

Red de nodos inteligentes basados en las normas IEEE 1451 como una estructura de información

Ferraro Matías¹; Gallina Sergio H²; Villagran Luis D.³;
Beltramini Paola⁴; Moreno Juan P.⁵; Aranda Marcos⁶

Departamento de Electrónica
Facultad de Tecnología y Ciencias Aplicadas,
Universidad Nacional de Catamarca

1: FTyCA, UNCa, matiasferraro@yhoo.com.ar

2: FTyCA-UNCa, sgallina@tecno.unca.edu.ar

3: FTyCA, UNCa, dvillagran@tecno.unca.edu.ar,

4: FTyCA, UNCa, pbeltramini@tecno.unca.edu.ar

5: FTyCA, UNCa, juanpablomoreno@gmail.com

6: FTyCA, UNCa, marcos_dario_1@hotmail.com

Abstract. El presente trabajo describe el desarrollo de una red de nodos basada en una estructura lógica de lista, con la explicación correspondiente de la selección de esta estructura entre otras. Además son considerados, el protocolo de comunicación a usar en la red y las funciones básicas que brinda la capa de red a la capa de aplicación del nodo. Finalmente, se resume las ventajas de esta red.

El objetivo que se busca es el establecimiento de una red de información que permita la interacción de nodos, caracterizada por no requerir altas velocidades ni gran ancho de banda, con una cantidad acotada de nodos y de fácil instalación.

Los resultados indican que la combinación de Internet 0, que facilita el internet sobre cualquier medio y la abstracción de pensar a un nodo como una unidad de información, identificada por una dirección, permiten el desarrollo de esta red apta para permitir que nodos independientes interactúen, con un mínimo de recurso de hardware y de software.

Palabras Clave: Red de Nodos, Listas, Internet 0, IEEE 1451

1 Introducción

Para la implementación de una red de nodos, se ha desarrollado un nodo inteligente [1] basado en los estándares IEEE 1451. Cada nodo contiene los datos y los procedimientos necesarios para el control de uno o varios dispositivos, lo que les permite funcionar como un sistema distribuido sin depender de servidores centrales, no obstante estos nodos pueden interactuar entre sí, mediante la implementación de una red de baja velocidad a los efectos de reducir costos y simplificar la infraestructura física. Por ello nos proponemos determinar una estructura de red simple con baja carga de software para la transferencia de información entre los nodos.

Entre los protocolos posibles de implementar, que cumplen con las condiciones de diseño que han sido establecidas, se ha optado por Internet 0 [2] [5] sobre una red

cableada de dos hilos. El internet 0 se diferencia de otros protocolos por la duración de un bit y su velocidad de propagación, basándose en que: si un bit es menor que el tamaño de una red, entonces es necesario adaptar impedancias para eliminar los reflejos, por otro lado, si un bit es de mayor tamaño que la red, los transmisores y receptores se pueden construir de manera más sencilla y económica.

Siguiendo con este planteo, analizamos las estructuras lógicas conocidas, que nos permiten la búsqueda de información. La lista doblemente enlazada resulto la más conveniente y por el contrario, la menos conveniente resulto la estructura de Árbol, ya que la misma contempla la utilización de un nodo raíz, que debe estar representado por un servidor.

Nuestro estudio se fundamenta en la consideración de diferentes aspectos. El marco teórico está integrado por: Estructura de Información, Nodos Inteligentes e Internet 0.

2 Estructura de información. Listas

Entre los diferentes tipos de estructura de la información, podemos mencionar los árboles, las pilas, las colas y las listas, entre otras. De estas estructuras nos centraremos en las listas y dentro de ellas en las listas doblemente enlazadas (figura 1). Estas nos interesan particularmente porque tienen un enlace con el elemento siguiente y con el anterior, característica que permite recorrer la lista en ambos sentidos, ya sea para efectuar una operación con cada elemento o para insertar, actualizar y borrar.

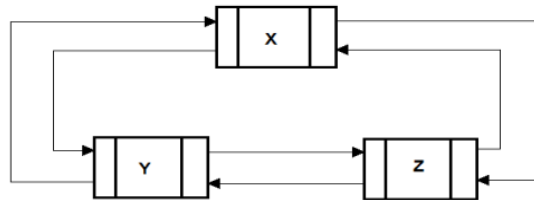


Fig. 1. Estructura de una lista doblemente enlazada

Una lista es una estructura secuencial de datos. A la vez, un elemento de la lista o nodo es básicamente un elemento de información con sus enlaces hacia los nodos anterior y posterior. Así se conforma una lista enlazada que además es dinámica, es decir que su tamaño puede cambiar durante la operación. Otra de sus ventajas fundamentales es que es flexible a la hora de reorganizar sus elementos, a cambio se a de pagar una mayor lentitud a la hora de acceder a cualquier elemento.

En la lista de la figura 1 se puede observar que hay tres elementos de información, x, y, z. Supongamos que queremos añadir un nuevo nodo, con la información p, para hacerlo basta con crear ese nodo, introducir la información p, y establecer los enlaces hacia los nodos siguiente y anterior. Resultan también simples los mecanismos de quitar un nodo o realizar la búsqueda de información dentro de la lista.

3 Nodos inteligentes

El diagrama de la figura 2, muestra una unidad funcional compuesta por: el *network capable application processor* (NCAP), el *Smart transducer interface module* (STIM) y los transductores (Sensor – Actuador) [3]. Esta unidad se denomina nodo inteligente, siendo inteligente por ser autónomo y tener la capacidad suficiente para realizar una aplicación específica independiente de la red pero, simultáneamente, forma parte de una red compuesta por otros nodos inteligentes, y posee la capacidad para interactuar con ellos y constituir de esta manera un sistema.

Si hacemos una abstracción podemos pensar que esta unidad funcional, constituye una unidad de información donde no es interpretada como un conjunto de hardware y software sino como un conjunto de datos. Por ejemplo si el nodo inteligente tiene un actuador y un sensor, siendo el actuador implementado por un relé que enciende y apaga una lámpara, y el sensor, representa un sensor de humedad, el dato que contiene el nodo inteligente visto como una unidad de información, es una palabra binaria de dos byte, donde el byte menos significativo representa el estado de la lámpara (01h la lámpara está encendida y 00h apagada) y el byte más significativo indica el porcentaje de humedad (00h indica 0% y 64h indica 100%). Se puede así abstraer aún más el nodo inteligente y que es pensado como el bloque que se muestra en la figura 3, donde se observa como vemos un nodo como un elemento de una lista.

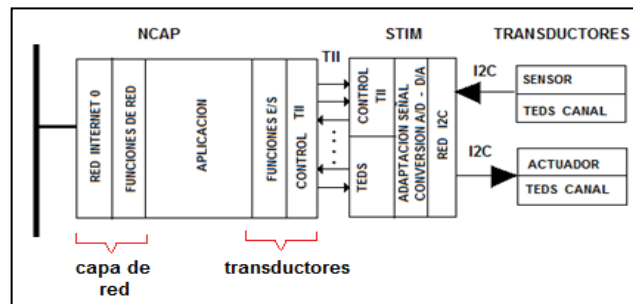


Fig. 2. Estructura de un nodo inteligente basado en el estándar IEEE 1451

En la visión del nodo, de la figura 3, se simboliza un nodo que posee la dirección 100, este nodo apunta a otro nodo con dirección 115 y es a la vez apuntado por su antecesor. Conteniendo además como información dos byte, uno que representa el estado de la lámpara y otro que contiene el valor de la humedad ambiente, como ya se mencionó anteriormente.

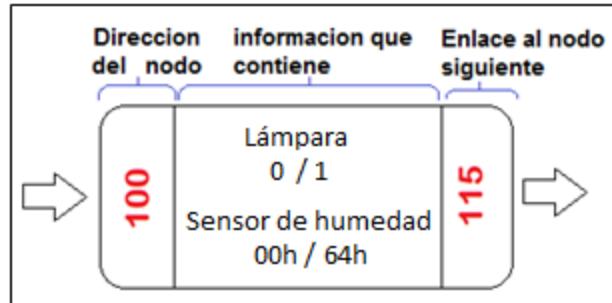


Fig. 3. Visión de un nodo inteligente como una estructura de dato

4 Capa de Red

La capa de red contiene las funciones de red que brindan soporte a la aplicación y las funciones del protocolo Internet 0 que controlan la codificación y el envío y la recepción de bits.

4.1 Driver de Internet 0

Como método de codificación utilizamos el método I0, el cual codifica un bit simple dividiéndolo en dos intervalos de tiempo. Si el impulso de reloj ocurre en el centro del primer intervalo entonces es codificado como un 1, y si ocurre en el centro del segundo intervalo es codificado como un 0. Cualquier otro impulso puede ser rechazado como ruido.

Existe cierta similitud entre la codificación I0 y las técnicas de codificación bifase. Este esquema asegura que todos los bits presentan una transición en la parte media, proporcionando así un excelente sincronismo entre el receptor y el transmisor. Una desventaja de este tipo de transmisión es que se necesita el doble del ancho de banda para la misma información que en el método convencional. Dos ventajas significativas son:

- Sincronización: debido a la transición que siempre ocurre durante el intervalo de duración correspondiente a un bit, el receptor puede sincronizarse usando dicha transición. Debido a esta característica, los códigos bifase se denominan auto-sincronizados.
- Detección de errores: se pueden detectar errores si se detecta una ausencia de la transición esperada en la mitad del intervalo. Para que el ruido produjera un error no detectado tendría que intervenir la señal antes y después de la transición.

Esta capa provee los drivers de transmisión y recepción de bits, estos realizan además la codificación para Internet 0:

Void IOCodifica(Trama, Sentido): Nos permitirá codificar el mensaje bajo el protocolo Internet 0, recibiendo como parámetro la trama armada por la capa de red.

```

Inicio
  Si Sentido = 0 → Habiolitar transmisión derecha
  Si Sentido = 1 → Habiolitar transmisiónizquierda
  Mientras no sea FindeTrama
    Leer byte
    Desde i = 0 hasta i = 7
      Si bit ==1
        Transmite ← "10"
      sino
        Transmite ← "01"
      FinSi
    DisDesde
  FinMientras
Fin

```

Char IODecodifica(mensaje): Decodifica el Internet 0.

```

Inicio
  Inicio recepción de byte
  Mientras i <= 15
    Leer bit n, bit n+1
    Si bit n ==1 & bit n+1 == 0
      Recibido ← recibido + "1"
    sino Si bit n ==0 & bit n+1 == 1
      Recibido ← recibido + "0"
    sino
      Error en recepción
    FinSi
  FinMientras
Fin

```

4.2 Funciones de red

La estructura de la figura 4 es un ejemplo de una estructura de red basada en una estructura lógica de lista, la red puede estar interconectada de cualquier forma entre los nodos inteligentes y poseer cualquier número de nodos.

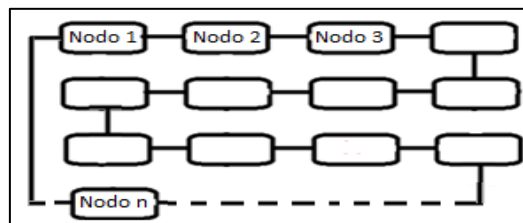


Fig. 4. Estructura de un sistema basado en nodos inteligentes

Para tener definida una estructura de lista enlazada, es necesario tener definida la estructura de información de cada nodo, los punteros a los nodos siguiente y anterior y las funciones de red tales como: transmisión, recepción, control de errores y armado de cabeceras.

En cada nodo se debe realizar una declaración del tipo:

```
struct lista_doble {
    float dato;
    lista_doble *siguiente;
    lista_doble *anterior;
};
```

Donde dato representa la información contenida en el nodo, que puede ser de cualquier tipo. En este ejemplo se trataría de datos de tipo numérico de punto flotante; *siguiente, se trata de un puntero al siguiente elemento de la lista y *anterior, es un puntero al elemento anterior de la lista. Este puntero enlaza con el elemento predecesor de la lista y permite recorrerla en sentido inverso.

Nuestra lista tendrá una serie de operaciones básicas definidas como funciones de biblioteca de la capa de red, que nos permiten el manejo del sistema, siendo alguna de ellas, el armado de la trama, validación de mensajes recibidos, transmisión y recepción.

Int IdentificarDestino(Mensaje): Nos permitirá determinar la dirección del nodo destino a partir del conocimiento del canal transductor donde se dirige el mensaje. Devuelve el número del nodo destino

Inicio

```
IDTrCh ← ExtraerNrocanal_transductor(mensaje)
IDNodoDestino ← Nodos(IDTrCh)
```

Fin

Int CalculaLargo(mensaje): Calcula el largo de la trama constituida por la dirección del nodo destino, el nodo emisor, el mensaje.

Inicio

```
Desde i=0 hasta fin_de_mensaje
    Largo_mje = largo_mje + 1
Fin-desde
```

Fin

Cadena ArmarTrama(IDNodoDestino, IDNodoOrigen, Largo, Mensaje): Esta función recibe como parámetros los elementos que constituyen la trama, agrega el número de secuencia y el checksum y devuelve la trama a transmitir.

Inicio

```
String Trama[]
Int i, j
NroSecuencia ← Calcular_Nro_Secuencia
Desde i = 0 hasta 7
    Trama[i] = IDNodoDestino[i]
    Trama[i+8] = IDNodoOrigen[i]
    Trama[i+16] = Largo[i]
```

```

        Trama[i+24] = NroSecuencia[i]
    Fin-desde
    Desde j = 0 hasta largo
        Trama[j+32] =Mensaje[j]
    Fin-desde
    CkSum ← Calculo-Check-Sum(Trama)
    Trama = Trama & CkSum & finTrama
Fin

```

Void EnviarTrama(Trama): Colocar sobre la red física la cadena de bits del mensaje, no devuelve ningún valor. Como se puede observar requiere de una función para determinar si el mensaje será enviado hacia la izquierda o derecha. El algoritmo utilizado es el más simple y se tomara la dirección izquierda si IDNodoDestino < IDNodoOrigen y dirección derecha en caso contrario.

```

Inicio
    Sentido ← SentidoDeTransmision(IDNodoDestino,
        IDNodoOrigen)
    IOCodifica(Trama, Sentido)
Fin

```

Cadena RecibirTrama(): Lee el canal de ingreso y arma la trama recibida, Devuelve una cadena de caracteres, guardándola en un buffer intermedio de recepción.

```

Inicio
    Carácter ← IODecodifica()
    Mientras Carácter ≠ fin_de_trama
        Trama = trama + Carácter
        Carácter ← IODecodifica()
    Fin-mientras
Fin

```

Boolean ValidaDireccion(Trama): Identifica si el mensaje recibido es para este nodo o debe ser retransmitido. Devuelve un booleano identificando como verdadero un mensaje valido para este nodo y como falso un mensaje a ser retransmitido. Verifica además si el mensaje recibido fue enviado por quien recibe, en este caso destruye el mensaje

```

Inicio
    IDNodoDestino ← Extraer número nodo Destino
    IDNodoOrigen ← Extraer número de nodo Origen
    NroSecuencia ← Extraer número de secuencia
    Si IDNodoDestino == ID de este nodo
        MjeValido ← verdadero
    Sino
        MjeValido ← Falso
    Fin-si
    Si IDNodoOrigen == ID de este nodo
        DestruirMensaje(NroSecuencia)
    Fin-si
Fin

```

A estas funciones las complementan otras funciones adicionales tales como: cálculo del checksum, Verificación del checksum, Leer buffer de salida, escribir buffer de entrada, entre otras.

5 Implementación de la red de nodos

La red propuesta en la figura 5, es un ejemplo, implementado con ocho nodos, se emplean multiplexores para transmitir y demultiplexores para recibir, con esto se consigue direccionar los mensajes hacia adelante y hacia atrás de un nodo específico; por simplicidad no se dibujan las líneas de selección de los multiplexores y de los demultiplexores.

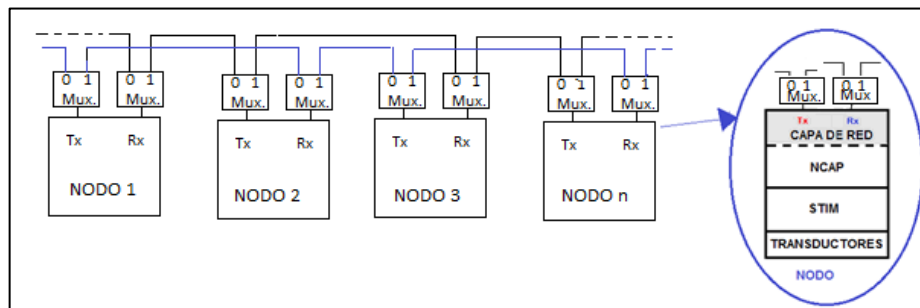


Fig. 5. Red de nodos

La comunicación entre nodos es por dos hilos y se detalla con precisión en el artículo *Internet 0 en Entornos Residenciales* presentado en el Congreso Argentino de Ingeniería – 2012 [2] [5]

Mediante la implementación de un algoritmo, los nodos pueden transmitir hacia adelante y hacia atrás independientemente de la posición en que se encuentren. La precisión en este algoritmo es fundamental a la hora de analizar la velocidad y la carga de la red. De la misma manera, la información recibida puede provenir del nodo anterior o posterior, la identificación de la ruta activa se realiza por hardware. Ante la llegada de dos mensajes simultáneos, el nodo en función de sus prioridades abortará una de las comunicaciones y continuará con la otra.

6 Transmisión de datos

La utilización del protocolo Internet 0 (I0) implica la reducción del número de capas, respecto de Internet común. Esto redundará en una reducción del código, lo cual posibilita disminuir los tiempos de procesamiento computacional. Este protocolo I0 elimina las capas de transporte, Sesión y Presentación.

La capa de aplicación genera el mensaje del usuario constituyendo así la unidad de datos de protocolo (PDU – a los efectos de mantener la nomenclatura) y le añade una cabecera, que contiene la dirección de canal transductor destino, constituyendo

así la unidad de datos de interfaz (IDU). La IDU se transfiere a la capa de red y a través del enlace físico, a la capa de red del nodo destino, y se procesa el mensaje. Para ello ha sido necesario todo este proceso:

1. La PDU se prepara en la capa de aplicación del emisor.
2. Antes de entregar la PDU a la capa de red se le añade la correspondiente cabecera con la dirección del canal transductor que será el destinatario y se transformaría así en una Unidad de datos de interfaz (IDU).
3. La capa de red recibe la IDU, le añade la información de control consistente en: Dirección del nodo destino donde se encuentra el transductor destino, Su propia dirección de nodo emisor, largo total del mensaje y checksum.
4. Al llegar al nivel físico se envían los datos que son recibidos por la capa de red del receptor.
5. En el receptor la capa de red controla la integridad del mensaje (checksum). Identifica el nodo destino y acepta o retransmite el mensaje.
6. El receptor que acepta el mensaje, extrae la cabecera y traslada el mensaje a la capa de aplicación, la cual lo procesará.

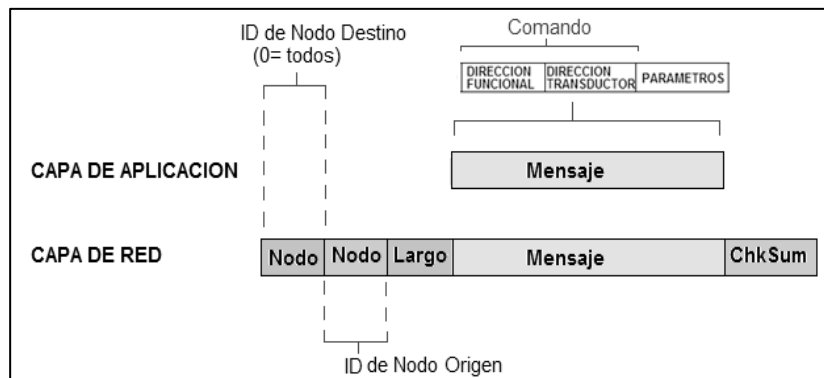


Fig. 6: Estructura de datos del mensaje y de la trama

La transferencia de información entre la capa de aplicación y la capa de red, del NCAP, se realiza mediante dos buffers intermedios, uno de mensajes salientes y otro de mensajes entrantes. Estos buffers poseen una capacidad inicial de 80 bytes y pueden ser modificados en forma dinámica. El almacenamiento dentro de los buffers es de tipo big endian.

7 CONSIDERACIONES FINALES

Basados en las consideraciones precedentes y en modelo simulado, podemos enunciar nuestra propuesta con la estructura considerada apta para el armado de una red de nodos inteligentes, partiendo de las siguientes premisas: no se requiere alta veloci-

dad para la transferencia de información, la cantidad de mensajes es baja y en general se circunscriben a entornos de baja cantidad de nodos.

- No se requiere de un nodo maestro.
- Más de un mensaje puede circular simultáneamente. Por ejemplo en el esquema de la figura 4, los nodos 1, 2, 3 y 4 (de la habitación A) puede estar intercambiando mensajes entre si y simultáneamente los nodos 7 y 8 (de la habitación B) pueden establecer otra comunicación.
- Se reduce el problema de colisiones.
- Se pueden seguir varios caminos para obtener la información, si un enlace se rompe se puede seguir otro enlace.
- No es una estructura estática, se pueden agregar/quitar tantos nodos como se desea, modificando los enlaces necesarios
- Los algoritmos de búsqueda de información dentro de una lista están ampliamente estudiados por las ciencias informáticas.
- Cada nodo ejecuta su aplicación y si necesita información de otros nodos puede hacer uso de su lista de nodos para su aplicación, pero si la información que necesita no la posee su entorno (sub-red) puede ejecutar un algoritmo de búsqueda fuera de su sub-red, leyendo las TEDS de otros nodos y ampliar de este modo su propio entorno.
- Los NCAP no tienen que atender todos los mensajes que circulan por la red y la decisión de atender o ignorar un mensaje se realiza en la capa de red.

Consideramos que la estructura propuesta resulta de interés para redes de uso doméstico e industrial, y nos apoyamos en las ventajas presentadas y en la simplicidad del cableado,.

REFERENCIAS

- [1] S.H. Gallina; P. Beltramini; D. Villagran; M. Ferraro; L. Arjona; D. Lobos. “Diseño De Un Nodo Con Capacidad Plug & Play”; IV Congreso de microelectrónica aplicada. Universidad Tecnológica Nacional- Facultad Regional Bahía Blanca. Año 2013.
- [2] S.H. Gallina; P. Beltramini; D. Villagran; G. Peretti; S.F. Felissia. “Internet Cero: Domótica Con Nodos Inteligentes” Producción Científica de la F.T.y C.A. III. Paginas 159 a 164. Año 2012
- [3] IEEE 1451.0. Draft Standard for a Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats. January 2007
- [4] V. Cerveron “Estructura De Datos” Universidad de Valencia. Instituto Universitario de Investigación de robótica y tecnologías de la Información y las Comunicaciones (IRTIC). <http://www.uv.es/uvweb/institut-universitari-investigacio-robotica-tecnologies-informacio-comunicacio-IRTIC/>
- [5] J.A. Pucheta; S.H. Gallina; D. Villagrán; P. Beltramini; G. Peretti; S.F. Felissia. “Internet 0 En Entornos Residenciales” Congreso Argentino de Ingeniería – 2012