.minute=35 y nvoFHora.second=23**

### Variable enumerada

El **contenido de la variable sirve para traducirlo de una lista**: Por ejemplo una variable tipo **SNVT\_lev\_disc** puede tener solo **cinco valores diferentes** con **significados concretos**: **-1: ST\_NUL, 0:ST\_OFF, 1:ST\_LOW, 2:ST\_MED, 3:ST\_HIGH y 4:ST\_ON**

# Sistemas Lonworks

## Comunicaciones y protocolo

### Variables de red LonWorks: SNVT

#### Ejemplos de variable simple

Tipo	Descripción	Tamaño	Valores	Resolución
SNVT_rpm	Revoluciones por minuto	2 bytes	0 a 65535	1 rpm
SNVT_lev_cont	Nivel en %	1 byte	0 a 100%	0,5%
SNVT_density	Densidad en kg/m <sup>3</sup>	2 bytes	0 a 32767	0,5 kg/m <sup>3</sup>

#### Ejemplo de variable estructurada

Tipo	Dato estructura	Descripción	Tamaño	Valores
SNVT_reg_val		Valor de registro	6 bytes	
	SNVT_reg_val.raw	Valor a registrar	32 bits	-2147483648 a 2147483647
	SNVT_reg_val.unit	Tipo unidad registrada	1 byte	-1 a 43
	SNVT_reg_val.nr_decimals	Nº decimales del valor	1 byte	0 a 7

#### Ejemplo de variable enumerada

SNVT_date_day – 1 byte								
Valor	-1	0	1	2	3	4	5	6
Significado	No válido	Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado

- Buscar variables de diferentes tipos en la lista de [lonmark.org](http://lonmark.org)

Nota: Sean del tipo que sean, las variables solo requieren un enlace o 'binding' para 'conectarse'.

# Sistemas Lonworks

## Comunicaciones y protocolo

### Variables de red LonWorks: SCPT

Existe **otro tipo de variables** muy importantes en el **protocolo Lonworks**, las llamadas **SCPT (Standard Configuration Property Type)**, son **similares a las SNVT**, la única diferencia es que estas variables se usan para distinguir **parámetros de funcionamiento de un nodo** que suelen **cambiar muy poco**, solo **cuando se configuran** y además, en muchos casos se guardan en **memoria EEPROM** para que **no se pierda su valor cuando se retira la alimentación del nodo** por alguna causa. La gran mayoría de estas variables hacen referencia a valores en formato SNVT.

Existe una '**master list**' de este tipo de variables en **lonmark.org**, veamos unos ejemplos:

Tipo	Descripción	Formato
SCPTlimitCO2	Límite de CO2 permitido	SNVT_ppm
SCPTclOnDelay	Retardo para poner salida On	SNVT_time_sec
SCPTcoolSetpt	Valor por defecto para modo frio	SNVT_temp_p

Como se puede ver, en todos estos ejemplos, las variables de configuración usan formatos de SNVT y sirven para configurar **valores que no cambian con frecuencia**: límite de CO2, retardo para una salida, temperatura para un modo de trabajo, etc.

### Variables de red LonWorks: definidas por el usuario

Se puede dar el caso de que un **fabricante de un nodo o un programador no encuentre ningún tipo de variable que se ajuste a sus necesidades**, en este caso puede definir **sus propios tipos** que **NO SERÁN COMPATIBLES CON OTROS FABRICANTES** pero que los programas **SCADA / HMI podrán tratar y leer ya que el fabricante suministrará su estructura interna. Se les llama UNVT\_xxx (user ...)**

Por ejemplo: **UNVT\_info\_ali** (16 bytes), información fecha elaboración y caducidad de alimento, **estructura:**

**UNVT\_info\_ali.codigo**: **unsigned long** (2 bytes), código del artículo / alimento, entre 0 y 65535.

**UNVT\_info\_ali.fec\_ela**: **SNVT\_time\_stamp** (7 bytes), fecha de elaboración del artículo.

**UNVT\_info\_ali.fec\_cad**: **SNVT\_time\_stamp** (7 bytes), fecha de caducidad del artículo.

# Sistemas Lonworks

## Comunicaciones y protocolo

### Variables de red LonWorks

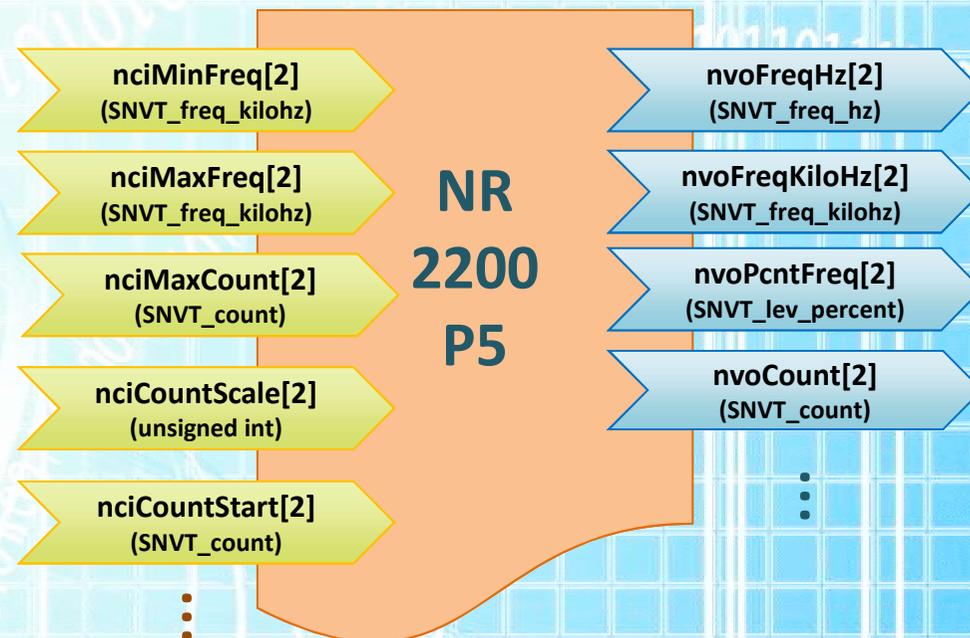
Como se habrá podido deducir de todo lo visto hasta ahora, *cada nodo tiene un programa y cada programa maneja una serie de variables de diferentes tipos*, esas variables estarán *descritas en la documentación técnica del nodo* y tendrán que ser *interpretadas por el integrador para poder usar el nodo y configurarlo de forma correcta*.

Por ejemplo, un nodo con dos entradas de impulsos de la firma NEUROLOGIC RESEARCH CORPORATION, el modelo 2200 ([http://www.neurologic-research.com/model\\_2200.htm](http://www.neurologic-research.com/model_2200.htm)) tiene un manual que se puede descargar de Internet y en el que encontraremos todos estos datos, aparte de otra información necesaria para su instalación:

<http://www.neurologic-research.com/Datasheets/2200bro0.PDF>

- En la **página Web del fabricante** suelen estar los **datos técnicos** que nos permitan **usar sus nodos**, entre todos estos datos están las **variables usadas**.
- También **se encontrarán instrucciones de montaje, una descripción del programa, etc.**
- Es frecuente que un **mismo nodo (hardware)** pueda usarse con **diferentes programas**, en este caso será necesario que el **fabricante proporcione estos programas**.
- El **nodo comentado aquí tiene diferentes aplicaciones** según el **programa que se cargue**, dando lugar a los modelos: **2200P5, 2200PM y 2200VM**
- En **la imagen de la derecha** se puede ver **parte del gráfico de variables** de la documentación del nodo.

❖ [Buscar documentación del WattNode](#)



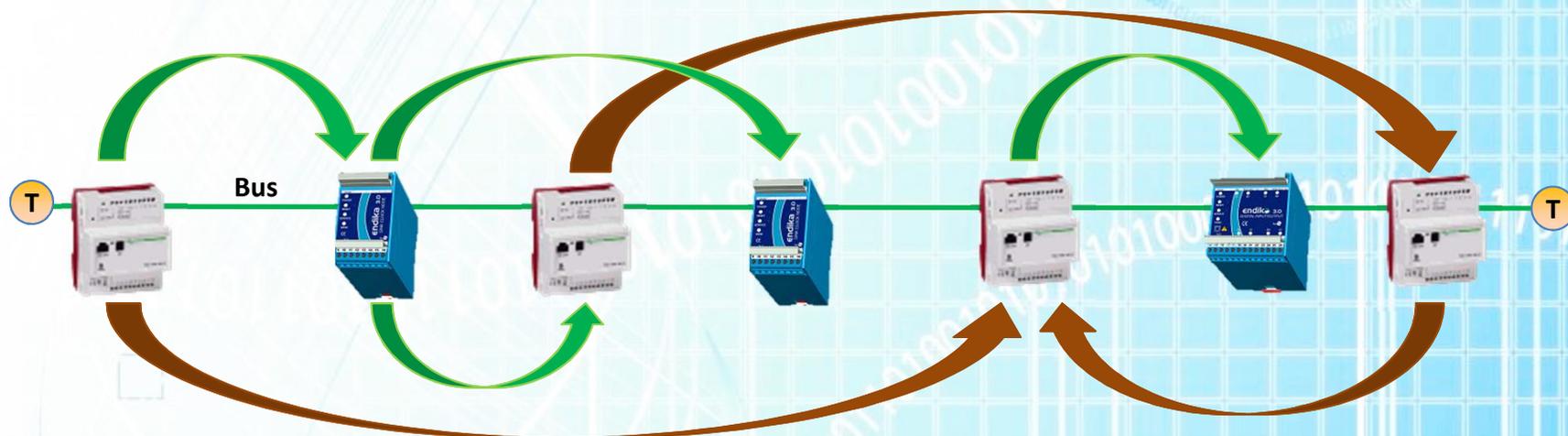
# Sistemas Lonworks

## Comunicaciones y protocolo

### Otros formato de mensajes LonWorks

El modo más común de comunicación en los nodos LonWorks es mediante variables SNVT estándar, es posible también que un fabricante o programador *construya sus propias comunicaciones usando 'mensajes explícitos'*, estos mensajes tendrán el *formato que desee el programador* y hacen *esa comunicación entre nodos un sistema 'no abierto'*, algunos fabricantes usan este tipo de mensajes para montar *sistemas 'propietarios'*, se pueden mezclar estos mensajes con variables estándar.

Por ejemplo, los productos de *TAC /Schneider* usan *mensajes explícitos para comunicarse entre sí* pero también permiten ciertas *comunicaciones con dispositivos de otros fabricantes mediante SNVTs*.



➔ Mensajes explícitos, solo los interpretan los que conocen el 'idioma'.

➔ Mensajes implícitos, usando variables estándar se mantiene la interoperabilidad entre fabricantes

*LonWorks también permite la transmisión de ficheros entre nodos mediante un protocolo de transferencia*, se usa en situaciones especiales donde *se quiere transmitir mayor cantidad de información*. No podemos perder de vista que las velocidades de *transferencia de datos de LonWorks no está orientadas al intercambio de ficheros*, pero puede ser *útil en situaciones puntuales* (un registro de datos de un nodo, lectura de una tarjeta con más de 31 bytes, etc.).

# Sistemas Lonworks

## Comunicaciones y protocolo

### Resumen de limitaciones y capacidades

En la siguiente tabla resumen tenemos los límites de organización de las instalaciones LonWorks:

- **Nodos (devices) en una subred (subnet):** 127
- **Subredes (subnets) en un dominio (domain):** 255
- **Nodos (devices) en un dominio (domain):** 32,385
- **Dominios en una red Lon:**  $2^{48}$
- **Máximo nº dispositivos en un sistema:**  $32K * 2^{48}$
- **Miembros en un grupo**
  - ✓ Sin recibí/repetido (Unacknow./Repeated) Sin límite
  - ✓ Con petición de recibí (Acknowl.) 63
- **Grupos en un dominio:** 255
- **Canales en una red:** Sin límite
- **Bytes en una variable de red:** 31
- **Bytes en mensajes explícitos:** 228
- **Bytes en un fichero de datos:**  $2^{32}$

### Entradas/salidas especiales: reset / service

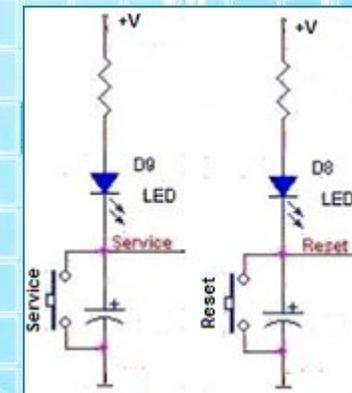
Los nodos tienen dos pines, **Reset y Service**, que tienen funciones especiales. *Suelen tener dos pulsadores y dos LEDs.*

**Pulsador Reset:** Reinicia el nodo (reset por hardware).

**Pulsador Service:** Provoca que se *emita* por la red el **NeuronID** y el **nombre de la aplicación en el nodo**.

**LED de Servicio y LED de Reset:** La combinación de estos dos indicadores nos da información del **estado del nodo**.

Estado	LED Reset	LED Servicio
Normal	Apagado	Apagado
Orden Reset	Encendido	Apagado
Error nodo	Parpadeando	Apagado
Error nodo	Apagado	Encendido
Desconfigurado	Apagado	Parpadeando



# Sistemas Lonworks

## Estados del nodo

### Resumen de estados posibles de un nodo

Un nodo, que funcione correctamente, puede estar en cuatro estados:

- **Applicationless**

- ✓ Si **no tiene ninguna aplicación** (programa de usuario) cargado o se está cargando en ese momento una aplicación.
- ✓ Si hay un **'checksum error'** (error de suma de chequeo) que determina que la aplicación no se ha cargado correctamente.

- **Unconfigured**

- ✓ **Tiene aplicación**, pero **no se ha cargado en el ninguna configuración (comunicaciones, enlaces, etc.)**, en este estado, la aplicación interna se puede ejecutar, no ejecutar o ejecutar partes, el programador lo decide.
- ✓ Este estado se produce también si hay un **error de carga de configuración** (checksum error).
- ✓ **Desconexión lógica de la red, ninguna comunicación y led 'Service' parpadeando.**
- ✓ Para salir de este estado usaremos herramientas software **que identifiquen al nodo en la red y realicen los enlaces de variables necesarios.**

- **Offline**

- ✓ **El nodo tiene aplicación y está configurado** (identificado en red, enlaces, etc.) pero **no se ha pasado a estado OnLine.**
- ✓ Para salir de este estado lo haremos la misma herramienta de software de configuración de red / nodos.
- ✓ En este estado se pueden ejecutar algunas partes del programa, todas o ninguna, el programador lo decide.
- ✓ **No se comunica con la red.**

- **Configured**

- ✓ **Tiene aplicación y configuración correcta. Conexión lógica a red y puede tener comunicaciones.** LED servicio 'off'.

- **Online**

- ✓ **Tiene aplicación, está configurado y se ha puesto en modo 'online', con comunicaciones de red**

❖ **Nota:** El modo normal de operación es **'Configured-Online'**, se puede **ver el estado** del nodo **con varias herramientas software.**

# Sistemas Lonworks

## Archivo XIF

A lo largo de esta presentación, hemos visto algunos conceptos que es necesario repasar para continuar:

- Cada nodo tiene un **programa único** que le permite realizar unas **labores específicas**.
- Cada nodo tiene un **'transceiver'** que puede ser de **diferentes tipos y con diferentes configuraciones de comunicación**.
- Cada **nodo tiene una serie de variables de entrada y salida que son específicas de ese nodo**.
- Las **variables de los nodos tienen unos nombres y tipos** que es importante conocer.
- Los **nodos tienen que ser instalados en la red** y sus **variables tienen que enlazar con otros nodos**.
- Los **enlaces de variables tienen que ser exactos**, es decir, **no se pueden enlazar variables de diferentes tipos**.
- Puede **haber diferentes versiones de software para un mismo nodo** (más modernas) que se **comporten de forma distinta**.

### Conclusiones

Las **herramientas que vayan a configurar los nodos** en una instalación tiene que **'conocer' todas las características de un nodo** para poder **usarlo**, estas características estarán en un **archivo conocido como archivo XIF** que **cada tipo de nodo debe tener**, contiene:

- **Información básica del dispositivo (tipo de Neuron, transceiver, buffers, etc.)**
- **Lista de objetos Lonmark (variables de red SNVTs, SCPT, etc.)**
- **Nombres y descripción de variables de red y etiquetas de mensajes explícitos.**
- **Propiedades de configuración del nodo (CPs).**
- **Identificación del programa** (aplicación en el nodo).

Por ejemplo, **un nodo que 'da la hora'** tiene un **tipo de transceiver (FT10A)**, **unas variables de red** (la salida de fecha / hora, la entrada de puesta en hora, etc.), tiene también un **software** que puede tener una versión, etc. Todos estos datos van en un fichero de **'intercambio de información (eXchange Informatio File)**, conocido como **fichero XIF**.

Como **precaución lógica**, si el **contenido del archivo XIF y el nodo no coinciden**, **no se podrá instalar el mismo**, ya que estaríamos haciendo **operaciones no válidas**, por ejemplo **configurando un nodo con diferente versión software o diferentes variables de red**.

**Nota:** En caso de **no tener este fichero**, se puede **extraer de un nodo parte de esta información y construir un fichero XIF**, pero **no será tan completo como el del fabricante**, por ejemplo, **no aparecerán los nombres de las variables de red**.

# Sistemas Lonworks

## Herramientas software gestion red LonWorks

En varias ocasiones se ha mencionado que se usarán herramientas para la creación, configuración, puesta en marcha, etc. de una red de nodos **LonWorks**.

La **herramienta de integración más usada y creada por Echelon** para estas acciones es el **software llamado LonMaker**.

Esta herramienta permitirá:

- En cuanto a las **instalaciones proyectadas o realizadas**:
  - ✓ **Crear proyectos / redes LonWorks en nuestro ordenador**, basándonos en los diseños de red realizados.
  - ✓ **Gestionar posteriormente esto proyectos** (ampliaciones, modificaciones, mantenimiento, etc.)
  - ✓ **Mantener una copia de seguridad de la instalación** (en poder del integrador / instalador).
- En cuanto a la **integración y puesta en marcha de una instalación**:
  - ✓ **Asignación del direccionamiento lógico de red (domain / subnet / node).**
  - ✓ **Puesta del nodo en estado Online o pasarlo a Offline temporalmente, etc.**
  - ✓ **Verificación de acceso al nodo, estado, errores, lectura y escritura de variables, etc.**
  - ✓ **Descarga de enlaces (bindings) a los nodos** para que se **establezcan las comunicaciones.**
  - ✓ **Configuración y puesta en marcha de routers con modo de funcionamiento asignado.**
  - ✓ **Sustitución de nodos averiados.**
  - ✓ **Actualización o descarga de software de aplicación a nodos.**
  - ✓ **Ampliación de nuevos nodos en la instalación.**
  - ✓ **Bajas de nodos en la instalación**
  - ✓ **Gestión de los archivos XIF**
  - ✓ **Gestión de 'PlugIns'.**
  - ✓ **Copia de seguridad** de la base de datos de la instalación.
  - ✓ Etc.
- En casos puntuales, **reconstrucción de una base de datos perdida** partiendo de la **instalación existente**.



**Con esto finalizamos la introducción a redes LonWorks, el siguiente capítulo recogerá aplicaciones prácticas de estas instalaciones.**