Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERIA ELECTRÓNICA DE COMUNICACIONES

Trabajo Fin de Grado

Implementación de técnicas de control distribuido, protocolo OSPF en redes inteligentes de energía.

Autor: José Manuel Muñoz Calles

Tutor: Fco. Javier Rodríguez Sánchez

Año 2013

Universidad de Alcalá

GRADO EN INGENIERIA ELECTRÓNICA DE COMUNICACIONES

Trabajo Fin de Grado

Implementación de técnicas de control distribuido, protocolo **OSPF** en redes inteligentes de energía.

Autor: José Manuel Muñoz Calles

Tutor: Fco. Javier Rodríguez Sánchez

TRIBUNAL:

Presidente: Emilio José Bueno Peña

Vocal 1º: Pedro Martín Sánchez

Vocal 2º: Fco. Javier Rodríguez Sánchez

CALIFICACIÓN: .....

FECHA: .....

Como no puede ser de otra manera, dedico este Trabajo Fin de Grado a todos los que han hecho para que yo pudiera dejar de hacer y dedicara ese tiempo a conseguir el título.

Y especialmente a Natalia, Asia, Maxim, Nati, Jose y Patricia.

Vale.

Índice General

Índice de ilustraciones

Índice de tablas

1 RESUMEN

El presente **Trabajo Fin de Grado** titulado ***Implementación de técnicas de control distribuido, protocolo OSPF en redes inteligentes de energía***, profundiza en la **reconfiguración de los enlaces de las subestaciones secundarias** en una **red de distribución eléctrica inteligente**, mediante la implementación de un protocolo robusto, sencillo y fiable, como el **protocolo de enrutamiento OSPF**.

De esta forma, los **Agentes virtuales** podrán **modificar dinámicamente el mapa de conexiones sin la intervención de los operadores del sistema** en caso de fallo o falta, por necesidades del servicio, para disminuir pérdidas o para mejorar la fiabilidad de la red.

1.1 Palabras Clave

Smart Grids, Reconfiguración de redes de distribución eléctrica, Subestaciones secundarias, Centros de Transformación, Protocolo **OSPF**, Proyecto PRICE.

2 ABSTRACT

This **Thesis** called ***Control distributed technics implementation, OSPF protocol in Smart Grids***, develops a idea to **reconfigure the connections between secondary substations** in **intelligent distribution network**, introducing the robust, simple and reliable **routing protocol OSPF**.

Thereby, **virtual Agents should be able to modify dynamically the grid without system operator interference**, in case of fault or fail, for service requirements, to minimize losses or improve network reliability.

2.1 Keywords

Smart Grids, Power Distribution Grid Reconfiguration, Secondary substations, Transformers, **OSPF** protocol, PRICE project.

3 ACRÓNIMOS

ACRÓNIMO	*DESCRIPTION (EN)*	DESCRIPCIÓN (ES)
ACO	*Ant Colony Optimization*	Optimización de Colonia de Hormigas
AMI	*Advanced Metering Infrastructure*	Infraestructura de Medida Avanzada
AMR	*Automated Meter Reading*	Lector de Medidas Automatizado
AS	*Autonomous Systems*	Sistema Autónomo
ATM	*Asynchronous Transfer Mode*	Modo de Transferencia Asíncrono
BDI	*Belief–Desire–Intention Software Model*	Modelo de Software Creencia – Deseo - Intención
BI	*Business Intelligence*	Inteligencia para los Negocios
CB	*Circuit Breaker*	Corto circuito, disruptor, disyuntor
CIM	*Common Information Model*	Modelo de Información Común
CP	*Connection Point*	Punto de conexión
DDS	*Data Distribution Service*	Servicio de Distribución de Datos
DER	*Distributed Energy Resources*	Recursos de Energía Distribuida
DFR	*Distribution Feeder Reconfiguration*	Reconfiguración de la Alimentación Distribuida
DMS	*Distribution Management System*	Sistema de Gestión Distribuida
DNO	*Distribution Network Operators*	Operadores de la Red de Distribución
DVR	*Dynamic Voltage Restorer*	Restaurados de Tensión Dinámico
FDDI	*Fibber Distributed Data Interface*	Interfaz de Datos Distribuida por Fibra
GC	*Grid computing*	Computación en Red
GOOSE	*Generic Object Oriented Substation Event*	Evento de Subestación Orientado a Objeto Genérico
GSSE	*Generic Substation Status Event*	Evento de Estado de Subestación Genérico
HMI	*Human Machine Interface*	Interfaz Humano Máquina
ICT	*Information and Communications Technologies*	Tecnologías de la Información y las Comunicaciones
IED	*Intelligent electronic device*	Dispositivo Electrónico Inteligente
IGP	*Interior Gateway Protocol*	Protocolo de Pasarelas de Interior
IP	*Internet Protocol*	Protocolo de Internet
MIB	*Management Information Base*	Base de Gestión de la Información,
MMS	*Manufacturing Message Specification*	Especificación de Mensajes de Fabricación
NAN	*Neighbourhood Area Networks*	Redes de Área Vecinales
NERC	*North American Electric Reliability Corporation*	Corporación para la Fiabilidad Eléctrica en Norte América
Ontología[1].	*Ontology*	El término ontología en informática hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades.
OSPF	*Open Shortest Path First*	Primer Camino más Corto Abierto
OWL-DL	*Ontology Web Language - Description Logic*	Lenguaje Ontológico de la Red – Lógica Descriptiva

[1] http://es.wikipedia.org/wiki/Ontolog%C3%ADa_(Inform%C3%A1tica)

PSO	*Particle Swarm Optimization*	Optimización por Enjambre de Partículas
RDF	*Resource Description Framework*	Marco Descriptivo de Recursos
RMU	*Ring Main Unit*	Unidad Principal del Anillo
SA	*Simulated Annealing*	Recocido Simulado
SCADA	*Supervisory Control And Data Acquisition*	Supervisión, Control y Adquisición de Datos
SCL	*Substation Configuration Language*	Lenguaje de Configuración de Subestaciones
SMDS	*Switched Multi-megabit Data Service*	Servicio de Datos Multi-megabit Conmutado
SNMP	*Simple Network Management Protocol*	Protocolo de Gestión de Redes Simple
SPARQL	*SPARQL Protocol and RDF Query Language*	Lenguaje de Consulta RDF y Protocolo SPARQL
STATCOM	*Static Synchronous Compensator*	Compensador (condensador) Síncrono Estático
SWT	*Semantic Web Technologies*	Tecnologías Semánticas de la Red
TCP	*Transmission Control Protocol*	Protocolo de Control de Transmisión
UDP	*User Datagram Protocol*	Protocolo de Datagramas de Usuario
UML	*Unified Modelling Language*	Lenguaje de Modelado Unificado
XML	*eXtensible Markup Language*	Lenguaje de Marcas Extensible

4 RESUMEN EXTENDIDO

Las redes eléctricas de transporte y de distribución, ver *Figura 1*, gracias a la integración de las *ICT*[2], están sufriendo una profunda transformación en su arquitectura y en sus sistemas, en la búsqueda de un uso más eficiente y racional de sus recursos, lo que permitirá un ahorro de costes para las empresas, un mejor servicio al consumidor y un menor impacto en el medio ambiente. Este fenómeno se está produciendo no solo a nivel español o europeo, sino que es una tendencia global.

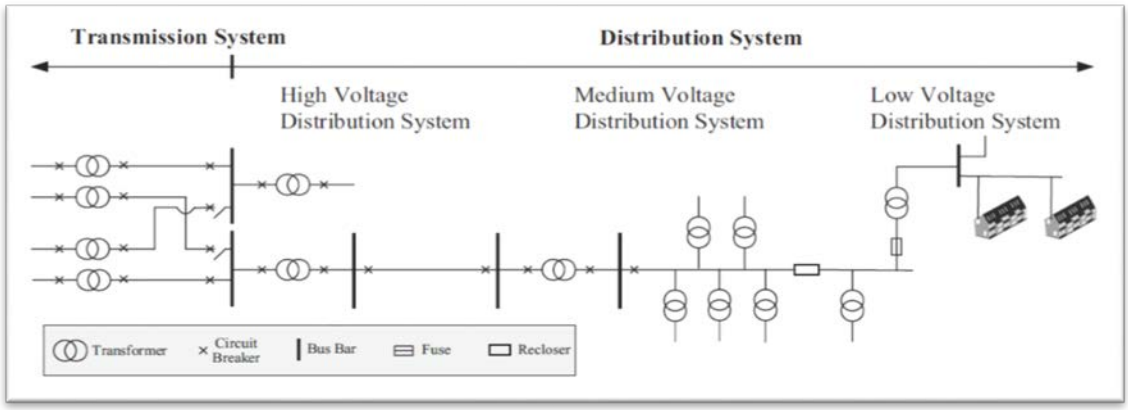


Figura 1 - Ejemplo red de distribución.

La evolución de la red eléctrica basada en una estructura en árbol unidireccional y pasiva, hasta una red inteligente ***(Smart Grid)*** en todos sus tramos, sin duda, debe cambiar nuestra visión pasando de un enfoque vertical y estructurado, a uno horizontal y distribuido, en el que las decisiones se tomen de forma descentralizada. Los proyectos ENERGOS y PRICE son dos referentes de la implementación de dichas tecnologías.

En las redes de distribución en las ***Smart Grid*** *(SG)*, el Centro de Transformación *(secondary substation)* se ha convertido en un elemento inteligente clave que gestionado por un Agente virtual, mediante un uso intensivo de comunicaciones, permite gestionar la red de forma distribuida y automática.

De esta forma, los Agentes, entre otras muchas operaciones, podrán modificar dinámicamente el mapa de conexiones sin la intervención de los operadores del sistema en caso de fallo o falta, por necesidades del servicio, para disminuir pérdidas o mejorar la

[2] Disculpas anticipadas por introducir una gran cantidad de acrónimos y anglicismos, inevitable.

fiabilidad de la red.

En esa dinámica, la reconfiguración de las redes de distribución, abre un campo al que aportamos esta idea consistente en implementar una **técnica de control distribuido basada en el protocolo OSPF,** cambiando la topología de sus enlaces y conexiones para adaptarlas a las necesidades del servicio o de la operativa propia del sistema. En base al protocolo de enrutamiento **OSPF** desarrollaremos la función de reconfiguración de los enlaces de las subestaciones secundarias en una red de distribución eléctrica inteligente mediante la implementación de un protocolo robusto, sencillo y fiable, que desde los años 90 se utiliza en el *backbone* de Internet.

Con este Trabajo Fin de Grado hacemos una pequeña contribución al proyecto del Departamento de Electrónica de la Universidad de Alcalá, en el marco del proyecto PRICE, en colaboración con el grupo *GEISER (Grupo de Electrónica e Ingeniería aplicada a Sistemas de Energía Renovables - Electronic Engineering applied to Renewable Energy System Group).*

5 INTRODUCCIÓN

En este trabajo desarrollado en el ámbito de las ***Smart Grids***, se pretende justificar la implementación del protocolo de enrutamiento **OSPF**, en el entorno de las redes de distribución controladas por Agentes, para reencaminar el transporte de energía entre unos nodos y otros cuando se produce una falta o un fallo en algún elemento de la red, o se quieren minimizar las pérdidas energéticas.

Haremos un recorrido por las tecnologías y desarrollos necesarios para plantear el proyecto, en los métodos de intercambio de información basados en CIM y por último profundizaremos en el conocimiento del protocolo de routing **OSPF**.

A continuación, explicaremos la configuración específica necesaria del **OSPF** en el entorno de una red de distribución eléctrica, esbozaremos la necesidad de desarrollar un interfaz específico denominado ***iOSPF-SG*** que permita cubrir las necesidades adicionales de este protocolo en nuestro entorno de trabajo. Efectuaremos los cálculos necesarios con la ayuda de MATLAB que nos permitan conocer el camino óptimo entre dos puntos de la red en el caso de producirse un cambio de estado en uno de los nodos o de los enlaces y obtendremos la tabla de encaminamiento resultante para cada uno de los nodos.

Este trabajo se subdivide en dos grandes bloques, en el primero fundamentamos la ***Base Teórica*** que nos permitirá abordar con garantías la segunda parte de ***Desarrollo de la Investigación***. Además de estos apartados, incluimos en nuestro trabajo otros capítulos introductorios y complementarios.

El primer gran bloque ***Base Teórica*** contiene cuatro capítulos, la red de distribución en las ***Smart Grids*** descrita con detalle, haciendo hincapié en los dispositivos, sistemas de gestión y tratamiento de faltas (ver *6.1 La Red de Distribución en la Smart Grid)*; las normas para el intercambio de información (ver *6.2 Normalización para el intercambio de información.);* una breve introducción al proyecto PRICE (ver *6.3 Proyecto PRICE);* y un análisis en profundidad del protocolo de routing **OSPF** (Ver *6.4 Protocolo **OSPF**)*.

En cuanto al bloque ***Desarrollo de la Investigación***, hacemos una modesta revisión del estado de las investigaciones en determinados aspectos que desarrollaremos más adelante (ver *7.1 Líneas de investigación - State of the Art*); una descripción del diseño que tomamos como base para efectuar nuestro desarrollo (ver *7.2 Descripción del diseño*); los cálculos

efectuados (ver *7.3 Cálculos efectuados*) y por último las conclusiones del estudio y recomendaciones para trabajos futuros (ver *7.4 Conclusiones*).

De esta forma, habremos justificado la validez del protocolo **OSPF** y de su algoritmo *Dijkstra*, para reorganizar la red de forma dinámica y autónoma calculando los caminos más adecuados en el segmento definido.

6 BASE TEÓRICA

Las ***Smart Grid*** utilizan de forma avanzada información y comunicaciones para controlar un sistema de energía eléctrica eficaz y fiable. Puede verseer un ejemplo de esta transformación en la *Figura 2*.

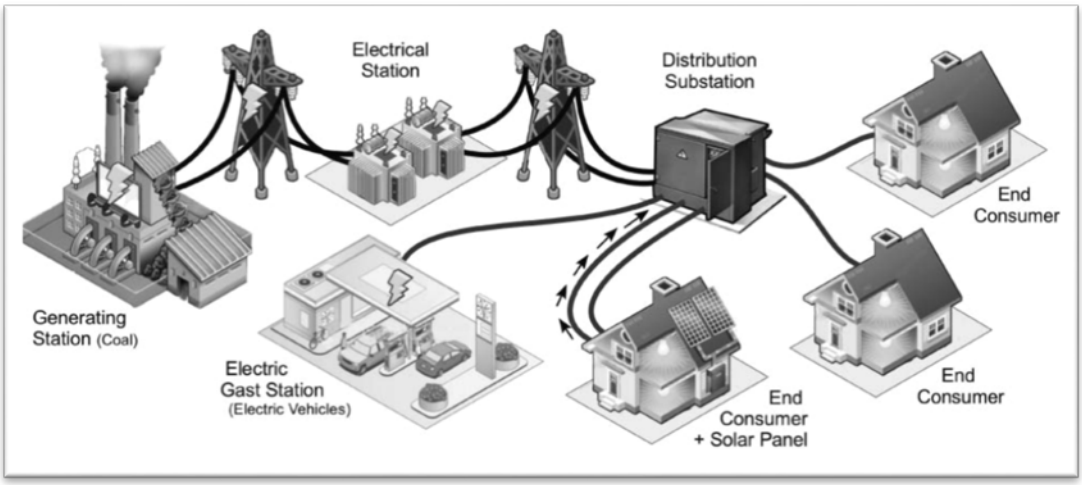


Figura 2- Esquema simplificado de una Smart Grid

En la ***Smart Grid***, más que en otras aplicaciones comerciales de intercambio de datos, es necesario que la información se obtenga en tiempo real para la monitorización y control de la red [1]. De esta forma, en caso de fallos u otras contingencias en la red, el número de datos que se pierdan debe ser el menor posible, entre otras cuestiones, por ello el tamaño de los mensajes intercambiados debe ser pequeño.

6.1 La Red de Distribución en la *Smart Grid*

Las redes de distribución eléctricas conectan el sistema de transmisión en alto voltaje con los usuarios finales. Es decir, el sistema tradicional implantado durante los últimos 70 años es unidireccional, desde la generación hasta el cliente final a través de la red de distribución. El reto de las ***Smart Grids*** [2] en la red de distribución, las cuales se caracterizan por su complejidad, es plantear un cambio radical y de planteamiento que les permita afrontar, los siguientes retos:

- Las redes de distribución son construidas habitualmente como redes malladas pero operadas radialmente, no obstante, su topología cambia frecuentemente durante las operaciones debido a las faltas y las tareas de mantenimiento.

- La estructura de la red cambia y se expande por la llegada de nuevos consumidores.
- Normalmente las trifásicas están desequilibradas.
- Cumplimiento estricto de los objetivos de calidad y rendimiento.
- La comunicación entre los diferentes elementos de la red es limitada y en muchos casos local. Una monitorización que nos permita comprender en detalle el funcionamiento de la red de distribución requerirá una gran cantidad de datos.
- Cargas en la red.
    - La composición de las cargas es compleja y no está suficientemente conocida.
    - El patrón de consumos de carga en distribución varía dinámicamente con el tiempo, las tendencias de las variaciones de carga son más complejas de predecir que las de la redes de transmisión.
    - No es posible obtener mediciones simultáneas de todas las cargas, las mediciones de cargas son normalmente insuficientes y pueden contener errores graves y datos malos.
    - La correlación entre cargas no está suficientemente estudiada.

6.1.1 Dispositivos automáticos

6.1.1.1 Equipamiento de una subestación

Los componentes típicos de un sistema de automatización de una subestación [3] se muestran en la *Figura 3*.

Todos los componentes tradicionales de una subestación, disyuntores, seccionadores, transformadores de corriente y tensión, transformadores de potencia, están cableados y conectados a sistemas de control y gestión, ya sea través de sistemas analógicos o digitales. Los sistemas de control indicados son:

- *HMI, el Interfaz Humano-Maquina* que muestra el diseño de la estación y el estado de los equipos de estaciones.
- El ordenador que gestiona localmente los datos de la subestación mediante una aplicación.

- Pasarela al sistema SCADA.

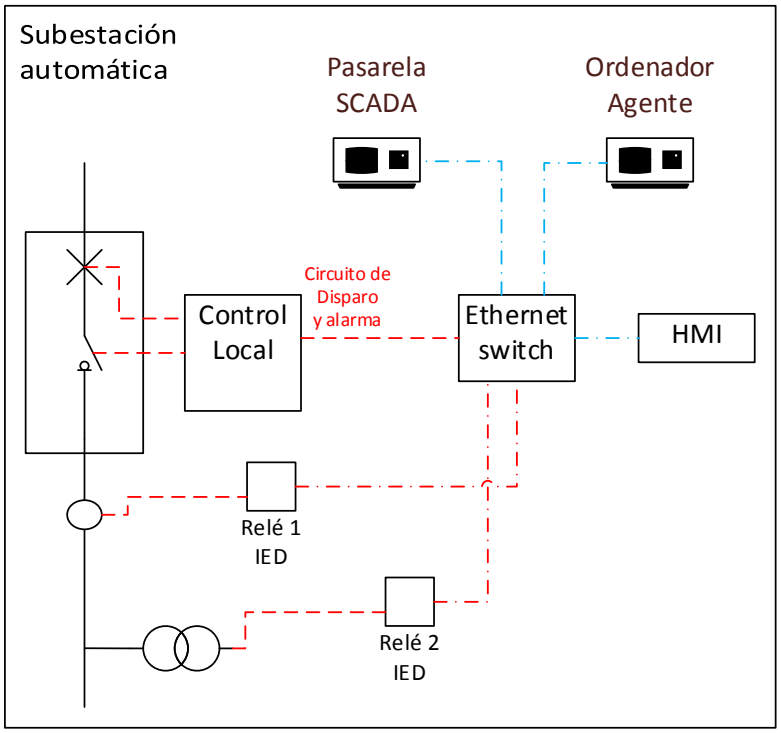


Figura 3 - Esquema de subestación automática.

Dotar de inteligencia a una subestación tradicional, requiere de una gran cantidad de dispositivos y muchos kilómetros de cableado. Las comunicaciones entre dispositivos son mediante Ethernet y utilizan principalmente protocolos basados en IEC 60870-101/104.

6.1.1.2 Dispositivos Electrónicos Inteligentes - *Intelligent electronic devices (IED)*

Los IEDs describen un amplio rango de aparatos que permiten una o varias funciones de protección, medida, grabación de faltas y control. Un *IED* consiste en una unidad de procesado de señal, un microprocesador con entradas y salidas y un interfaz de comunicaciones EIA 232/EIA 483, Ethernet, *Modbus* o DNP3. Entre otros, tipos tenemos:

6.1.1.2.1 Relés IED.

Combinan sus funciones propias de protección con otras de medida, grabación y monitorización de la línea de transmisión, como muestra la *Figura 4*. Algunos de los modelos[3] existentes son:

- Sobre intensidad de corriente instantánea de trifásica: Tipo 50.

[3] Designación IEEE/ANSI.

- Sobre intensidad de corriente retardada de trifásica: Tipo 51.
- Sobre intensidad de corriente retardada o instantánea de trifásica controlado por tensión o de tensión moderada: Tipos 50V y 51V.
- Sobre intensidad de corriente retardada o instantánea por falta de tierra: Tipos 50N y 51N.

Las medidas tomadas localmente primero son evaluadas y después puestas a disposición de todos los procesadores en el *IED* de protección. Un usuario puede leer estas medidas digitalizadas mediante una pequeña pantalla del equipo, y además un teclado está disponible para introducir comandos o configuraciones.

Varios algoritmos para diferentes funciones de protección son almacenados en una memoria ROM, por ejemplo el Tipo 50 continuamente chequea las medidas locales de corriente frente a un valor fijado (que puede ser introducido por el usuario localmente o remotamente), para determinar si existe una sobre intensidad de corriente en el alimentador al que el disruptor está conectado, si la corriente es mayor que el valor fijado, un mensaje de alerta es generado y comunicado al circuito.

Este tipo de *IEDs* tienen un contacto tipo relé que está cableado en serie con la bobina de disparo del disyuntor y la orden de disparo completa el circuito, abriendo así el *CB*.

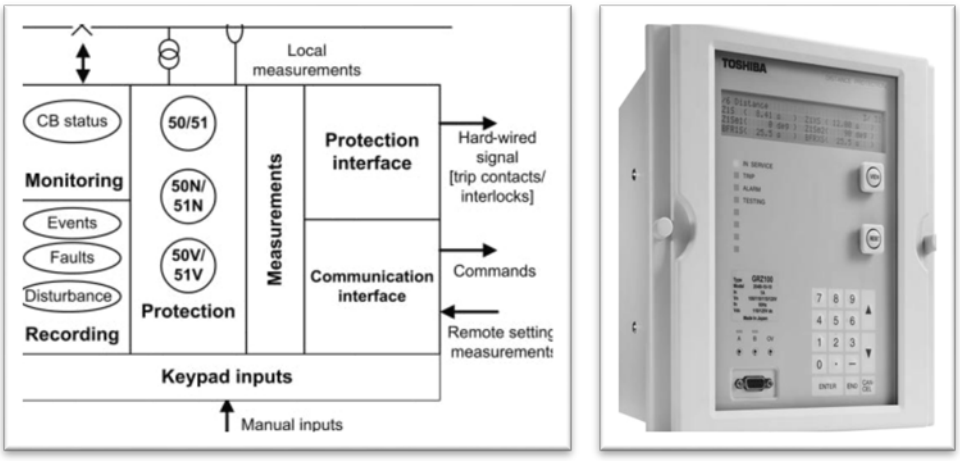


Figura 4 - Configuración típica relé IED e imagen.

6.1.1.2.2 Medidor IED.

Proveen un amplio rango de funciones y características para la medición de los parámetros de una línea con trifásica o monofásica. Típicamente miden tensión, corriente, potencia,

factor de potencia, energía consumida en un periodo, demanda máxima, valores máximos y mínimos, distorsión harmónica total y componentes harmónicos.

6.1.1.2.3 Grabador IED

Aunque muchos de los *IEDs* de protección y medida mencionados anteriormente disponen de capacidad de almacenamiento de datos, estos equipos son utilizados de forma adicional para monitorizar y grabar los cambios en la subestación y en los alimentadores de salida. Una grabación continua de eventos con una resolución mayor de 1 ms está disponible en algunos *IEDs*. Estas grabaciones pueden ser consultadas por expertos para hacer comprobaciones sobre eventos pasados, lo cual es de gran interés en situaciones de falta, las grabaciones nos permiten identificar el comportamiento del equipamiento primario y secundario de la red antes y durante la interrupción.

6.1.1.2.4 Controlador - *Bay controller*

Este equipamiento es empleado para controlar y monitorizar la aparamenta, el transformador y otro equipamiento interno, facilitando acciones de control remoto, ya sea desde el centro de control o de un punto de control en la propia subestación. Las funcionalidades de este dispositivo pueden variar pero típicamente incluyen:

- Control de disyuntores.
- Comprobación del bloqueo de la aparamenta.
- Control de cambio del transformador.
- Control de secuencias automáticas programables.

6.1.1.3 Unidades terminales remotas - *Remote terminal units (RTUs)*

Es una denominación genérica de dispositivos de adquisición de datos (medidas y estados) utilizados en un sistema *SCADA*. Estos datos tomados por *RTUs* situadas en diferentes partes de la red de distribución, denominadas *RTU* de campo, envían sus medidas a la *RTU* de la estación, situada en el interior de la subestación.

Los *RTUs* de campo actúan como interfaz entre los sensores y la *RTU* de la estación, sus funciones principales son: monitorizar señales digitales y analógicas de los sensores (medidas) y de las señales de los actuadores (estados) y convertir las señales analógicas de los sensores y actuadores en formato digital. La *RTU* de la estación obtiene los datos mediante sondeo en intervalos predefinidos, sin embargo, cuando un estado cambia es

comunicado inmediatamente sin esperar al intervalo.

La última generación de *RTUs* son capaces de ejecutar funciones de control además de procesado y comunicación. Los programas embebidos permiten incluir nuevas funcionalidades como modificar los parámetros de seguimiento, los tiempos de toma de muestras, ejecutar órdenes condicionadas, enviar acciones de control a los circuitos finales y configurar y comunicar diferentes tipos de alarmas.

6.1.2 Sistemas de Gestión Distribuida *(DMS)*

Actualmente un *DMS* necesariamente incluye la monitorización, control, análisis y gestión en tiempo real, mediante un sistema centralizado dirigido a la intervención de los operadores del sistema. Un *DMS* es una colección de aplicaciones que permiten monitorizar, controlar y optimizar el rendimiento del sistema de distribución y es capaz de gestionar su complejidad.

Actualmente con la intervención de los *DNOs* pero en un futuro no muy lejano, apoyándose en agentes virtuales de control. Los objetivos a conseguir deberían ser los siguientes:

- Un sistema inteligente que fuera capaz de auto mantenerse.
- Mejoras en la fiabilidad y calidad del suministro.
- Mayor eficiencia y eficacia de la operación del sistema.
- Mejor gestión de los activos.
- Provisión de nuevos servicios.
- Mejora en la satisfacción de los usuarios – clientes.

Actualmente los parámetros de los modelos para los sistemas de distribución son obtenidos de los datos e información de los fabricantes de dispositivos y de los operadores, o de pruebas de campo, con el cambio de las condiciones externas, por ejemplo la temperatura ambiente o el envejecimiento de los equipos, dichos parámetros pueden cambiar a lo largo del tiempo e introducir errores en el modelado de la red, llegando incluso a hacer imposible la operación del sistema.

La aplicación de las *ICTs* permite conseguir la modelización de un sistema de forma más precisa mediante la aplicación de técnicas estadísticas de caracterización de los sistemas. La

identificación de un sistema, ampliamente utilizada en ingeniería de control, puede ser usada para construir modelos matemáticos de un sistema de distribución basados en una gran cantidad de datos medidos por el sistema *ICT*. Un conjunto de modelos actualizados continuamente y más precisos pueden ser obtenidos y usados en aplicaciones *DMS*.

6.1.2.1 Topología y Análisis

Una red de distribución de energía eléctrica consiste en una variedad de dispositivos que deben ser configurados de una forma concisa y determinada para el análisis de los sistemas de energía. La cartografía entre un modelo de planta física y uno de análisis de sistemas de energía debe ser llevado a cabo mediante herramientas de análisis topológicas a través de una reducción de la red, de esta forma se disminuye el tamaño de datos alimentados en otras herramientas de análisis y modelado para que resulten más sencillas de interpretar por el operador. Un ejemplo de lo mencionado anteriormente se presenta en la *Figura 5*:

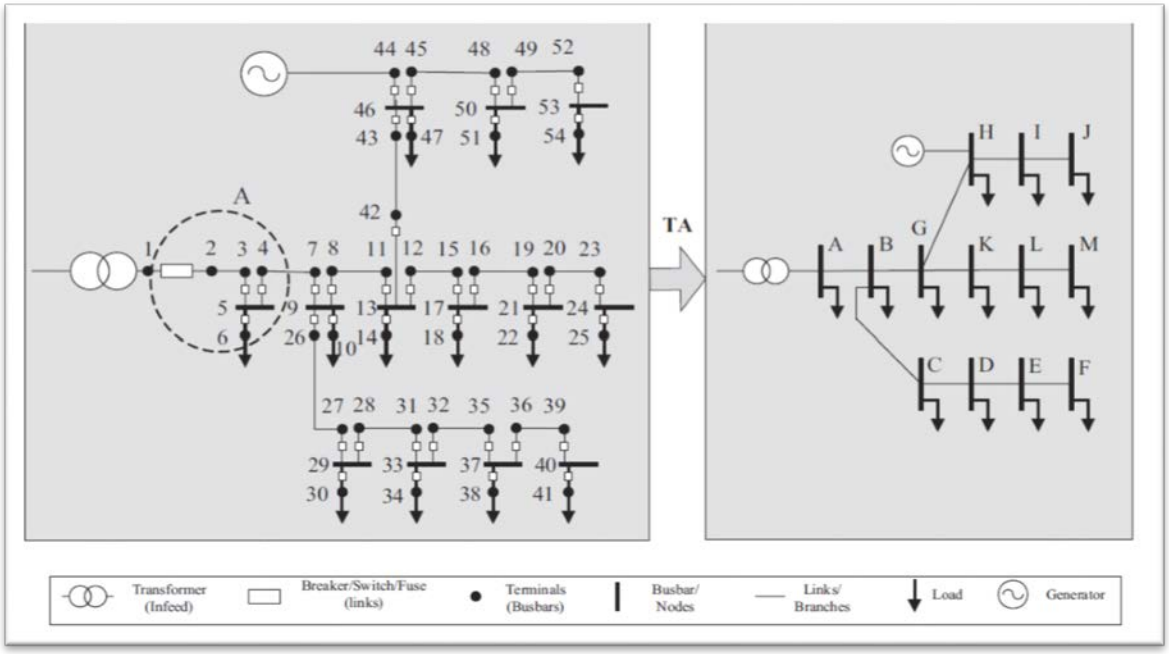


Figura 5 - Ejemplo de análisis topológico.

El análisis topológico facilita la implementación de aplicaciones *DMS* en tiempo real. Los conceptos más habitualmente utilizados son los siguientes:

- Barra (*Busbar*). Es simplemente un conector con una impedancia cero (o muy baja) para varios circuitos.
- Alimentación (*Infeeds*). Representa la energía entrante de una red no modelada "aguas arriba" (*upstream*). La alimentación está conectada a una barra en el modelo

de la planta física, una “isla energética” aparece si el estado de la alimentación es “ON” (encendido).

- Generadores. Un generador se conecta a una barra en el modelo físico de la planta, y una “isla energética” se obtiene si el estado de la alimentación es “ON” (encendido).
- Enlaces. Un enlace es una conexión con impedancia cero entre dos barras, por ejemplo un disruptor, un cortocircuito, un seccionador, o un cable corto. Cada enlace tiene dos estados ON y OFF, en el caso de estado ON, aparecerán representados dos barras a cada lado del nodo.
- Cargas. Una carga se conecta a una barra en el modelo de planta física, una carga es un elemento que consume energía.
- Nodo. Un nodo son salidas en el análisis topológico y representan un conjunto de barras conectadas entre sí mediante enlaces.
- Islas energizadas. Son salidas en el análisis topológico y representan un conjunto de nodos alimentados conectados por ramas activadas, en funcionamiento. Las islas separadas eléctricamente pueden ser identificadas y permiten un análisis de flujo de energía separado para cada una de ellas con un nodo estacionario.
- Ramas. Una rama tiene una impedancia distinta de cero, por ejemplo una línea aérea, un cable o un transformador. Una rama está “viva” si su estado está a “ON” y forma parte de una isla energizada o apagada (“muerta”) si su estado es “OFF” o forma parte de una isla desenergizada.

6.1.2.2 *DER, MicroGrids* y Celdas

Comparado con un generador centralizado, hay numerosas diferencias en cómo debe ser conectado y controlado un *DER*, la presencia de éstos en una red de distribución puede alterar sustancialmente el flujo de las corrientes de falta y cambiar las fuentes de servicios auxiliares, de tal forma que la operación de los *DER* necesita integrarse al *DMS* mediante un sistema que garantice la operación.

Una *MicroGrid* puede definirse como una red eléctrica de baja tensión independiente compuesta por pequeñas unidades de generación distribuida, ya sean células fotovoltaicas, celdas de combustible (hidrógeno), micro turbinas o aerogeneradores, junto con dispositivos de almacenamiento de energía y cargas controlables. La integración entre *DMS* y MicroGrids

debe llevarse a cabo mediante enlaces (un nuevo dispositivo de la red) denominados *MicroGrid Central Controllers (MGCC).*

El concepto de *Celda* es considerada como un área grande dentro de una red de distribución. Hay muchos tipos de celdas de diferentes niveles de tensión que tendrán un diferente enfoque en función de sus requerimientos técnicos y de los *DER* que tengan que controlar. De forma similar a las *MicroGrids*, la integración del *DMS* y las *Celdas* pueden ser implementadas mediante enlaces que denominamos *Controladores de Celdas*.

6.1.3 Pronóstico de cargas

Hay un gran número de cargas en una red de distribución y el consumo de energía de cada una de ellas es relativamente pequeño, por ello, la medida en tiempo real de las cargas es cara. La estimación y predicción de cargas se usan para la operación y planificación de una red de distribución.

Las predicciones se dividen en periodos de tiempo cortos, medios o largos, siendo los de corto plazo, en un rango de una hora a una semana, los más importantes para aplicaciones *DMS*. Las cargas varían en el corto plazo debido al tiempo (diario, fin de semana, vacaciones), la meteorología (temperatura y humedad) y tipología de consumidores (residencial, comercial o industrial)

La introducción de la medición inteligente y los sistemas de gestión doméstica de la energía cambiarán el comportamiento de los consumidores y nos llevarán a una gestión de la demanda más dinámica lo que supondrá unas cargas más volátiles, que serán más difíciles de predecir.

6.1.3.1 Estimación de estados

La monitorización en tiempo real de los sistemas de distribución es muy limitada (actualmente), por la falta de sensores y sistemas de comunicación, lo que hace que las medidas sean insuficientes y que el sistema no sea observable en su conjunto. Una vez que el sistema estuviera completo, cualquier cifra podría ser calculada.

La introducción de generadores distribuidos, vehículos eléctricos, bombas de calor, cargas controlables, etc., proporcionan numerosas incertidumbres que con altas penetraciones pueden causar dificultades operacionales en la red. Es por ello, que la provisión de un sistema de información preciso del estado de la red para los operadores es crítico para permitirles operarlo de una manera rentable.

La estimación de estados es usada para eliminar errores en las medidas y estimar la situación del sistema en un determinado momento, las técnicas son ampliamente utilizadas en sistemas de transmisión donde las medidas son redundantes. En cambio en un sistema como el nuestro, deberemos hacer un gran número de pseudo-medidas a partir de las estimaciones y previsiones de las cargas. Muchos valores no pueden ser medidos pero pueden ser estimados.

Los métodos de estimación de estados más habituales incluyen el *Weighted Least Square (WLS)*, el *Weighted Least Absolute Value (WLAV*) y otros [4].

6.1.3.2 Análisis de flujo de energía

El análisis del flujo de energía o flujo de carga, provee una solución equilibrada de la red de energía para unas condiciones de la red específicas que incluye los conceptos de topología de la red y niveles de carga. Las soluciones del análisis de flujo de energía incluye la medición del voltaje (|V|), ángulos de fase (θ) e inyecciones de potencia activa (P) y reactiva (Q) de todos los nodos y barras, además de los flujos de energía en líneas de transmisión, cables y transformadores.

Es un elemento básico para el análisis, operación y planificación de las redes de distribución, y ha sido formulado en multitud de ecuaciones algebraicas no lineales y adecuadas técnicas matemáticas como las clásicas de Gauss-Seidel, Newton-Raphson, y otras muchas que han sido descritas en muchos libros de texto [5].

6.1.4 Faltas en el sistema de distribución.

Cuando se produce una falta en el sistema de distribución, la tensión del sistema disminuye bruscamente en una amplia área de la red que solo se recupera cuando la falta es reparada. Los sistemas de distribución utilizan protecciones que reaccionan en un tiempo típico de 500 ms. Una actuación rápida ante la presencia de estas faltas es importante a nivel industrial y comercial y menor, pero cada día más importante, en entornos domésticos.

Las faltas por corto circuitos son inevitables en cualquier sistema de distribución por lo que la interrupción en cualquier equipamiento sensible a la carga solo puede ser evitado implementando alguna de estas medidas:

- Asegurarse que el equipamiento de carga utilizado es robusto ante las transiciones y cambios de tensión.
- Utilizar disyuntores y circuitos de protección muy rápidos.

- Añadir equipamiento al circuito que permita mitigar las bajadas bruscas de tensión como *Dynamic Voltage Restorer (DVR)* o *STATCOM*.

Una interrupción prolongada puede provocar un grave siniestro con pérdidas económicas, especialmente en procesos industriales, por ello, muchos reguladores imponen penalizaciones por las pérdidas de electricidad en los consumidores.

Una red de distribución está caracterizada como segura e indemne bajo condiciones normales y anormales. El dimensionamiento, la rentabilidad y la seguridad de estos sistemas (en particular en las ciudades) dependen en gran medida de poder controlar las corrientes de un cortocircuito cuando se produce. Con el rápido incremento y variabilidad de las cargas y la introducción de generadores distribuidos, la importancia se incrementa.

Durante una falta, se produce el flujo de corriente desde la red a tierra o a otra fase de la red. Cálculos precisos de las faltas son un prerrequisito en el correcto dimensionamiento del equipamiento eléctrico, configurando dispositivos proactivos y asegurando la estabilidad. Normalmente se consideran dos tipos de faltas:

- Simétricas (balanceadas). Afecta a todas las fases por igual, por lo que la simetría del sistema se mantiene. Una guía detallada del tratamiento de las mismas se encuentra en la norma IEC 60909 [6].

- Asimétricas (desequilibradas). Este tipo de faltas incluye los cortocircuitos línea a línea (fase a fase), de línea a línea y a tierra y de línea a tierra que son las más habituales.

El *equipamiento en tecnologías de la información - Information Technology Equipment (ITE),* cada día más implantado en todos los ámbitos, es muy sensible a las variaciones de tensión en la red. Por ello, se han efectuado numerosos estudios específicos y como conclusión se ha obtenido la famosa curva *ITI (CBEMA)*[4] que especifica las variaciones de tensión en alterna que puede soportar el *ITE* en función del tiempo transcurrido desde la falta.

En estas circunstancias, los operadores de una red de distribución son conscientes de la necesidad de incrementar la velocidad de detección para aislar una falta y restaurar el

[4] http://www.itic.org/clientuploads/Oct2000Curve.pdf - The ITI (CBEMA) Curve, Information Technology Industry Council (ITI).

suministro, para lo que se utilizan reconectadores automáticos, interruptores controlados a distancia, mediciones remotas y cada día más, agentes locales.

Además de las incidencias de las faltas en el sistema, hay muchas otras perturbaciones que determinan la calidad del suministro, los problemas de *'Power Quality'* abarcan un gran abanico de problemas que pueden alterar la operación y el suministro en la red [7]. Brevemente destacamos las siguientes:

- Interrupciones cortas
- 'dips' de tensión
- 'swells' de tensión
- Tensión y corriente transitorias
- Distorsión armónica de la tensión y corriente
- 'flicker' de tensión
- Desbalanceo de la tensión
- Desequilibrio y/o salto del ángulo de fase.

6.1.4.1 Componentes para el aislamiento de una falta y restauración del servicio.

Donde quiera que se produzca una falta en una parte de la red de distribución, la corriente de fuga debe interrumpirse rápidamente, la sección donde se ha producido debe quedar aislada del resto de la red sana, y una vez que la falta ha sido eliminada, restaurar el suministro a los clientes.

- Esto se consigue con lo que llamamos de forma genérica aparamenta, que incluye: Disyuntores que son capaces de provocar y romper las corrientes de fuga.
- Reconectador, que es básicamente un disruptor con una capacidad limitada para interrumpir la falta y un patrón variable automático de disparo y cierre.
- Interruptor-seccionador que tiene una capacidad de actuación limitada para provocar una falta, aunque es capaz de hacer y romper la corriente de carga normal.
- Seccionador que es capaz de permitir y cortar la corriente de carga normal, pero no la corriente de fuga.

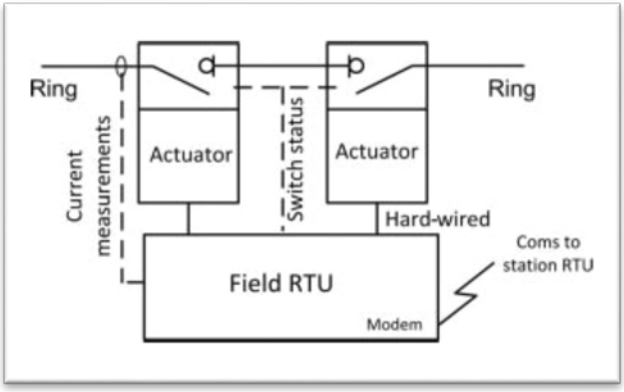


Figura 6 - Aparamenta automatizada.

En una subestación de 11 kV la aparamenta suele estar colocada en el interior y el equipamiento de protección y control de la aparamenta se sitúa en una ubicación específica en la zona de control. La aparamenta automatizada actual utiliza sistemas de gas comprimido o actuadores magnéticos.

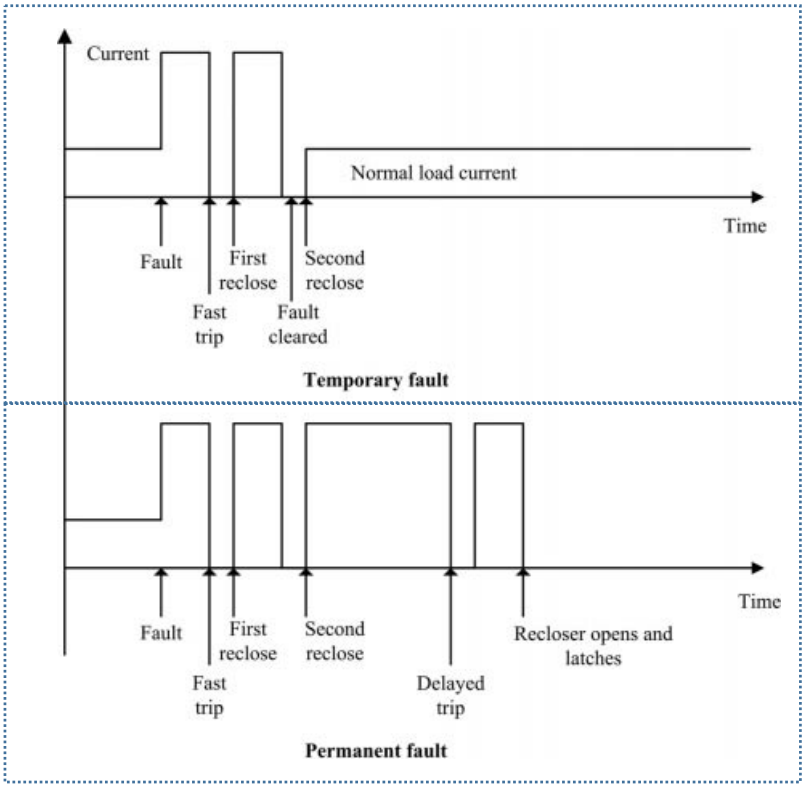


Figura 7 - Secuencia en faltas temporales y permanentes.

Muchas de las faltas en líneas de transmisión son transitorias y se “auto eliminan” una vez que el circuito es desenergizado. En muchos circuitos de distribución se utiliza aparamenta que permite ejecutar un patrón variable con secuencias de apertura y cierre, que previenen la interrupción del servicio por faltas temporales de este tipo. Si después de varias secuencias de apertura y cierre se mantiene la falta, entonces se considera permanente.

6.1.4.2 Localización de la falta, aislamiento y restauración.

En una típica red de distribución de 11 kV como la mostrada en la *Figura 9*, cuando hay una corriente de fuga indicada entre L4 y L5, el dispositivo de protección contra sobre corrientes *IED1* abre el disyuntor *CB1* y aísla toda la rama dejando interrumpido el suministro desde L1 hasta L5. Como en nuestra red no hay elementos automáticos, la restauración del suministro requiere la intervención del personal de campo y en algunas áreas pueden pasar más de 80 min hasta que lo consigan.

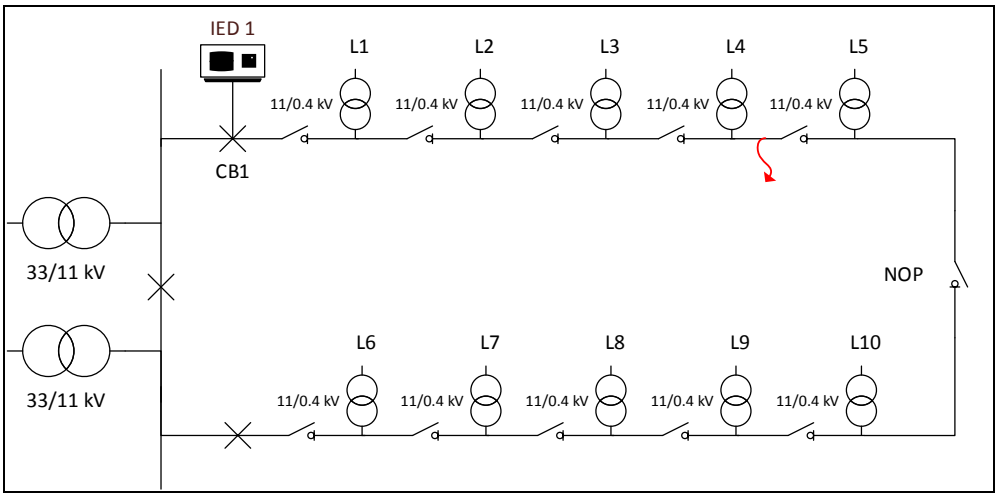


Figura 8 - Sección de una típica red de distribución.

La restauración del servicio normalmente es iniciada por llamadas de teléfono de uno o más clientes, en el área que ha ocurrido la interrupción, informando de una pérdida de suministro eléctrico. En ese momento, el personal de campo es enviado a la zona y localizarán la falta manualmente, para ello desde "aguas arriba" en las proximidades de *CB1*, irán bajando por la rama abriendo consecutivamente L1, L2 y L3, una vez identificado que en L4 está la falta, cerrarán el circuito en *NOP* para suministrar a L5. Cuando hayan reparado el daño en L4, restaurarán todo a su estado inicial.

En una sección como la anterior que estuviera completamente automatizada como la indicada en *Figura 10*, la intervención sería diferente, ya que el Agente que interpreta los datos enviados por el resto de dispositivos mediante un sistema de sondeo permanente para detectar un cambio de estado en la red.

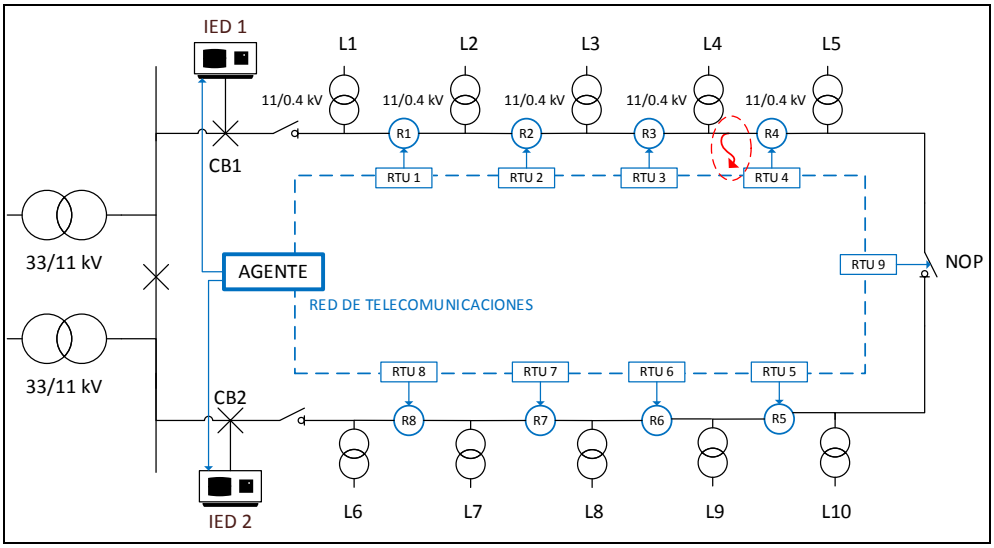


Figura 9 - Sección red de distribución automatizada.

Cuando la falta se produce en el lugar indicado, *IED1* detecta la corriente de fuga, abre *CB1* e informa al Agente, entonces éste envía comandos desde *RTU1* hasta *RTU4* para abrirse y solicita datos de la corriente y de la tensión de ellos en tiempo real y la secuencia de restauración sería el siguiente:

1. Enviar un comando a *RTU1* para cerrar *R1*, si la corriente de fuga existe iniciar un disparo, pero como no hay, *R1* permanece cerrado.

2. Comandos similares se envían a *RTU2*, *RTU3* y *RTU4* para cerrar *R2*, *R3* y *R4*. Cuando *R3* cierra, aparece la corriente fuga, lo que hace que se dispare y que se bloquee.

3. A continuación se envía un comando a *RTU9* para cerrar el *NOP*. De esta forma consiguen alimentar L5.

4. Finalmente queda aislada la corriente de fuga entre *R3* y *R4* y se restaura el suministro en L1, L2, L3 y L5.

6.1.5 Aplicaciones

La gestión de la red de distribución y la automatización de las operaciones requiere de un gran número de aplicaciones, como:

- **Sistemas de monitorización en tiempo real.** Permiten el envío a los operadores de alarmas cuando ocurren problemas graves relacionados con las tensiones en los nodos o con las condiciones de carga de los circuitos. Estos sistemas comparan las

mediciones en tiempo real con los valores o límites normales y si ocurre algún cambio anormal, genera eventos de tipo automático implementando funciones de control o envía una alerta a los operadores *DNOs*.

- **Operación del sistema.**
    - **Reconfiguración de la red.** Las redes de distribución son normalmente construidas como redes malladas pero son operadas de una forma radial, utilizando *"Puntos de enlace" (Open Points).* De esta forma la configuración de la red puede variar modificando el estado abierto/cerrado del interruptor, ya sea de forma automática o manual. Los objetivos principales de la reconfiguración de redes son:
        1. Restauración del suministro de electricidad a los clientes utilizando fuentes alternativas, en caso de fallo o faltas.
        2. Minimización de las pérdidas causadas por la potencia reactiva en un determinado momento o pérdidas de energía en un periodo de tiempo.
        3. Equilibrio de cargas entre diferentes fuentes o transformadores para equilibrar tensiones.
    - **Control Volt/VAR**. Este tipo de controles es usado para mejorar los perfiles de las tensiones y minimizar las pérdidas de las cargas. Para ello calcula el punto óptimo de funcionamiento de los reguladores de tensión, condensadores, *DER* y otros dispositivos eléctricos que permiten ajustar la respuesta a la demanda.
    - **Recoordinación de relés de protección.** Esta aplicación configura los relés de protección en tiempo real en base a unas reglas predeterminadas, esto se consigue mediante el análisis de los modos operacionales de los disyuntores, considerando la conectividad de la red en tiempo real, coordinándose con los *DER*, las *microgrids* y las condiciones atmosféricas.
    - **Operación del *DER*.** La integración del funcionamiento de los *DER* en el *DMS* tiene un gran impacto en el rendimiento de la red, ésta depende en gran medida de la configuración del sistema que se puede realizar mediante interfaz de potencia o agrupando varias *DER* en una *MicroGrid* [8] o celdas.

6.1.6 Gestión del Sistema

Para gestionar un *DMS* la plataforma debe integrar funciones del tipo *Automated Mapping (AM), Facilities Management (FM),* y *Geographic Information System (GIS),* que nos permitan vincular mapas digitales automatizados con las bases de datos de las infraestructuras.

El *sistema de gestión de interrupciones - Outage management system (OMS*), combina un *Call Center* de atención de llamadas y resolución de problemas con las herramientas del *DMS*, lo que permite identificar, diagnosticar y localizar faltas, para poder aislarlas, repararlas y restaurar el suministro. Facilita la comunicación con los clientes afectados, analiza la tipología del evento y almacena registros históricos de las interrupciones para facilitar el cálculo de índices estadísticos.

La gestión de las interrupciones es fundamental en las redes de distribución, con objetivos (y penalizaciones) para restaurar el suministro en un periodo de tiempo, en la sección donde se ha producido la falta. Los objetivos de un sistema *OMS* son:

1. Identificación de la falta.
    - Mediante el sistema tradicional basado en las llamadas telefónicas de los clientes, aunque puedan ser utilizados sistemas de voz automáticos *(Computer Telephony Integration – CTI).*
    - Sistemas automáticos integrados en la red para la detección, mediante el disparo/cierre de disyuntores, y aviso a un sistema *SCADA* o tomas de decisiones autónomas a nivel local.
2. Localización y diagnóstico de la falta.
    - En el método tradicional de llamadas de usuario, se localiza la zona agrupando las llamadas y mediante el seguimiento inverso de la topología de la red eléctrica se localiza el dispositivo de protección que se ha abierto, como puede ser un fusible, un seccionador, un disruptor de una subestación. La prioridad de la avería será calculada en función del número de clientes y de su tipología.
    - En las nuevas redes que disponen de mediciones en tiempo real de los consumos y de los parámetros de la red, se basan en sistemas *SCADA* que

informan directamente de su ubicación a los operadores de la red. El siguiente paso será la intervención automática de agentes virtuales locales que resuelvan lo que a día de hoy hacen los operadores.

3. Restauración del suministro. La intervención depende en gran medida de la severidad del problema. Si la falta es un problema sencillo, el personal de campo hace las reparaciones y restaura el suministro en un corto periodo de tiempo, pero si la falta es grave, es necesario aislar un área grande e ir restaurando el servicio parcialmente en zonas mediante los *"Puntos de Enlace"*. En estos casos se ayudan de modelos computarizados y de sofisticadas herramientas de análisis.

4. Análisis del evento y almacenamiento.

    - Una vez resuelto, se hace un análisis del evento y la información es guardada en un registro histórico que entre otros datos incluye la causa, el número de clientes afectados y la duración del corte. Esa información se utiliza posteriormente para calcular estadísticas que permitan planificar actividades de mantenimiento o para justificar el cumplimiento de los parámetros de calidad.

    - La integración de los sistemas automáticos sin duda derivará en ahorros de costes, eficiencia de las intervenciones y en una disminución de la duración de las interrupciones. La *Figura 6* representa la integración de la infraestructura *ICT* de la ***Smart Grid*** en el *OMS*.

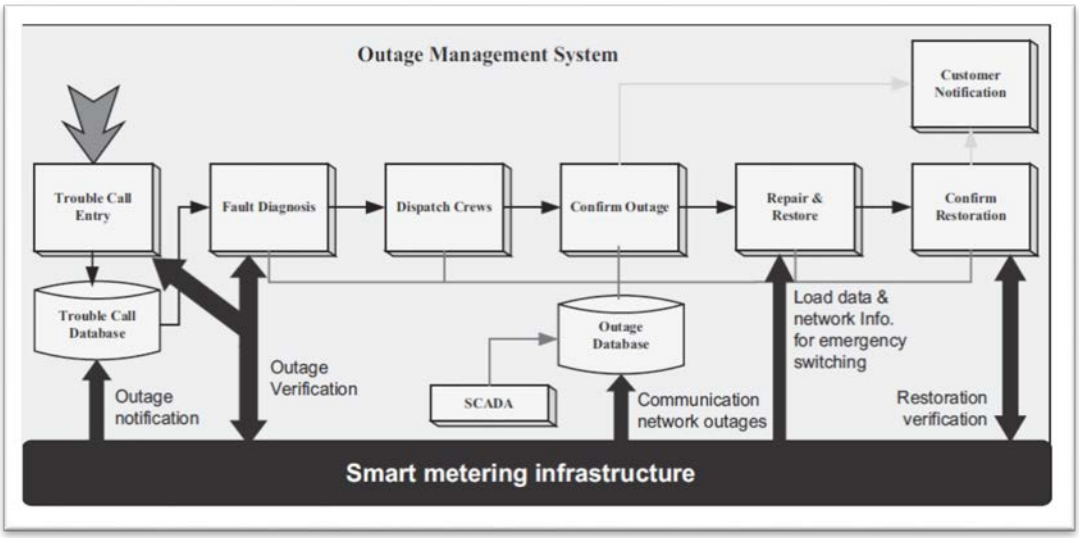


Figura 10 - Integración de la medición inteligente y el DMS.

6.1.6.1 Seguridad

Para el intercambio de datos en una ***Smart Grid***, puede llegar a ser necesario en determinados entornos, cumplir con las siguientes características de seguridad:

- Privacidad. Solo el emisor y el receptor definido puedan conocer el contenido de un mensaje.
- Integridad. El mensaje que llega al receptor en tiempo y forma, es exactamente igual al que fue enviado.
- Autenticación del mensaje. El receptor puede confirmar la identidad del emisor y que el mensaje no proviene de un impostor.
- No repudio. Un receptor es capaz de probar que un mensaje vino de un emisor en concreto y éste no puede negar que fue él, el que envió el mensaje.

Para ello se utilizan varios sistemas de seguridad y encriptación como pueden ser:

- Encriptación y desencriptación. La criptografía ha sido desde tiempo inmemorial la técnica más extendida de protección de información de adversarios.
- Autenticación. Es requerida para verificar la identidad de las partes que se comunican y evitar que impostores tengan acceso a la información.
- Firmas digitales. Una firma digital permite la firma de mensajes digitales por el emisor.

Hay varios estándares que pueden aplicarse a la seguridad del equipamiento de una subestación y muchas otras están en desarrollo. Actualmente las normas más utilizadas son:

- IEEE 1686: Estándar del IEEE para las capacidades de ciber seguridad en IEDs. Este estándar surge a partir de una propuesta de seguridad en los IED elaborada por la *NERC (North American Electric Reliability Corporation – Corporación para la Fiabilidad Eléctrica en Norte América)*
- IEC 62351: Estándar del IEC para la gestión de los sistemas de energía y el intercambio de la información asociada y comunicaciones seguras de datos. El IEC 62351 son una serie de documentos que especifican los tipos de medidas de seguridad a implementar en redes de comunicación y sistemas incluyendo varios perfiles como

TCP/IP, *MMS (Manufacturing Message Specification – Especificación de Mensajes de Fabricación)* y el IEC 61850.

6.2 Normalización para el intercambio de información.

En nuestro planteamiento es fundamental la creación de un sistema de normas y estándares que nos permitan intercambiar información y datos entre diferentes sistemas y fabricantes.

Entre los dispositivos, un sencillo *Smart Meter* debería ser capaz de comunicarse con los sistemas *SCADA* y con futuros sistemas o aplicaciones que se introduzcan en la red para mejorar el servicio.

6.2.1 Protocolos implantados actualmente

En ese sentido, tanto el IEC 62056 como el ANSI C12.22 son dos normas que describen la forma de comunicación de los medidores inteligentes *SM (Smart Meters).*

- El IEC 62056 define las capas de Aplicación y Transporte, mediante un conjunto de especificaciones denominadas *COSEM (Companion Specification for Energy Metering).*
- Por otro lado, el ANSI C12.22 especifica la forma de envío y recepción de los datos de las mediciones a y desde sistemas externos, sobre cualquier tipo de red de comunicaciones. Su arquitectura se define a continuación.

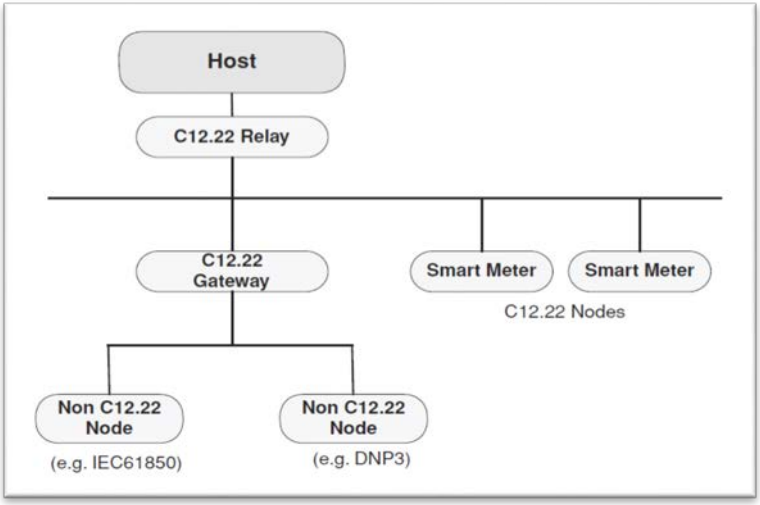


Figura 11 - ANSI C12.22 Básico, Arquitectura para medición inteligente.

Otro protocolo de mensajería de la capa de aplicación es el denominado *Modbus*, que facilita la comunicación entre dispositivos conectados sobre diferentes buses y redes.

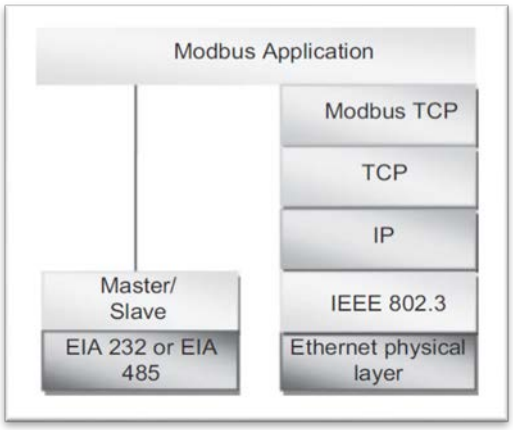


Figura 12 - Pila del protocolo Modbus

El *DNP3 (Distributed Network Protocol)* [9] es un conjunto de protocolos de comunicación desarrollado para interconectar varios tipos de equipamiento de adquisición de datos y control. Juega un papel crucial en los sistemas *SCADA*, donde es usado por los centros de control, los *RTUs* y los *IEDs*. DNP3 ha sido recientemente adoptado como el *IEEE* standard 1815–2010.

6.2.2 IEC 61850

IEC 61850 es un estándar abierto para la comunicación de subestaciones mediante el protocolo Ethernet, asegurando de esta forma la interoperabilidad del equipamiento conectado a la subestación. Las funciones están divididas en:

1. Funciones de soporte del sistema: gestión de la red, sincronización de tiempos y auto chequeo de dispositivos físicos.
2. Funciones de configuración y mantenimiento del sistema: gestión del software, configuración, parametrización y modos de prueba.
3. Funciones operativas y de control: configurar los parámetros del *switch*, gestión de alarmas y gestión delos eventos de faltas.
4. Funciones de automatización de procesos: protección, bloqueo, y gestión de la demanda de cargas.

IEC 61850 utiliza un modelo orientado a objetos para describir la información disponible en las diferentes partes del equipamiento de la subestación y el controlador. La norma contiene 10 capítulos descritos en la *Tabla 1*:

Tabla 1 - Capítulos norma IEC 61850

Capítulo	Descripción
1	Introducción y resumen
2	Glosario
3	Requisitos generales como requisitos de calidad, condiciones medioambientales incluyendo inmunidad a interferencias electromagnéticas y servicios auxiliares.
4	Definición de los requisitos de ingeniería como los tipos de parámetros (sistema, procesos y funcionales), herramientas de ingeniería (especificaciones del sistema, configuración del sistema y documentación y configuración de los IED) y aseguramiento de la calidad.
5	Requisitos de comunicación para funciones y modelos de dispositivos, incluyendo enfoque de nodo.
6	*Substation Configuration Language – Lenguaje de Configuración de Subestaciones (SCL).* Cada dispositivo en la subestación debe proveer su configuración conforme al SCL.
7	Estructura de comunicación para la subestación y equipamiento de suministro. Dispone de 4 partes definidas: el modelo de información para la automatización de la subestación, modelo de aplicación para nodos lógicos, el modelo de la estructura de la base de datos de dispositivos y las clases de nodos lógicos y clases de datos.
8 y 9	Definiciones para objetos de asignación y servicios para *Manufacturing Mapping Specifications – Especificaciones Asignación para Fabricación (MMS)* [5] y Ethernet. Definiciones de asignación de mensajes *GOOSE (Generic Object Oriented Substation Event – Evento de Subestación Orientado a Objeto Genérico)* y *GSSE (Generic Substation Status Event – Evento de Estado de Subestación Genérico)* a Ethernet. Asignación de servicios usados para la transmisión de valores analógicos muestreados.
10	Test de conformidad.

Además de definir los protocolos de comunicación, define una estructura de datos como aparece en el apartado *a)* de la *Figura 13*. El modelo de dispositivos comienza considerando un dispositivo físico y a continuación se definen los dispositivos lógicos de éste.

Cada dispositivo es entonces asignado a una de las 86 clases de nodos lógicos definidas en el IEC 61850, las cuales disponen cada una de su nombre. Finalmente, los datos relacionados con cada uno de los nodos lógicos se especifican individualmente, como se demuestra en el apartado b) del esquema.

En un *IED* se pueden encontrar dispositivos lógicos que realizan medidas, protección, monitorización, y registro de datos. Cada dispositivo lógico dispone de múltiples nodos

[5] Un estándar ISO 9506 utilizado en la industria.

lógicos reflejando sus funciones. Incluso los dispositivos lógicos asociados con dispositivos lógicos de protección está especificados como un único nodo y están divididos en 40 nodos lógicos que incluyen distancia, diferencias de potencial, sobre corrientes, y más.

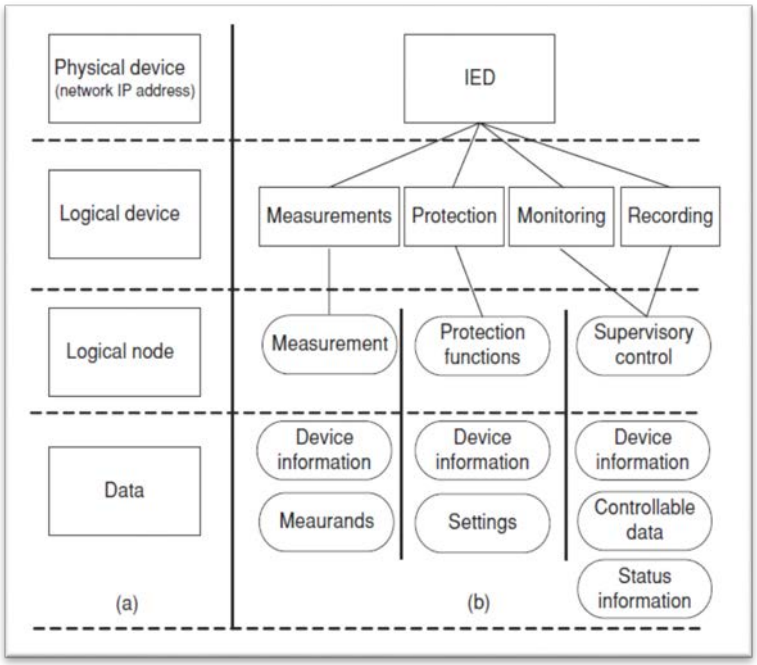


Figura 13 - Estructura de datos del IEC 61850

6.2.3 Common Information Model - CIM

El *Modelo de Información Común (CIM)* es un conjunto de normas para representar componentes del sistema eléctrico basado en el *Lenguaje de Modelado Unificado – Unified Modeling Language (UML)* [10] y sus usos principales son:

- Facilitar el intercambio de datos de la red del sistema eléctrico entre organizaciones.
- Permitir el intercambio de datos entre aplicaciones dentro de una organización.
- Intercambiar datos del mercado entre organizaciones.

Las normas que aúna son la IEC 61970 [11] modelo semántico que describe los componentes de un sistema a nivel eléctrico y las relaciones entre ellos, la IEC 61968 [12] que extiende el modelo para cubrir aspectos del intercambio de datos del software como seguimiento de activos, programación de trabajo y facturación a clientes y la IEC 62325 [13] que cubre los datos intercambiados entre los participantes del mercado de la electricidad.

El *CIM* surge de la necesidad de las operadoras eléctricas de intercambiar datos de una forma habitual en un mercado desregulado como el actualmente implantado en Europa y Norte

América, que necesita asegurar una operación fiable en la redes interconectadas.

Las compañías eléctricas usan una gran variedad de formatos para gestionar su actividad y almacenar e intercambiar sus datos, por lo que necesitan multitud de traductores para importar y exportar datos entre los múltiples sistemas y aplicaciones. Este crecimiento exponencial en complejidad debido al incremento del número de aplicaciones y registros ha requerido establecer un estándar de intercambio de datos.

Como hemos indicado, el *CIM* se basa en el *UML* que es usado para modelar una amplia variedad de elementos en el ciclo de vida del desarrollo de software incluyendo estructuras de datos, interactuaciones de sistemas y casos de uso. El modelado no está orientado a la implementación de una tecnología en particular y puede ser realizado en múltiples plataformas.

Uno de los conceptos fundamentales para entender *UML* son los diagramas de clase y sus entradas, mediante la creación de objetos comprensibles comunes (formas básicas) los conceptos que subraya el *CIM* son construidos poco a poco.

El uso de *UML* se complementa con el lenguaje *XML eXtensible Markup Language* combinado con el *RDF Resourde Description Framework.*

6.3 Proyecto PRICE

El *Proyecto PRICE*[6] *(Proyecto Conjunto de Redes Inteligentes en el Corredor del Henares),* liderado por Gas Natural Fenosa e Iberdrola, surge de la necesidad de buscar soluciones que permitan la correcta integración de recursos de generación distribuida en la red eléctrica.

En el ámbito de las ***Smart Grids*** se plantea la necesidad de acometer una nueva arquitectura de red inteligente, desarrollando diversas herramientas para dar soporte a la operación de esta red y especialmente desde el punto de gestión de la propia red y sus activos, buscando soluciones que sean interoperables y comunes a empresas distribuidoras en un ámbito geográfico compartido.

El PRICE se articula en cuatro subproyectos que cubren la supervisión y automatización de la red (PRICE-RED), la gestión energética (PRICE-GEN), la gestión de la demanda (PRICE-GDE) y la generación distribuida (PRICE-GDI).

[6] http://www.priceproject.es/

El PRICE-RED pretende diseñar y desarrollar una nueva plataforma interoperable única de supervisión y automatización de la red de centros de transformación (subestaciones secundarias), mediante la integración de sistemas y equipos que permitan la supervisión y automatización completa de la red de distribución. Sus objetivos técnicos se centran en el desarrollo, despliegue e integración de una solución avanzada de supervisión y automatización de los centros de transformación.

6.3.1 ENERGOS

Muchos de los conceptos desarrollados por PRICE fueron tomados del Proyecto ENERGOS, finalizado recientemente, que sentó las bases de los elementos básicos que permiten hacer posible una red inteligente capaz de gestionar en tiempo real y de forma óptima el nuevo modelo de red eléctrica, cumpliendo los requisitos más exigentes en la gestión de energía.

Como resultado de la investigación se obtuvieron entre otros, desarrollos relativos a adquisición y tratamiento de información en tiempo real, planificación y operación de red, mantenimiento predictivo (redes aéreas, redes subterráneas...), supervisión y control automático de *microrredes*, análisis geográfico de la demanda, comunicaciones o simuladores.

6.3.2 Power Grid Distribution Nodes (PGDIN)

La arquitectura distribuida de nodos inteligentes ha sido denominada *PGDIN* [14],, siendo éstos nodos situados en las subestaciones. Básicamente se trata de entes que permiten efectuar actividades de gestión de la red de una forma inteligente y colaborativa, mediante el procesado distribuido de datos semánticos. Diseñados para ejecutar aplicaciones de *Business Intelligence - Inteligencia de Negocio (BI)* combinada con *Semantic Web Technologies – Tecnologías Semánticas de la Red (SWT)* y elementos de *Computación en Red - Grid computing (GC).*

Las *tecnologías semánticas - Semantic Technologies*, juegan un papel fundamental en esta tarea porque permiten a los dispositivos de diferentes fabricante entenderse cuando intercambian datos e información.

Para ello, los autores usan el *CIM (Common Information Model)*. Todas estas normas se han agrupado y ordenado generando una versión semántica denominada ENERGOS *Ontology* [15] (Ontología ENERGOS), de acuerdo con los requisitos de publicaciones al respecto [16] [17]: Esta nueva ontología se especifica el lenguaje de especificaciones ontológicas *OWL-*

DL, que forma parte de *OWL* [18].

En la literatura relacionada con la computación en red o distribuida (*Grid Computing*), los nodos de procesado de información son meros esclavos de un nodo coordinador, sin embargo, siguiendo lo descrito en la arquitectura propuesta por el proyecto ENERGOS, cada *PGDIN* es un nodo inteligente que procesa y gestiona recursos por sí mismo.

El nodo inteligente incorpora una variedad de potentes tecnologías que permiten gestionar los recursos, procesar eventos y datos de la red de una forma lógica y autónoma. Su arquitectura y tecnologías quedan resumidas en el esquema de la *Figura 14*:

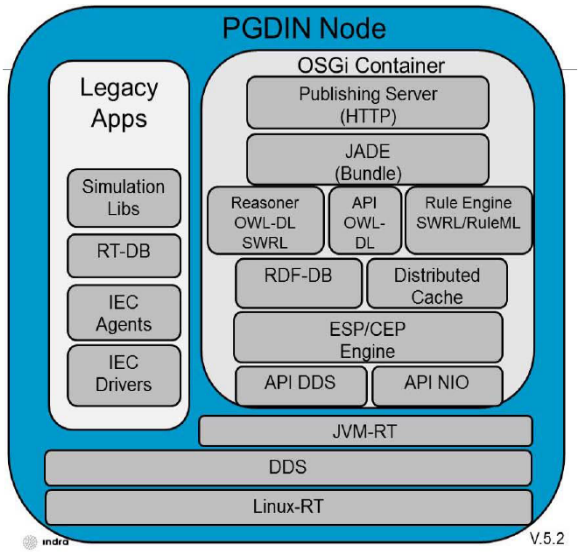


Figura 14 - Esquema lógico nodo PGDIN

Como no puede ser de otra forma, dispone de unas rutinas de comunicación y un esquema de colaboración con otros semejantes (habitualmente instalados en subestaciones), que les permite conocer de forma global el estado de la red de distribución.

Dentro de las rutinas creadas para atender a las necesidades de información y datos, el *PGDIN* está configurado para trabajar de forma estable recopilando información de la *"red local"* que cuelga de él. En el caso de recibir un aviso, comunicación o cualquier otro estímulo, analiza la información y actúa en función de la programación incluida.

El funcionamiento del ecosistema está claramente influenciado por la capacidad de comunicación del *DDS* implementado, el cual interconecta los *PGDIN*s. De esta forma, el tamaño de los mensajes intercambiados debe ser lo menor posible.

Además, en función de cómo es la información que recibe (crítica, alta, normal) su

tratamiento debe hacerse en tiempo *Real Time (RT)*, en casi tiempo real *Quasi Real Time (Q-RT)* o histórico (*H*), cuando los datos se van almacenando y después se tratan (cuando sea necesario).

Otra de las limitaciones o cuestiones a tener en cuenta es el contenido de la base de datos de información que debe ser cargada previamente en el *PGDIN*, en base a las medidas históricas y comportamientos del nodo que gestiona, de la parte de la red de la cual el representa el conocimiento. Las decisiones que se deben implementar tienen que basarse en un análisis histórico.

Una aportación imprescindible alrededor de un *PGDIN* es conocer sus Puntos de Conexión. Los Puntos de Conexión son nodos en una red mallada que disponen de un interruptor que, en caso de necesidad, une dos líneas de transmisión que normalmente se hallan separadas, independientes.

Esta información es básica para un *PGDIN*, ya que desde ellos es posible obtener información de los recursos energéticos que no están conectados directamente al *PGDIN* dado. Por lo tanto, el descubrimiento de los *CPs* no es una tarea trivial, ya que requiere inferir el conocimiento del repositorio de datos local de los *PGDNs*, por medio de un uso intensivo de la *SWT*, que forman parte del *PGDIN*.

Su ejecución contempla el uso de la implementación efectuada basada en tecnologías semánticas:

1. La base de datos del *PGDIN* (con un perfil *OWL*) define un vocabulario valido que define la información que puede ser intercambiada entre los *PGDIN* para el descubrimiento de los *CPs*, por lo tanto este perfil *OWL* define el alfabeto para construir consultas *SPARQL* y deducciones *SWRL*. Ambas, consultas e inferencias (deducciones, conclusiones) son procesadas por el módulo *OWL-reasoner*[7], que es parte del diseño del *PGDIN* (ver *Figura 14*).
2. Cuando el *PGDIN* descubre (infiere) un nuevo conocimiento, ejecutando el "razonador" sobre la Ontología ENERGOS, la nueva información es almacenada en el mismo repositorio de datos local (base de conocimiento basada en *Jena TDB*). Esta información se guarda como entidades *RDF*, que son estancias de las clases definidas

[7] En este caso basado en Pellet.

en el perfil *OWL*.

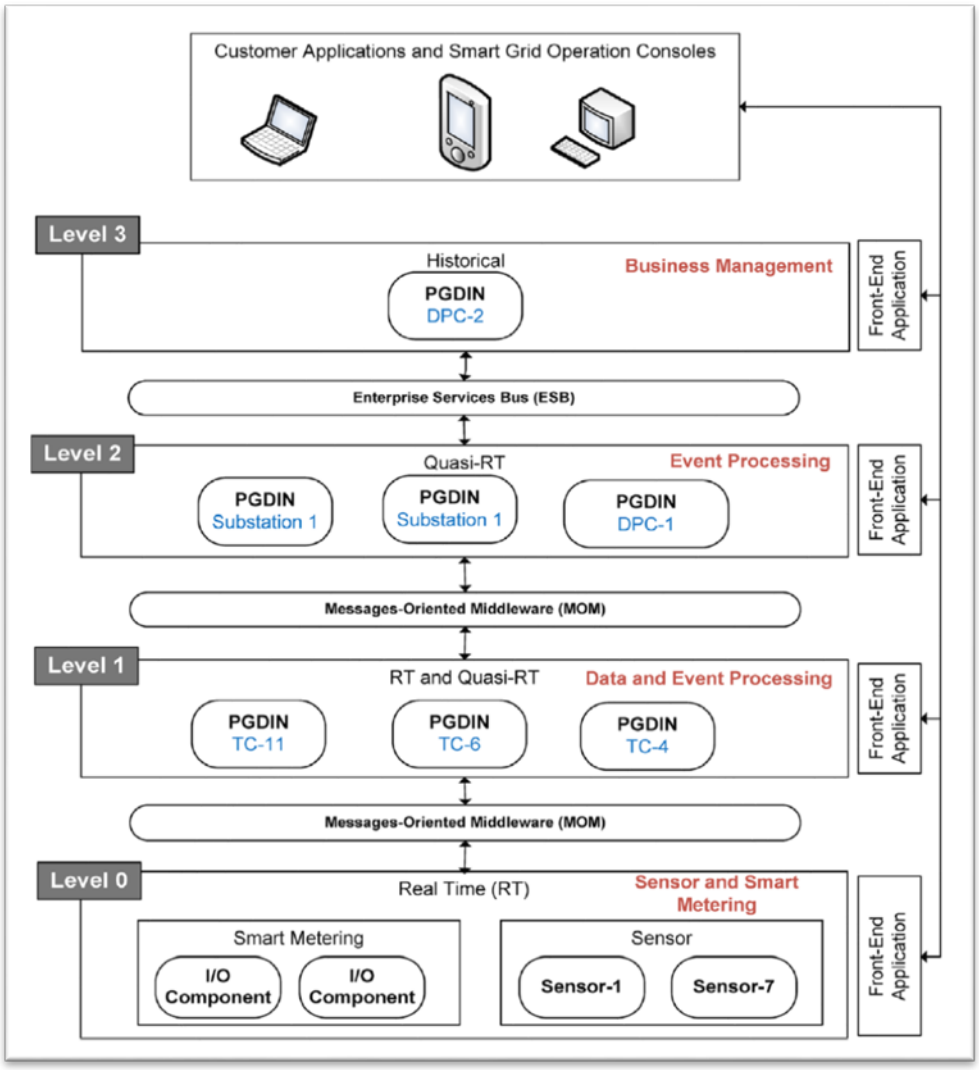


Figura 15 - Arquitectura ENERGOS.

El nodo inteligente está distribuido a lo largo de la arquitectura. Cada capa simultáneamente ofrece decisión (acorde con el uso de los datos) y la abstracción de la localización (subestaciones de transporte o distribución, *Centro de Proceso de Datos - Data Processing Center (DPC)* o la red entera). Una breve explicación de cada una de las capas en el PGDIN se expone a continuación:

- Nivel 0 –*Level 0.* Comprende los sensores y la capa de medición inteligente (ver *Level 0* en la *Figura 24*), en la que los datos son procesados en tiempo real – Real Time (RT). Geográficamente, los datos son capturados y medidos por los dispositivos de red correspondientes.

- Nivel 1 – *Level 1*. Describe la capa de procesado de eventos y datos (ver *Level 1* en

la *Figura 24)*, en la que el procesado de datos inteligente y el procesado de eventos complejos presentan una combinación de operaciones en *RT* y casi – *quasi RT*. Geográficamente este nivel se lleva a cabo en las subestaciones secundarias.

- Nivel 2 – *Level 2*. Representa la capa de procesado de eventos (ver *Level 2* en la *Figura 24*), en la cual las tareas procesadas de eventos complejos se ejecutan en quasi RT. Geográficamente esta capa se puede ejecutar en subestaciones secundarias, subestaciones primarias y en el DPC.

- Nivel 3 – *Level 3*. Identifica la capa de gestión de negocio (ver *Level 3* en la *Figura 24*), en ella se trabaja con datos históricos, por lo tanto, estos procesos orientados a negocio se centran el toda la red.

Todas las capas se comunican con aplicaciones de cliente y con las llamadas consolas inteligentes de operadores mediante aplicaciones *front-end.*

6.4 Protocolo OSPF

El protocolo ***OSPF*** *(Open Shortest Path First – Primer Camino más Corto Abierto)* [19] es uno de los protocolos de routing (encaminamiento) más usado en routers (encaminadores) de paquetes en redes TCP/IP, y en definitiva en Internet.

De forma un poco más específica, **OSPF** se puede denominar un protocolo de tipo *IGP*, usado en *Sistemas Autónomos (ASs)*, entornos cerrados donde se comunican entre sí varios routers, donde prima la rapidez de recalcular las rutas en el caso de que la red cambie (por el motivo que sea). En los entornos abiertos, que forman parte del grupo denominado *Exterior Gateway Protocols (EGPs)*, es mucho más importante poder establecer determinadas políticas de encaminamiento en función de nuestras necesidades y disponer de información enrutamiento global.

La especificación del protocolo **OSPF** es pública y la primera versión publicada en Octubre de 1989, está definido en la RFC 1131. La especificación Versión 2 de este protocolo está recogida en la RFC 1247, publicada en Julio de 1991.

El protocolo **OSPF** se ejecuta directamente en la capa IP, y cuenta por ello con el número 89 de los protocolos IP según la *IANA* [20].

6.4.1 Algoritmos de estado de enlace – Link State

EL **OSPF** utiliza los algoritmos de estado de enlace para calcular las bases de datos de encaminamiento, se identifican por ser distribuidas y replicadas en todos los miembros de su *AS.*

Cada router contribuye a la creación de la base de datos describiendo su propio entorno: el conjunto de enlaces activos en los segmentos de su red local IP, y los routers vecinos con los enlaces asignados a cada uno de ellos con un coste. Este algoritmo toma su nombre precisamente de esto, cada enlace entre routers (nodos) tiene un coste asignado y el router publicita a los demás del coste de sus enlaces. Cuando todos los routers (nodos) de un segmento publican el coste de sus enlaces, se crea el mapa actual de la red donde se ubican. El coste de un camino entre dos routers (nodos) es la suma de los enlaces por los que tiene que pasar.

Desde el conocimiento del mapa de la red, cada router ejecuta un cálculo del camino más corto para llegar a un destino utilizando el algoritmo *Dijkstra* [21]. Los algoritmos de este tipo son generalmente buenos por sus buenas propiedades para la convergencia[8]. Cuando la red cambia, las nuevas rutas son encontradas rápidamente con un mínimo de recursos.

6.4.2 Link State Advertisements (*LSAs*)

En el corazón de este protocolo hay una base de datos distribuida y replicada que describe la topología del enrutamiento, esto es, la colección de routers en el dominio y como están interconectados. Cada router en el dominio es responsable de describir su parte local de la topología mediante *anuncios del estado de enlaces - Link State Advertisements (LSA).*

Éstos son fiablemente distribuidos a todos los otros routers del dominio en un proceso llamado “inundación fiable” (*reliable flooding*). Conjuntamente los *LSAs* generados por todos los routers conforman la llamada *base de datos de estado de enlaces – link-state database (LSdb),* que es idéntica para cada uno de ellos, excepto durante los breves periodos de convergencia.

Usando la *LSdb* como entrada de datos cada router calcula su propia tabla de encaminamiento IP, permitiendo el envío correcto del tráfico IP.

Cuando una red (dominio) se encuentra en estado de reposo, esto es que ninguno de los

[8] Se denomina convergencia al proceso de encontrar el siguiente nuevo salto cuando cambia la red.

routers (nodos) o enlaces van a quedar fuera de servicio, el único tráfico intercambiado por routers **OSPF** vecinos son paquetes de datos cortos denominados *Hello* y el refresco ocasional de algunas partes de la *LSdb*

Cada router en **OSPF** tiene un *identificador – RouterID*, que consiste en un número de 32 bit que identifica unívocamente cada router del dominio. Aunque este identificador puede ser cualquier cosa, en la práctica siempre se asigna a una de las direcciones IP del Router.

Cada uno de los routers del dominio origina uno o más *LSAs*, para describir su entorno local del dominio. Como hemos mencionado, todos los *LSAs* juntos conforman la *LSdb* que es utilizada para los cálculos de encaminamiento. Para organizar la *LSdb*, ya que puede estar compuesta por miles de mensajes *LSA*, y para permitir una actualización y eliminación ordenada, cada uno de ellos dispone de información propia que los identifica, así como información de la topología del dominio.

El formato de los mensajes *LSA*, definidos en las especificaciones, tiene una cabecera común de 20 bytes, en la que se incluyen como campos fundamentales:

- *LS Type - tipo de mensaje LS.* Define el tipo de mensaje *LSA* que se envía, hay 5 tipos diferentes en función de la configuración del router (y del dominio)
- *Link State ID – Identificador del mensaje LS*. Identifica unívocamente el mensaje *LSA* enviado por un router, de otros del mismo tipo que haya enviado.
- *Advertising Router Field – Campo del router anunciante.* Es fijado con el identificador del router (Router ID).

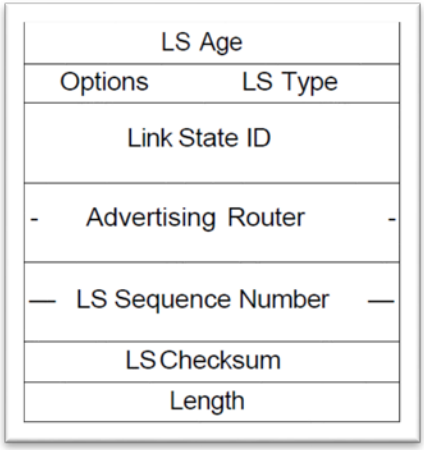


Figura 16 - Cabecera de LSA

Los routers del dominio solo pueden manipular sus propios mensajes *LSA*. Cuando un router desea actualizar uno de los *LSA* que ha generado, cuenta con un campo denominado *LS Sequence number – Numero de secuencia LS* que es incrementado. Cuando otro router del dominio recibe el *LSA* modificado (actualizado), compara el número de secuencia, siendo el más alto siempre el más reciente (el último), y por lo tanto cualquier dato obtenido de un *LSA* con un número inferior debe ser sustituido por éste.

Un *LSA* puede corromperse (modificar su contenido) durante el tránsito por la red y provocar estragos que pueden derivar en cálculos de rutas erróneos, agujeros negros o paquetes de datos en bucles. Para detectarlo, se añade información redundante al *LSA* en el campo *LS Checksum* en forma de suma de comprobación o cálculo de paridad.

Como hemos indicado, un *LSA* solo puede ser modificado (o eliminado) por el router que lo ha generado, no obstante en caso de caída o fallo de éste, los *LSA* generados por él siguen circulando en el dominio por un periodo de tiempo limitado. El campo denominado *LS Age – Edad del LS*, indica el tiempo que pasó desde la última vez que fue actualizado y se utiliza para gestionar su eliminación en caso necesario.

De esta forma, bajo circunstancias normales, cada *LSA* en la *LSdb* es actualizado por lo menos cada 30 minutos (según un parámetro configurable), y si un *LSA* no ha sido actualizado antes de una hora (valor normal de la variable denominada *MaxAge* también configurable), entonces se asume que no es válido y se elimina de la *LSdb* (recordamos que ésta es común para todos los routers del dominio). Esta propiedad también es utilizada por el router para eliminar un *LSA*, denominado *premature aging – envejecimiento prematuro*, que consiste en fijar *LS Age* con el valor de *MaxAge*.

El contenido máximo del *LSA* viene dado por el valor que se incluya en el campo *Length – Longitud*, de 16 bits, el tamaño máximo del mensaje podrá ser desde unos 20 bytes (el tamaño de la cabecera) hasta unos 65.000 bytes[9]. Sin embargo, como norma general el tamaño de los mensajes suele ser mucho menor siendo lo normal unos pocos cientos de bytes.

[9] Se restringe el tamaño porque después hay que encapsularlo en un paquete IP que debe tener como máximo 65.535 bytes

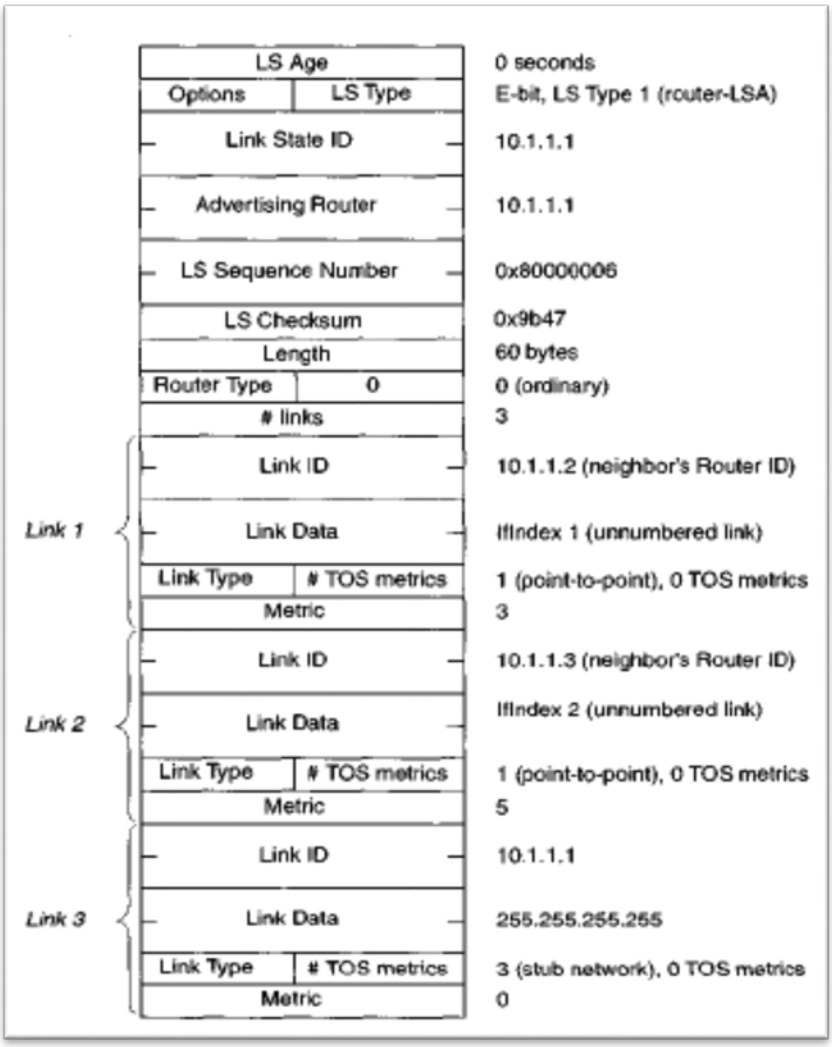


Figura 17 - Mensaje LSA con dos enlaces.

Como podemos ver en *Figura 18*, junto con la cabecera, se envían las características de los enlaces (interfaces) del router, que campos tienen y se envía también el enlace propio. En nuestro ejemplo el nodo tiene dos enlaces “externos” y uno “interno”, por lo tanto se envía información de tres.

En las conexiones punto a punto con líneas en serie, cada router origina un *LSA* único, el *LSA*- Tipo 1 denominado router-*LSA*, para notificar los enlaces activos y su métrica, las direcciones IP y los vecinos, sus características son:

- A los enlaces (interfaz) no se les asigna una dirección IP, no se hallan en ninguna subnet y pueden ser numerados como queramos, por ejemplo: un par de números. El primero podría corresponder con el valor *MIB-II Iflndex*[10] y el segundo con el coste

[10] Management Information Base – Tabla II – ValosIflndex. MIB es la tabla de gestión y configuración de un router **OSPF**.

de salida que se le ha asignado al interfaz (enlace).

- Los routers se identifican (como es habitual) con una de las direcciones IP de los interfaces.

Cada enlace contiene su coste en un campo denominado *Metric – Métrica*, en un valor desde 1 a 65.535 que indica el coste relativo de enviar paquetes de datos por ese enlace. Cuanto mayor sea el valor, menor cantidad de datos serán encaminados por ese enlace. La métrica de cada enlace debe ser configurada por el operador de la red y los criterios a seguir pueden ser variados: retardos de la línea de transmisión, ancho de banda, coste de envío de tráfico y otros, únicamente se debe tener en cuenta que la métrica tenga un significado claro, ya que las rutas calculadas son la suma de los valores de las distintas métricas en cada uno de los enlaces.

A título informativo comentar que la métrica de un enlace puede ser asimétrica, esto es, que el valor de un enlace en un extremo puede no ser el mismo que en el otro, pero no suele ser lo habitual.

6.4.3 La base de datos LS (*LSdb*)

El conjunto de todos los *LSA* de todos los routers de una red conforman la *LSdb*, y cada router tiene una exactamente igual a la de los demás, excepto durante el periodo de convergencia. Las *LSdb* son intercambiadas entre routers vecinos cuando un router descubre a otro; después de eso, la *LSdb* se sincroniza mediante el procedimiento llamado *reliable flooding*.

La *LSdb* proporciona una completa descripción de la red: de los routers, de los segmentos de red y de cómo están interconectados. Partiendo de la *LSdb* uno puede dibujar un mapa completo de la red, observándola uno puede conocer inmediatamente el estado de todos los routers de la red.

Tabla 2 - Ejemplo LSdb con 5 routers en red.

LS Type	Link State ID	Adv. Router	LS Checksum	Nº secuencia LS	*LSA*ge
Router-*LSA*	10.1.1.1	10.1.1.1	Ox9b47	0x80000006	0
Router-*LSA*	10.1.1.2	10.1.1.2	Ox219e	0x80000007	1,618
Router-*LSA*	10.1.1.3	10.1.1.3	Ox6b53	0x80000003	1,712
Router-*LSA*	10.1.1.4	10.1.1.4	Oxe39a	Ox8000003a	20
Router-*LSA*	10.1.1.5	10.1.1.5	Oxd2a6	0x80000038	18

6.4.4 Comunicación entre routers OSPF

Los routers **OSPF** se comunican intercambiando directamente paquetes directamente sobre la red IP, sin utilizar servicios tipo UDP o TCP como hacen otros protocolos. Cuando un router recibe un paquete IP con el número de protocolo igual a 89, sabe que ese paquete contiene datos **OSPF** y a continuación quitando el encabezado IP, obtiene directamente el paquete **OSPF**.

La cabecera de los paquetes **OSPF** tiene un tamaño de 24 bytes y en ella está contenida la siguiente información:

- Campo del tipo de paquete **OSPF**. Hay cinco tipos diferentes, todos usados en la sincronización de la *LSdb*, los tipos son:

    1. *Hello packets – Paquetes Hello*, se usan para descubrir y mantener las relaciones con los vecinos.
    2. *Database Description packets – Paquetes descriptivos de la base de datos.*
    3. *Link State Request packets – Paquetes de solicitud LS.*
    4. *Link State Update packets – Paquetes de actualización LS.*
    5. *Link State Acknowledgment packets – Paquetes de reconocimiento LS.*

- El *RouterID*[11] del que hace el envío. De esta forma el router que lo recibe de dónde viene ese paquete.
- *Checksum*. Permite al receptor determinar si se ha producido algún daño del paquete en tránsito, de ser así, el paquete es descartado.
- Campos de autenticación. Por seguridad, estos campos permiten al receptor verificar, sin género de duda, que el paquete fue enviado por el router que parece identificado en el *RouterID* y que además el contenido del paquete no ha sido modificado por un tercero.
- Identificativo de área **OSPF**. Permite al receptor asociar el paquete recibido con el nivel apropiado de la jerarquía **OSPF** y asegurar que ésta ha sido configurada consistentemente.

[11] Identificador del Router

6.4.4.1 Descubrimiento y mantenimiento de vecinos

En **OSPF** un router descubre a sus vecinos enviando periódicamente paquetes *Hello* por todos sus interfaces, por defecto, envía uno por cada interfaz cada 10 s, aunque este parámetro es configurable en el parámetro *HelloInterval*. Un router conoce la existencia de su vecino cuando recibe de éste un *Hello* a su vez.

La parte del protocolo **OSPF** responsable del envío y recepción de los paquetes *Hello* es llamado *protocolo Hello* de **OSPF**. La transmisión y recepción de paquetes *Hello* también permite a un router detectar el fallo de uno de sus vecinos; si el tiempo transcurrido sin recibir un *Hello* de un vecino excede el parámetro establecido *RouterDeadInterval* (es configurable pero su valor por defecto es 40 s), el router que no lo recibe para de anunciar la conexión con el router que debería haberlo enviado y empieza a enviar paquetes de fallo a su alrededor.

Sin embargo, en la mayoría de las ocasiones el fallo de la conexión con un vecino será conocido más pronto por el *protocolo de datos de enlaces – data-link protocol*. Detectar los fallos de los vecinos en el tiempo correcto es crucial para el correcto funcionamiento de **OSPF**, el tiempo de detección de los fallos de los vecinos determina el tiempo de convergencia, desde el momento en el que el resto de la maquinaria (la inundación de *LSAs* actualizados y el rehacer la tabla de encaminamiento a partir de los nuevos cálculos), requiere para encaminar los paquetes alrededor del fallo por lo menos unos segundos.

El *protocolo Hello* de **OSPF** también establece que los vecinos sean consistentes de la siguiente manera:

- Asegurando que la comunicación entre vecinos es bidireccional.
- Asegura que los routers se pongan de acuerdo en los parámetros *HelloInterval* y en el *RouterDeadInterval*, de forma que asegura que cada router envíe los mensajes lo suficientemente deprisa para que en el caso de que se pierda ocasionalmente uno de ellos, no se cierre el enlace erróneamente.

6.4.4.2 Sincronización inicial de la base de datos.

Cuando la conexión entre dos vecinos se establece por primera vez, cada uno de ellos debe esperar a que sus respectivas *LSdb* se sincronicen antes de enviar tráfico de datos por la conexión. En caso contrario, las discrepancias entre ambas bases de datos pueden provocar

que calculen tablas de enrutamiento incompatibles, bucles o agujeros negros.

OSPF utiliza una estrategia que consiste no en enviar la *LSdb* completa la primera vez que se encuentran dos vecinos, sino que solamente envía los encabezados *LSA* y el vecino solicita únicamente aquellos más recientes. Este procedimiento es más eficiente que simplemente enviar la base de datos completa, y se denomina *Database Exchange – Intercambio de la base de datos* en las especificaciones **OSPF**. Ésta es la tarea que más a menudo debe realizar la máquina de estados finitos de un vecino.

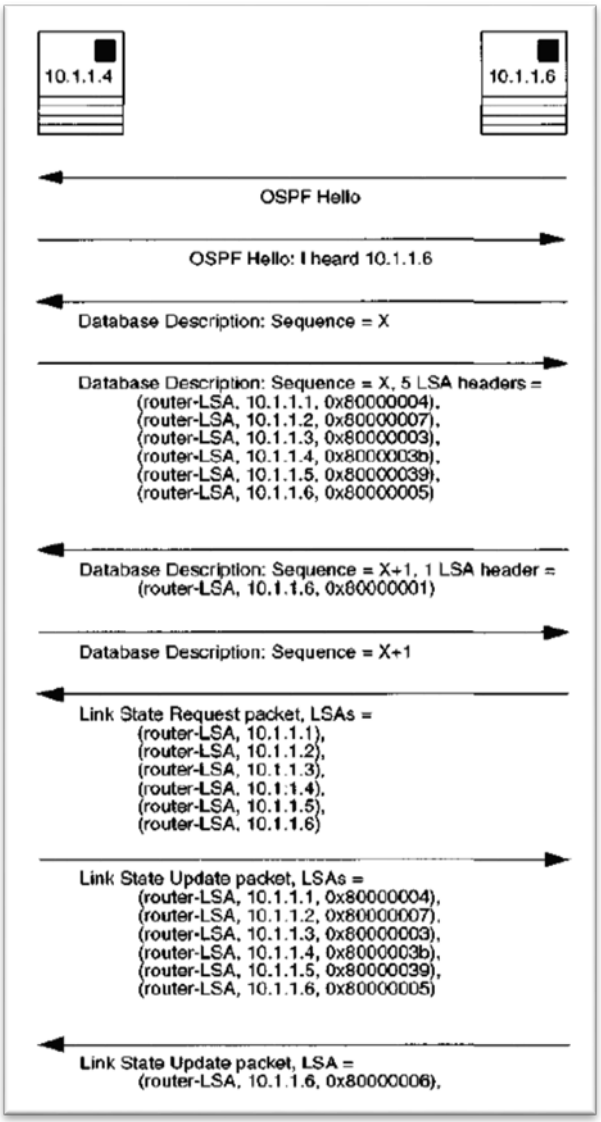


Figura 18 - Ejemplo Database Exchange

Cuando se toma la decisión de sincronizar las tablas, cada uno de los dos vecinos hace dos cosas: envía las cabeceras de todos los *LSA* actualmente en su base de datos de acuerdo con el orden establecido y comienza a inundar la conexión con actualizaciones de *LSA*, esto se

hace para asegurar que el intercambio de la *LSdb (Database Exchange)* finaliza en un periodo de tiempo concreto.

En particular, un router sabe cuál de los *LSAs* de su vecino no tiene y cuáles son los más recientes. El router envía paquetes *Link State Request* al vecino solicitando los *LSAs* deseados y éste le responde *"inundando"* con paquetes *Link State Update* de sus *LSAs*.

Después de enviar una secuencia completa de paquetes *Database Description* al vecino, y haber recibido la secuencia completa de paquetes *Database Description* del vecino, y tener todos sus paquetes *Link State Request* contestados por *Link State Updates* del vecino, el router declara que la conexión se ha sincronizado y advierte que ya está dispuesto a cursar tráfico de datos. En este punto, el vecino se dice que es completamente adyacente al router emisor; al comienzo del procedimiento *Database Exchange*, los dos routers eran meramente adyacentes.

6.4.4.3 Inundación fiable- reliable flooding.

Los *LSAs* actualizados con nueva información son enviados al dominio de routers mediante un procedimiento denominado *reliable flooding*. Este procedimiento comienza cuando un router desea actualizar uno de sus propios *LSAs*, esto puede ser porque el estado ha cambiado, por ejemplo, uno de los enlaces del router se ha vuelto inoperativo o bien, el router desea eliminar uno de los *LSA*, lo que consigue poniendo el campo *LSAge* en un valor igual a *MaxAge*.

En cualquier caso, una vez hechas las modificaciones en los *LSA*, lo empaqueta en un *Link State Update*, que puede o no contener otros *LSAs*, y entonces lo envía por todos sus interfaces produciendo la *inundación*. Cuando uno de sus vecinos recibe el paquete *Link State Update*, examina cada uno de los *LSA* contenidos en la actualización. Por cada *LSA* incorrupto, conocido el *LS Type*, y siendo más reciente que la copia (si hay alguna) residente en su propia base de datos, éste instala el *LSA* nuevo en la *LSdb* propia y envía un *reconocimiento – acknowleged (ACK)* de vuelta al router que lo envió. Tras ello, el receptor reencapsula el *LSA* en un nuevo paquete *Link State Update* y lo envía por todos sus interfaces, excepto por el que se lo envió. Este procedimiento se repite hasta que todos los routers del dominio han actualizado el *LSA*.

Con el fin de lograr la fiabilidad en el dominio, un router retransmitirá periódicamente un *LSA* enviado.

Este método es muy fiable en cuanto a errores, esto quiere decir que aun cuando se produzcan errores de transmisión o cuando uno de los routers o de los enlaces de la red fallen, la red continúa funcionando normalmente, la *LSdb* continúa estando sincronizada y la mayor parte del tráfico se mantiene en un nivel aceptable. **OSPF** consigue la robustez gracias a las siguientes características:

- **OSPF** fluye por todos los enlaces. Gracias a ello, el fallo de uno de los enlaces no perturba significativamente la sincronización de la base de datos, ya que las actualizaciones de *LSA* fluyen simultáneamente en rutas alternativas alrededor del enlace caído.

- Debido a errores de software, un router puede accidentalmente borrar uno o varios *LSA* de su *LSdb*. Para asegurar que el router eventualmente recupera la sincronización con el resto de routers del dominio, **OSPF** obliga que todos los miembros del dominio que generen *LSAs*, deben actualizarlos cada 30’ (valor configurable) incrementando obligatoriamente el número de secuencia del *LSA* y reinundando todo el dominio.

- Para detectar la corrupción, un *checksum* calculado por el router que lo origina se incluye permanentemente en el *LSA*, y cuando un receptor lo recibe tras una inundación, verifica el checksum. Un cálculo erróneo significa que ha sido corrompido, se descarta, no se envía *“ACK”* y espera recibir uno correcto.

- Para preservar un excesivo tráfico de control en la red, provocado por ejemplo por un elemento que cambia rápidamente de estado (un enlace entre dos routers que se abre y cierra), **OSPF** impone límites inferiores para originar un *LSA*, en particular se utiliza el parámetro *MinLSInterval* (configurable pero con un valor por defecto) que impide que se actualice como mucho una vez cada 5 seg.

6.4.5 Cálculo de la ruta

El coste de una ruta establecida entre routers es la suma de los costes (métricas) de los enlaces que recorre un paquete desde un router origen a uno destino.

Las métricas de un enlace son estáticas, el protocolo **OSPF** no responde a una sobrecarga de la red modificando sus valores, solo modifica dinámicamente las rutas alrededor de los enlaces o nodos que fallan, pero no hace asignación dinámica de los valores de los enlaces. En una red distribuida esto generaría que cambiándose dinámicamente los valores para una

determinada ruta, todos se moverían en masa hacia ella y cuando esta estuviera "llena" se volvería a cambiar y vuelta de nuevo, pudiendo llegar a generar saturaciones todavía más graves en algunas rutas.

El Algoritmo de Dijkstra es un sencillo algoritmo que calcula de forma eficaz y simultánea todas las rutas más cortas posibles a todos los destinos dentro de un dominio, el algoritmo incrementalmente calcula un árbol de los caminos más cortos según lo indicado en *Figura 18*.

Comienza añadiéndose a sí mismo el router que hace el cálculo, y añadiendo los routers vecinos a la lista de candidatos siendo los costes de la ruta iguales a los de los enlaces con sus vecinos. El router en la lista de candidatos con el menor coste es entonces añadido al árbol y los vecinos de esos routers son examinados para incluirlos (o modificarlos) en la lista de candidatos. De esta forma el algoritmo se repite hasta que la lista de candidatos está vacía.

El Algoritmo Dijkstra cuando está examinando el enlace, puede colocar un destino en la lista de candidatos o modificar la entrada de un destino en la lista de candidatos. Esta operación requiere ordenar la lista de candidatos porque siempre deseamos saber que destino tiene un coste menor.

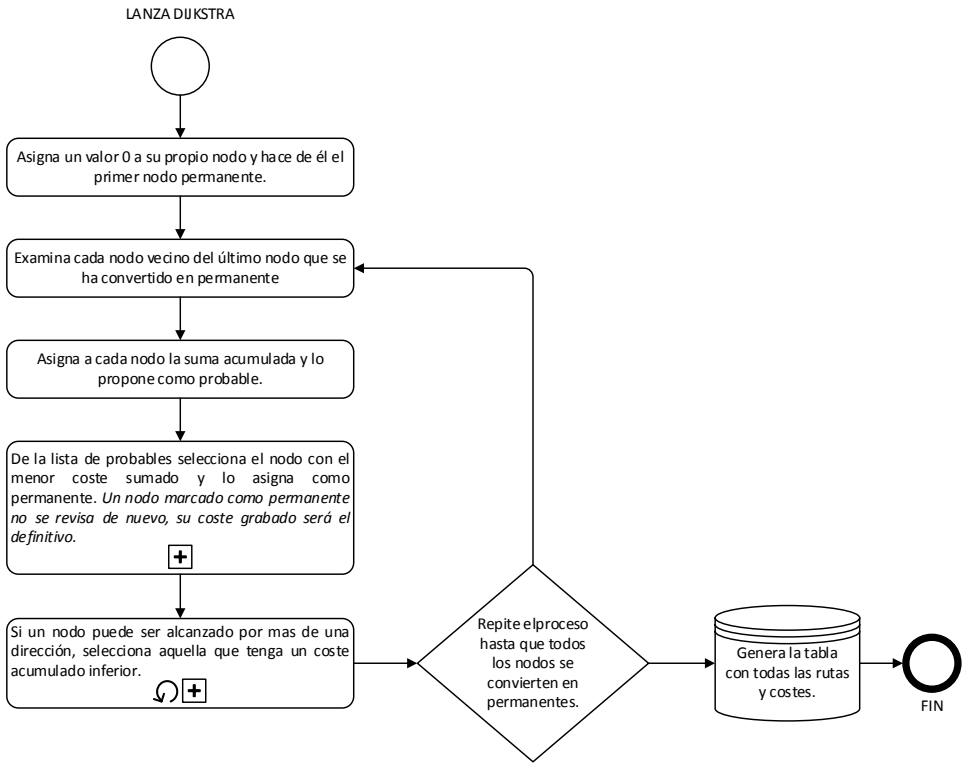


Figura 19 - Flujo de datos Dijkstra

Un router **OSPF** conoce el camino completo de cada destino, sin embargo, en el paradigma de enrutamiento salto a salto de IP, solo el primer salto es necesario para cada destino. Un ejemplo típico de la tabla de enrutamiento es la generada en la *Figura 19.*

Destination	Next Hop(s)	Cost
10.1.1.1	10.1.1.1	5
10.1.1.2	10.1.1.2	3
10.1.1.4	10.1.1.2 10.1.1.5	4
10.1.1.5	10.1.1.5	1
10.1.1.6	10.1.1.2 10.1.1.5	10

Figura 20 - Tabla típica de enrutamiento de un nodo.

6.4.6 Otros datos de interés sobre OSPF.

Existen en Internet muchas otras configuraciones de red que no son las *punto a punto* que hemos considerado hasta ahora en nuestra descripción. Otras tecnologías de enlace son por ejemplo: *Ethernet, 802.5 Token Ring*, anillos *FDDI*, subredes *Frame Relay, ATM, SMDS*, paquetes de radio, y otras.

OSPF corre sobre estas tecnologías de una forma diferente a como lo hace en enlaces *punto a punto*, su configuración y funcionamiento se organiza de forma diferente en las siguientes clases: *subredes punto a punto, subredes de difusión, subredes multiacceso de no difusión - nonbroadcast multiaccess (NBMA*) y *subredes punto a multipunto*.

En algunas cuestiones el funcionamiento de **OSPF** en *subredes de difusión* tiene ventajas con respecto a lo que hemos visto hasta ahora en las *subredes punto a punto*, si bien su configuración es más compleja, las principales relacionadas con el protocolo *Hello* de **OSPF** son: descubrimiento automático de vecinos, eficiencia, aislamiento, convergencia más rápida.

La clase *subredes de difusión* utilizan el concepto *Designated Router – Router Designado* y el *Backup Desgnated Router – Respaldo del Router Designado*. Esto es, en vez de sincronizar entre sí la *LSdb* todos los routers del dominio, como hemos visto hasta ahora, un router se elige para sincronizar la *LSdb*, mantenerla actualizada y después la retransmite al

resto y otro se selecciona de respaldo por si el principal deja de funcionar.

El *MIB* es la lista completa de parámetros configurables en **OSPF**, está organizada en 12 grupos y contiene 99 variables, sin embargo más de la mitad (61) son de solo lectura. En la práctica, únicamente se modifican el *RouterID*, las direcciones de los interfaces (variable ***OSPF****ifipAddress*) y los costes de los interfaces (***OSPF****ifMetriValue*).

El método estándar para configurar y monitorizar los protocolos y dispositivos de Internet es el *Simple Network Management Protocol (SNMP) – Protocolo de Gestión de Redes Simple*. El SNMP maneja los datos de gestión de un dispositivo de Internet.

OSPF implementa una gran cantidad de medidas de seguridad para evitar que el dominio sea atacado por otros routers no autorizados. Implementa autenticación criptográfica, verificación de mensajes, gestión de claves públicas o privadas y firmas digitales.

El tamaño máximo de una red **OSPF** depende de las indicaciones del vendedor, muchos de ellos incluyen el máximo de equipos que pueden instalarse, lo que sí es importante saber es que cuanto más grande sea el área, más recursos consume. A título informativo algunas áreas han alcanzado las 350 máquinas y otros vendedores recomiendan no exceder los 50 routers.

7 DESARROLLO DE LA INVESTIGACIÓN

Los métodos y algoritmos de reconfiguración de redes de distribución son desde la aparición de las redes inteligentes objeto de numerosas investigaciones y desarrollos.

La reconfiguración de una red de distribución es un proceso que consiste en cambiar el estado de las conexiones de la red a nivel de subestación, para reencaminar la energía eléctrica después de un fallo en el suministro o de algún otro tipo de criterio de optimización. Las técnicas de reconfiguración son una herramienta fundamental para operar el sistema de distribución al menor coste posible e incrementar la seguridad y la fiabilidad del sistema.

En este trabajo hemos querido plantear una implementación práctica en un modelo basado en inteligencia distribuida con agentes *PGDIN* embebidos en *Centros de Transformación* (subestaciones secundarias), que están especialmente diseñados para trabajar en red con un sistema de toma de decisiones horizontal, colaborativo.

En particular nuestra aportación es analizar la implementación en ese tipo de *agentes* del protocolo **OSPF** ampliamente utilizado en Internet para efectuar el encaminamiento (*routing*) de paquetes de datos en capa IP entre nodos de la red. Mencionaremos el algoritmo que utiliza para calcular las rutas, *Dijkstra*, pero queremos hacer hincapié en otras de sus características fundamentales como es la utilización de una base de datos distribuida y única y el sistema de intercambio de información entre nodos de la red.

7.1 Líneas de investigación - *State of the Art*

La base de la investigación de este proyecto cuenta con numerosas referencias en artículos científicos y publicaciones. A continuación detallamos aquellas que por sus referencias (nº de veces nombradas) o por su innovación según nuestro criterio, convienen destacar.

7.1.1 Multi agentes

De acuerdo con todas las previsiones de la Unión Europea dentro de su programa marco *Towards smart power networks, Lessons learned from European research FP5*, la descentralización de la red de distribución será un mecanismo que nos permitirá coordinar, controlar y monitorizar las redes eléctricas malladas para proporcionar al cliente final el mejor servicio posible [22].

En este contexto, los agentes en las redes eléctricas pueden ser entendidos como sistemas

multisensoriales que permiten generar conocimiento útil para tomar decisiones. Este punto de vista se hace imprescindible para los operadores supervisores de la red. Los agentes deben funcionar de forma autónoma.

La implantación de los agentes inteligentes [23] en las ***Smart Grid***, consta de programas de software que pueden gestionar, controlar y monitorizar diferentes componentes de la red y son capaces de comunicarse unos con otros para resolver complejos problemas, como el balanceo de carga dinámico o conocer cómo actuar de acuerdo a su experiencia previa preprogramada.

Actualmente el objetivo consiste en incrementar la flexibilidad de la red, de tal forma que las zonas monitorizadas pueden ser redefinidas, por ello, los autores han diseñado un sistema multi-agente [23] compuesto por cuatro clases diferentes de agentes: *agente de control - control agente, agente de distribución de energía - power distribution, agente de usuarios - user agent* y *agente de base de datos - database agent*. De esta forma se hace una clara diferenciación de sus responsabilidades, aunque comparten el mismo objetivo, asegurar las cargas críticas cuando el sistema sufre un corte de energía.

Previamente, se define el concepto de celda, como un conjunto de fuentes, cargas, interruptores – conmutadores y líneas que componen el componente más simple que puede ser gestionado por un agente. De esta forma, el comportamiento del agente está enfocado en optimizar el valor general de su celda y varios agentes de celdas interconectadas están coordinados para seguir una determinada estrategia.

La arquitectura multi-agente ha sido diseñada y desarrollada con dos aspectos clave: generalización y escalabilidad. El primero significa que la arquitectura es común y permite el desarrollo de nuevos agentes que sean requeridos que cubran nuevas necesidades, el segundo, la escalabilidad, se consigue mediante un servicio de replicado basado en un sistema *maestro esclavo - master-slave.*

Como hemos mencionado, una de las claves del sistema multiagente es el análisis de la normalidad, lo que es el comportamiento normal de la red. Un componente de normalidad define cómo un objeto debería comportarse de acuerdo con el evento de interés, considerando una situación anormal (non-normal) como sospechosa o anómala. Esta definición de normalidad nos permite inferir que algo va mal, desde un punto de visto de monitorización, si una situación no está definida (o aprendida) por el agente de normalidad

es detectada, en otras palabras, definimos (o aprendemos) la normalidad para detectar lo anormal.

El uso del concepto de normalidad, nos lleva para su análisis al uso de *lógica borrosa – fuzzy logic*, un modelo matemático ampliamente utilizado para resolver incertidumbres y vaguedades en los problemas del mundo real. Por otro lado, es un modelo adecuado para trabajar con datos facilitados por las capas inferiores, cuyos componentes son habitualmente imprecisos. A partir del análisis de unos datos que sabemos que son normales, se pueden definir situación fuera de lo normal, organizándolas en cinco estadios: Absolutamente anormal, posiblemente anormal, sospechoso, posiblemente normal o absolutamente normal; actuando en cada momento de una forma determinada.

Las variables que vamos a controlar son: voltaje, intensidad de cada una de las fases, el desequilibrio de corriente entre fases, detección de fusibles fundidos y detección de fraude. La detección de fraude se calcula conociendo lo que es entregado a la red por el transformador y restándole el sumatorio de las medidas de los contadores inteligentes, más un % de pérdidas.

Para cada una de ellas, se fijan unos umbrales que representan cada uno de los estadios antes mencionados, y se actúa en consecuencia.

La arquitectura multi-agente se estructura en 3 capas:

- Capa de perceptiva. Principalmente compuesta por los sensores y medidores situados en la subestación que nos permiten obtener los datos que necesitamos. Estas mediciones se encapsulan en eventos (unidad básica de información) que son enviados mediante comunicaciones a la siguiente capa para su interpretación.

- Capa conceptual. En esta capa se implementan los agentes responsables de monitorizar la distribución de las subestaciones. Esta capa notifica las alarmas que sean justificadas, en función de la configuración, a la siguiente capa. Para ello cuenta con el “agente de registro” que se encarga de almacenar la información recibida de la capa anterior en una base de datos, la cual una vez disponible mediante un agente de carga, se evalúa por el agente de normalizadas para efectuar el análisis. Como se puede apreciar en esta capa coexisten múltiples agente junto con la información a tratar: *information fusion agent, pattern agent, voltage normality agent intensity normality agent unbalanced intensity detection agent blown fuse detection agent*

fraud detection agent.

- Capa de soporte para toma de decisiones. En esta capa se usa toda la información disponible obtenida de la capa anterior, para que los operadores o personal de supervisión puedan fácilmente gestionar las situaciones anormales.

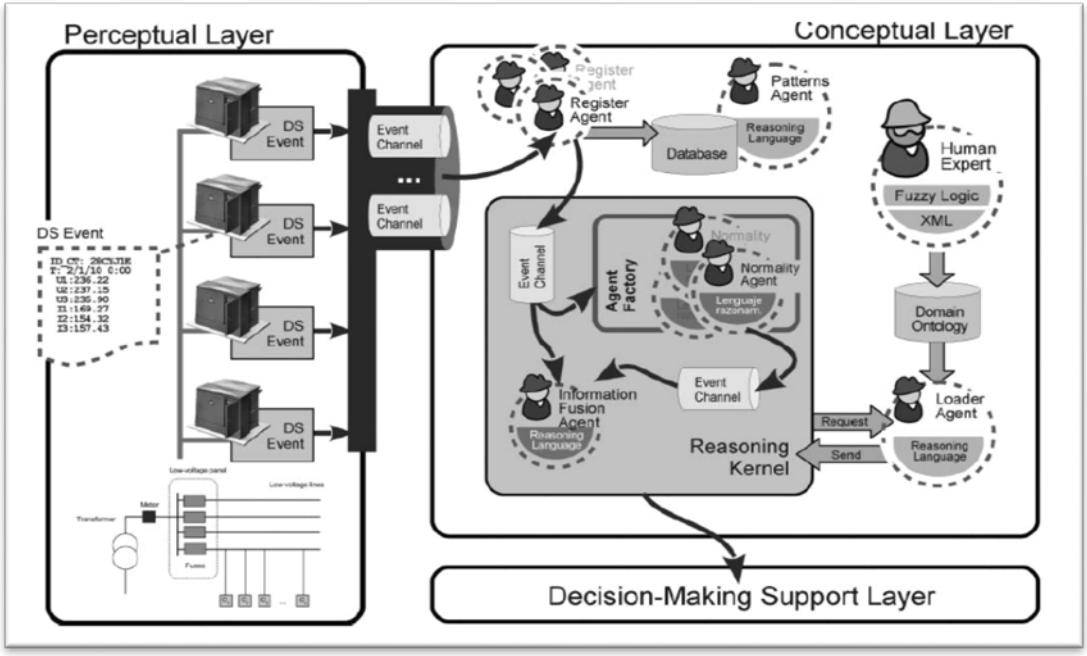


Figura 21 - Esquema de una arquitectura multiagente.

7.1.2 Reconfiguración de la red de distribución a nivel inferior.

La propuesta denominada *MultiAgent Systems (MAS*) [24], propone dotar de inteligencia no solo a las subestaciones, si no a los elementos que haya "aguas abajo", interruptores y cargas.

Esta propuesta organizativa crea dos grandes grupos, por un lado los *denominados Local Agent (LAG)*, formados por *Load Agents* para cargas de la red y *Switch Agents* para las conexiones - interruptores y los *Global Agents (GAG)* para las subestaciones.

En cada uno de los *GAG* de la red se implementa un algoritmo de reconfiguración, que cuando tiene conocimiento de una falta en un determinado punto de la red, empieza a reconfigurarla para asegurar el suministro en determinadas cargas críticas.

A este nivel de detalle de toma de datos e información, los *Load Agents y Switch Agents* se comunican entre sí para, en caso de falla localizar la ubicación donde se ha producido. Tras ello, se lo comunican a las *Global Agent* (situado en la subestación), y mediante el algoritmo de reconfiguración que tiene cargado, seleccionada cual sería la ruta más adecuada y

reconfigura “su red local” para dar servicio a cargas críticas (Hospitales, Industriales, etc.)

Esta propuesta requiere de un uso intensivo de comunicaciones y datos en tiempo real que a día de hoy creemos no factible.

7.1.3 Metadatos de intercambio de información

La ***Smart Grid*** demanda una interoperabilidad sintáctica para conseguir físicamente un intercambio de datos y una interoperabilidad semántica para entender e interpretar adecuadamente su significado [15]. Ésta es un conglomerado del legado actual y de las arquitecturas por venir que requerirán ser representadas conjuntamente en la red eléctrica y además deberá permitir la colaboración entre diferentes entidades del sistema para poder desarrollar actividades complejas. Todo ello basados en el estándar definido en las normas IEC 61970 e IEC 61968, *CIM* y en el IEC 61850.

En una arquitectura de nodos inteligentes desplegada en la estructura de la red de distribución de la ***Smart Grid***, cada uno de ellos tiene un perfil de la ontología global y además dispondrán de capacidad lógicas que les permitirán tomar decisiones a nivel local para los requerimientos en tiempo real y casi-tiempo real.

Como hemos mencionado, el proyecto ENERGOS diseñó una red inteligente que permitía gestionar la nueva red eléctrica en tiempo real con un enfoque de flujos de información multidireccional, el proyecto ENERGOS contemplaba cuatro capas en su modelo, si bien el elemento fundamental era el *PGDIN*.

Cada *PGDIN* debe ser dotado de mecanismos de procesado semántico que le permitan intercambiar información con otros nodos en una arquitectura abierta de ***Smart Grids*** y facilitar un modelo semántico para la toma de decisiones.

El paradigma de una red multi-agente [25] introduce el concepto de agente inteligente de un nodo de la red que debe ser capaz de:

- Gestionar una base de datos interna, propia.
    - La base de conocimiento interna del agente basa su modelo de datos en *CIM XML* utilizando un modelo semántico de datos basado en *UML*[12]

[12] La última versión del modelo semántico está disponible en http://cimug.ucaiug.org/ y para manejarlas se puede utilizar la herramienta de http://www.cimtool.org/

- Intercambiar información y datos con otros nodos y componente que no tienen que ser necesariamente de su red.
    - Los mensajes de comunicaciones que intercambian los agentes son compatibles con el estándar *FIPA*[13] *(Fundation for Intelligent Physical Agents)* y por lo tanto entienden y usan el protocolo *Agent Communication Language (ACL)*. Ese protocolo soporta el *Message Transport Service (MTS)*
    - *MTS* puede utilizar diferentes *Protocolos de Transporte de Mensajes - Message Transport Protocols (MTP*), para efectuar la entrega física de los mensajes. Actualmente los *MTPs* incluyen HTTP y WAP, entre otros. Sin embargo, los MTPs no garantizan la entrega fiable entre participantes de un sistema, p.e. los agentes *FIPA*. Los *Servicios de Mensajería de Empresas Enterprise - Messaging Services (EMS)* o el *Middleware Orientado a Mensajes - Message-Oriented Middleware (MOM)* proveen alternativas de mensajería fiables.
- Sacar conclusiones de las tramas de datos recibidas en el tiempo asignado y con los requerimientos funcionales de cada nodo.
- Ser autónomo en sus decisiones ante la aparición de eventos inusuales.
    - Tercer paso: Agentes *BDI*. Desde que se especula con que cualquier nodo inteligente en la ***Smart Grid*** tiene que ser autónomo, proactivo y reactivo ante a determinados eventos inusuales, cualquier agente presenta un proceso de inferencia basado en estados cognitivos del mundo. En particular, se asume que un agente de la red es un agente *BDI* [26], actualmente ha y varias plataformas que soportan este tipo de agentes como *JADE6* o *JADEX7*.

Como hemos mencionado anteriormente, el *PGDIN* permite una toma de decisiones propia en base a los datos recibidos, puede colaborar con otros nodos para gestionar eventos y analizar situaciones que lo requieran. Su arquitectura incluye tres componentes semánticos principales que permiten una toma de decisiones semántica, estos son: un conjunto de reglas *SWRL (Semantic Web Rule Language)*, una ontología *OWL-DL (Web Ontology Language – Description Logic, Lenguaje Ontológico de la Red – Lógica Descriptiva)*, que caracterizan

[13] http://www.fipa.org/repository/aclspecs.html

el perfil de una base de conocimiento y un razonador de tipo *OWL-DL/SWRL* que es un motor que permite hacer consultas sobre dicha base.

La toma de decisiones autónomas para la que ha sido diseñado el *PGDIN*, se basa en una recopilación de información a través de un sistema de mensajería multimedia *DDS*, de almacenamiento convertido en objetos Java y procesado por un motor de procesamiento de eventos. El motor detecta patrones de comportamiento inusuales como cambios repentinos en las cantidades medidas, faltas, cortes o problemas de suministro. El esquema es el indicado en la *Figura 25*.

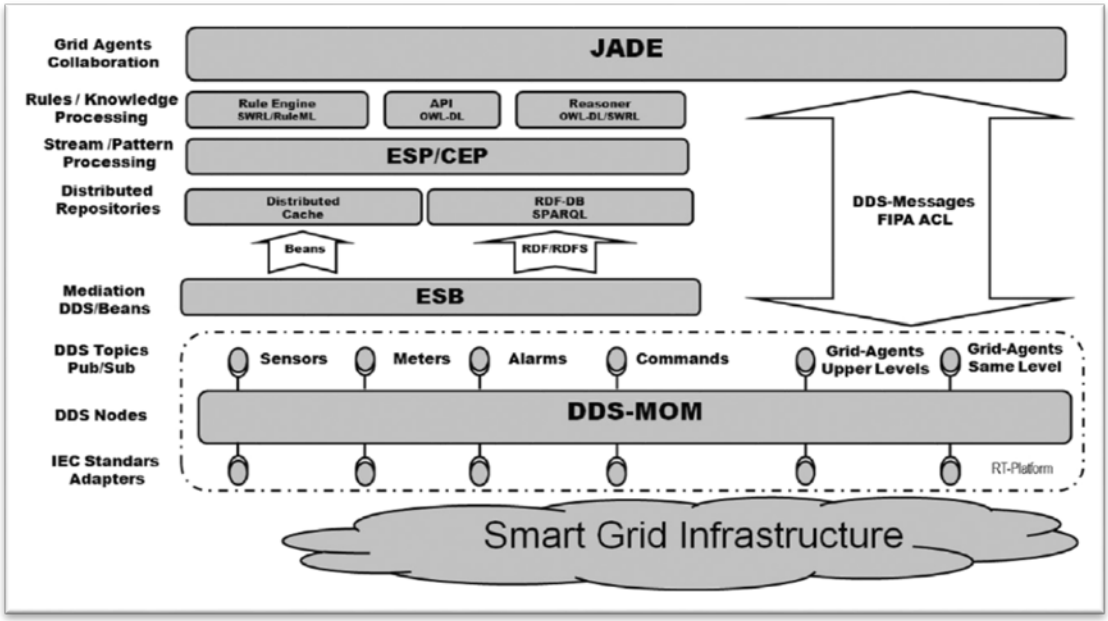


Figura 22 - Esquema de comunicaciones PGDIN.

El *PGDIN* gestiona las mediciones correspondientes a potencia activa y reactiva, tensión y corriente. Además proporciona a los operadores del sistema conocimiento para diagnosticar problemas e identificar soluciones.

La introducción de las validaciones como reglas *SWRL* en el modelo permite que la información sea consistente, ya que el "razonador", basado en software *Pellet*[14] comprueba la consistencia del modelo semántico, de acuerdo con los tramos de tiempos configurados e inferir nueva información, ya que el motor de normas (en este caso *Jess*[15]) ejecuta la validación de las normas *SWRL* obteniendo no solo cálculos (como en las hojas de cálculo normales), si no que identifica desviaciones de la información relacionada de la red. En

[14] http://clarkparsia.com/pellet/
[15] http://www.jessrules.com./

definitiva, podemos decir que el *PGDIN* no es más que un contenedor *OSGI (Open Services Gateway Initiative – Iniciativa de Intercambio de Servicios Abiertos)*

7.1.4 Algoritmos de reconfiguración

7.1.4.1 Dijkstra

Introduce el concepto de *Distribution System Planning (DSP)* para la selección óptima de las rutas de alimentación, el número de alimentadores, el tamaño de la subestación (nodo) y su localización [27].

De todos los algoritmos disponibles, esta metodología metaheurísitca por naturaleza es muy flexible, robusta, y minimiza el coste de inversión para su implementación. El algoritmo Dijkstra empleado aísla la sección donde se ha producido la falta, y el suministro es restaurado en el resto del sistema.

Éste método es mejor que cualquiera de los anteriores porque depende más del número de ramas que de nodos, si bien las rutas óptimas son obtenidas para minimizar el coste total, para ello es necesario hacer mediciones de tensión y de corriente.

El coste total de la planificación del sistema distribuido objeto de estudio es la suma de tres variables: el coste fijo anual incluido el coste de las líneas (ramas) y de las subestaciones (nodos), el coste por las pérdidas de energía y el coste de interrupción del servicio.

El autor propone resolver el algoritmo de manera fija en una red con 25 puntos de carga – nodos (transformadores de 10kV/0,4 kV) y 42 rutas posibles o segmentos que parten desde una Estación de Transferencia de la Red de Transmisión (25kV/10,5kV) que es considerada el nodo raíz, siendo la carga total de la red de 2,55 MVA.

Una vez cargados los datos de cada una de las ramas, en función de los criterios fijados anteriormente, obtienen una tabla que identifica cada uno de los nodos con la ruta más corta que los une con el nodo 1 (raíz).

7.1.4.2 CSGSA

El objetivo es resolver en dos pasos el problema de la reconfiguración en grandes redes de distribución, el primero aplicando el algoritmo de *Dijkstra* para buscar el camino más corto, aplicando a continuación el algoritmo del núcleo de los cromosomas y después el algoritmo genético. De esta forma, la solución a la búsqueda de la ruta óptima se denomina *Core Schema Genetic Shortest-path Algorithm (CSGSA)* [28].

Para resolver el problema, utilizan una programación dinámica, según muestra el esquema de la *Figura 26*.

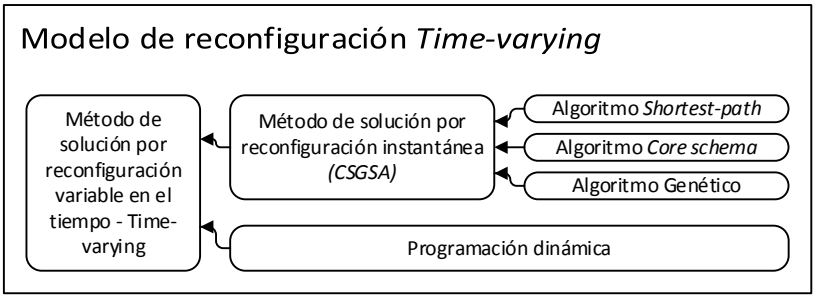


Figura 23 - Programación dinámica.

En el método propuesto, las subestaciones buscan de forma automática y constante la ruta óptima para minimizar las pérdidas de energía. Después hace una comparativa con otros algoritmos y métodos en función de tiempo de respuesta y optimización delos caminos buscados.

Los criterios para minimizar las pérdidas de energía son las *pérdidas de potencia – Power Losses* (KW) y el *coste operacional – Operation Cost (Yuanes)*

Por otro lado, contempla que no es lo mismo el comportamiento dependiendo del tipo de cargas en la red que consideremos, residencial, industrial/comercial media o industrial/comercial grande.

7.1.4.3 ACO-SA

En este artículo se presenta un algoritmo hibrido optimizado evolucionado basado en la combinación de dos de ellos, el *Optimización de Colonia de Hormigas - Ant Colony Optimization (ACO)* y *Recocido Simulado - Simulated Annealing (SA*) [29] para la *reconfiguración de la alimentación distribuida - distribution feeder reconfiguration (DFR),* considerando la *Generación Distribuida - Distributed Generators (DGs).*

Además, introduce un método de compensación basado en los costes de generación para los propietarios que tiene en cuenta la potencia activa y reactiva de la generación.

Los dos protocolos forman parte de un amplio elenco de algoritmos matemáticos que permiten en función de una serie de variables, encontrar el camino óptimo.

Actualmente, la generación distribuida cumple un papel determinante en los sistemas de generación distribuida. El algoritmo híbrido considerado se ha evaluado y demostrado como

el más eficaz considerando diferentes perfiles y variación de las cargas en el tiempo.

Según el artículo, el algoritmo minimiza la desviación del voltaje, el número de conmutaciones, el espacio en memoria es reducido, el tiempo de computación es bajo y la solución, en comparación con otros, es mejor en comparación con otros según varias de las pruebas efectuadas.

7.1.4.4 PSO

Particle Swarm Optimization (PSO) optimización por nube de partículas u optimización por enjambre de partículas hace referencia a una serie de métodos y algoritmos de optimización heurísticos que evocan el comportamiento de los enjambres de abejas en la naturaleza.

En este estudio es fundamental el tratamiento de los *Distributed Generators (DGs)* como parte fundamental en el tratamiento de las variables de la red de distribución.

En ese artículo se asume que el valor en los trasformadores (nodos) cambian en función de las siguientes variables: restricciones de la potencia activa de los *DGs*, los límites en las líneas de distribución, el aprovechamiento de los transformadores, las ecuaciones de flujo de potencia de las trifásica no balanceadas, los tiempos de operación diarios máximos admisibles de los transformadores, tiempos de operación diarios máximos admisibles de los condensadores y el factor de potencia de la subestación.

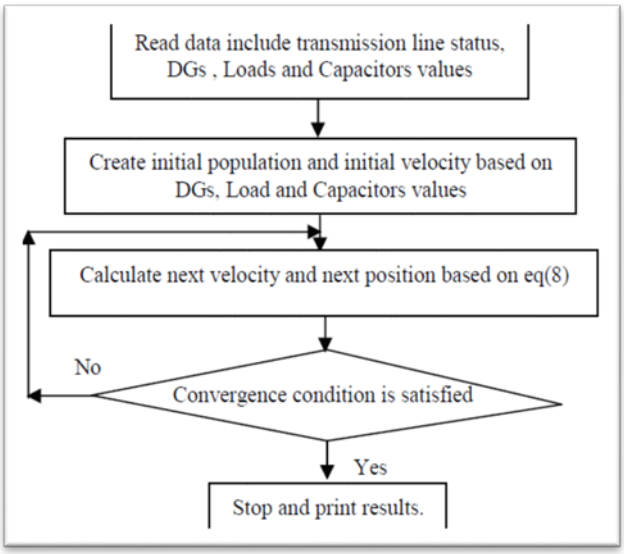


Figura 24 - Diagrama de flujo de PSO.

El funcionamiento es el siguiente, una partícula, como cualquier objeto vivo, tiene una memoria en la cual retiene la mejor experiencia. En esta técnica, cada solución candidata es

asociada con un vector de velocidad que es ajustado en función de la experiencia de cada partícula y de las experiencias de las partículas de sus "compañeros". De esta forma, las mejores experiencias de los grupos son siempre compartidas con todas las partículas y por ello, se espera que todas las partículas avancen hacia las áreas con mejor solución. El diagrama de flujo es el indicado en la *Figura 27*.

Finalmente hace una comparativa con otro algoritmo, llegando a la conclusión que tanto los resultados como las iteraciones necesarias son mejores que con el algoritmo *GA*.

7.2 Descripción del diseño

En nuestra investigación vamos a definir la configuración del protocolo **OSPF** para adaptarlo a la reconfiguración de una red inteligente de CTs, conociendo que la aplicación ejecutable y los variables que maneja, se encontrarían almacenados en el repositorio de datos del *PGDIN*.

En primer lugar vamos a justificar la utilización del protocolo **OSPF**[16] y una configuración adecuada según unos criterios que desglosaremos. Debido a la propia naturaleza del protocolo **OSPF**, que fue diseñado para la transmisión y enrutamiento de datos en la capa de red IP, necesitamos esbozar un nuevo elemento que complemente su arquitectura, el interfaz **iOSPF-SG**, que nos permitirá combinar un interfaz de datos con la conexión de la línea de transmisión.

A continuación, definiremos la arquitectura propiamente dicha de la red de distribución inteligente que nos permita visualizar claramente el comportamiento del protocolo **OSPF** implementado. Hemos supuesto una red mallada de tamaño medio, ya que las de mayor tamaño introducirían una compleja interpretación de los resultados y una de menor tamaño, no nos permitiría apreciar los detalles[17]. Por último, efectuaremos la simulación en un entorno de MATLAB.

No es objeto de este proyecto profundizar en la casuística específica de la reconfiguración como proceso que genera un transitorio en una red real, incorporando una serie de condicionantes importantes.

[16] Existen un gran número de implementaciones del protocolo **OSPF** en C++, Java, etc.
[17] Para facilitar una mejor comprensión hemos intentado que tanto la presentación como los cálculos sean muy visuales por lo que utilizaremos gran cantidad de gráficos y figuras.

7.2.1 Elección OSPF

En el mundo de Internet, la modificación de las redes de transmisión de datos para encaminar (routing) los paquetes por unas rutas óptimas, es una necesidad que ha sido ampliamente estudiada y resuelta, por lo que entendimos que alguna de estas tecnologías podría ser de utilidad en los desarrollos de reconfiguración de las redes eléctricas inteligentes que se están llevando a cabo actualmente.

Nuestra elección se basó en que tenía que ser un algoritmo que fuera robusto en sus cálculos, sencillo, ampliamente implantado/probado y con una experiencia contrastada. En este ámbito, encontramos que los protocolos más ampliamente implantados desde hace tiempo son estos dos: RIP y **OSPF**. Existen otros protocolos BGP, IGP, IGRP, etc., pero son más novedosos y complejos, e inicialmente los descartamos de nuestro planteamiento.

En la *Tabla 2* hacemos una comparativa de RIP y **OSPF**, hemos marcado en negrita las características que hemos definido como necesarias en nuestro ámbito de aplicación.

*Tabla 3 - Comparativa RIP y **OSPF***

RIP	**OSPF**
Vector de distancia – *Distance Vector*	Estado del enlace – *Link State*
Convergencia lenta	***Convergencia rápida***
Vista local de la red (Susceptible a bucles)	***Vista global de la red*** (área)
Administración sencilla	Administración compleja
Requisitos computacionales y de almacenamiento menos exigentes	Mayores requisitos computacionales y de almacenamiento
Consume más ancho de banda	***Más eficiente en ancho de banda***
Intercambio mensajes SOLO entre vecinos	***Mensajes enviados para N nodos*** y E enlaces de la res; O(N*E).
Tiempo de convergencia, variable, puede haber bucles. Problema de cuenta a infinito	***Tiempo de convergencia determinado***, complejidad de cómputo $O(N^2)$ y puede oscilar
Un nodo puede anunciar una ruta de coste erróneo, cada nodo vecino propaga ese error y alcance a toda la red potencialmente	Un nodo puede anunciar un *LSA* incorrecto: error, sabotaje, etc., pero ***cada nodo calcula sus propias rutas*** (alcance limitado)

En la comparativa se aprecian fácilmente la conveniencia de utilizar **OSPF**, no obstante, de todas ellas vamos a destacar éstas:

- Tiempo de convergencia rápido y determinado. Es muy importante que en caso de modificarse la red por cualquier motivo, la reconfiguración de todos los nodos se haga rápidamente y no exista la posibilidad de que no concluya y de que los cálculos

se alarguen incluso hasta el infinito.

- Robustez, cada nodo calcula sus propias rutas. Esto proporciona robustez, porque en caso de un fallo, el resto de nodos funcionarán correctamente.
- Eficiencia de ancho de banda. Interesa que la información que intercambien los nodos sea pequeña y que pueda dedicarse a otras cuestiones como el envío de datos de consumo de contadores, mediciones de la red, etc.

El corazón del **OSPF** es el algoritmo de *Dijkstra* . Se trata de un algoritmo de búsqueda gráfica usado para buscar el camino más corto desde un nodo dado a otros nodos de la red, dado un criterio de búsqueda utilizando siempre valores positivos. El coste de una ruta será la suma de los costes de los enlaces que atraviesa: coste_ruta (uxyz) = c(u,x)+c(x,y)+c(y,z) = 1+1+2 = 4

7.2.1.1 Configuración OSPF

Hemos implementado y adaptado las principales funciones y características, pero la norma permite incluir otras posibilidades, pero no son de utilidad para nosotros. Para adaptar el protocolo a nuestras necesidades, proponemos introducir una serie de cambios en algunos de sus variables configurables, el criterio seguido consiste en que el tiempo de respuesta del protocolo **OSPF** en nuestra red de CTs debe ser inferior al que utiliza para recalcular la caída en una red de routers de paquetes IP.

Nuestro planteamiento de configuración será la de "*point to point*", por cuestiones de robustez. El uso de **OSPF** en clase subred de difusión, con *Designated Router*, haría que su funcionamiento fuera más eficiente pero menos robusto, nosotros necesitamos que nuestra red sea sobre todo robusta.

Como parte de la configuración y modificaciones del *MIB* a incluir como configuración básica, es necesario asignar el nombre al nodo **OSPF** (router o CT) que se identifique correctamente[18], las direcciones IP de cada uno de sus interfaces y cómo no, el coste de los interfaces.

La configuración de **OSPF** permite modificar todas las variables de tiempos que afectan sobre todo al proceso de intercambio de mensajes entre nodos. Nuestras propuestas están dirigidas a que los nodos descubran lo antes posible la caída de un nodo o enlace en la red,

[18] En su uso como router este valor suele ser la dirección IP de uno de sus interfaces.

esto provocará un mayor tráfico en los canales de comunicaciones pero nos asegura una rápida respuesta ante eventos inesperados.

De esta forma las modificaciones en la temporización, que serán necesariamente iguales en todos los nodos del segmento de la red, quedan propuestas de acuerdo con la *Tabla 4*.

*Tabla 4 - Variables de tiempo **OSPF** modificadas.*

Variable	Valor por defecto (seg)	Descripción	Valor Red CTs
MinLSArrival	1	Cadencia máxima con la que un router puede recibir actualizaciones vía LSA mediante "inundación".	-
MinLSInterval	5	Cadencia máxima con la que un router puede actualizar un *LSA*.	-
CheckAge	300 *5 minutos*	Cadencia con la que un router verifica el *checksum* contenido en un *LSA* contenido en su base de datos.	180 *3 minutos*
MaxAgeDiff	900 *15 minutos*	Tiempo por el que dos instancias *LSA* pueden considerarse diferentes.	300 *3 minutos*
LSRefreshTime	1800 *30 minutos*	Tiempo en el que un router debe refrescar cualquier mensaje *LSA* originado por él.	900 *15 minutos*
MaxAge	**3600** ***1 hora***	**Edad de un *LSA*, cuando alcanza este valor sin ser renovado, se elimina.**	**1800** ***30 minutos***
***OSPF**IfTransitDelay*	1	Tiempo que se tarda en enviar un mensaje *LSA* a los vecinos.	-
***OSPF**IfRetransInterval*	5	Es el tiempo de espera de *ACK* tras mensaje *LSA*	3
OSPFIfHellolnterval	**10**	**Periodicidad de envío de mensajes *Hello*.**	**5**
OSPFIfRtrDeadlnterval	**40**	**Tiempo de espera de un mensaje *Hello* de un vecino.**	**20**
***OSPF**lfPollInterval*	120	En subredes *NMBA*, es la periodicidad con la que se envía un mensaje para saber si el vecino se ha despertado	-

Los valores de los tiempos modificados de **OSPF**, como criterio los hemos reducido a la mitad, son usados para afinar el descubrimiento de vecinos, los procedimientos de mantenimiento y los procesos de "inundación".

Recordamos que los mensajes *Hello* sirven para identificar si uno de los nodos está sin servicio. Cuando un vecino no envía paquetes *Hello* pasado el intervalo ***OSPF**IfRtrDeadlnterval*, entonces se interpreta que el vecino no está operativo, como consecuencia de ello, este valor siempre debe ser mayor que ***OSPF**IfHellolnterval* , y unas cuantas veces superior para evitar que la pérdida de dos mensajes *Hello* consecutivos puedan declarar inoperacional al vecino.

7.2.1.2 Criterio de aplicación de valores a los enlaces.

Lo primero de todo, es recordar que **OSPF** hace una asignación estática del valor de los enlaces, no dinámica. Esta condición es importante, porque no será posible cambiar el valor de una enlace una vez adjudicado un valor, simplemente indicar si está o no disponible, para cambiarlo hay que reiniciar la aplicación y cambiar la configuración básica de **OSPF**, la explicación la tenemos en *7.4.8 Cálculo de la ruta*.

A partir de esta restricción y conociendo que la reconfiguración en redes de distribución es esencialmente un problema de optimización y minimización de las pérdidas totales de una red, hemos planteado un modelo de cálculo de la métrica aplicable a nuestra red basado en las pérdidas de los enlaces.

Decidimos proponer para nuestra métrica una función significativa como son las pérdidas reales en los enlaces los cual cumple con el criterio fundamental de asignación en nuestro protocolo: cuanto menor sea el valor del enlace (métrica), mejor es su comportamiento y por lo tanto será la primera opción seleccionada cuando se efectúen los cálculos. Este valor puede ser fácilmente calculado por cada uno de los Agentes utilizando los parámetros que están monitorizando constantemente de tensión y corriente en la línea, con lo que es relativamente sencillo hacer una estimación de las pérdidas por cada uno de los nodos, a partir de los consumos de potencia activa y los datos de potencia reactiva.

De esta forma nuestra función, será muy sencilla:

$$M_1^n = K\ {P_{loss}}_1^{\ n}$$

Siendo:

M = Métrica de cada uno de los enlaces, valores asignados.

P_{loss} = Pérdida de potencia calculada en cada uno de los enlaces.

K = Constante que haga del dato de la pérdida un número entero entre 0 y 65535.

En nuestros cálculos no hemos considerado otros límites usados frecuentemente, como son la caída de tensión máxima admisible en la línea, los límites de corriente en la línea, los límites de la capacidad del transformador y otras restricciones operacionales de la red.

7.2.1.3 Otras funciones de cálculo de la métrica

Otros trabajos estudian la fiabilidad de las redes [30] y contemplan parámetros ampliamente usados como la Energía Esperada no Suministrada *(Expected Energy Not Supplied - EENS),*

Coste de Fallo Esperado *(Expected Outage Cost - ECOST)* o Índice de la Duración de la Interrupción Media del Sistema *(System Average Interruption Duration Index - SAIDI)*.

Sin embargo, otros combinan las probabilidades estadísticas de fallos con las pérdidas buscando una aproximación más exacta a la cuestión [31], pero que también introduce una mayor complejidad en el caso de una implementación real en una red como la nuestra con inteligencia distribuida en cada uno de los nodos.

Asimismo, en otro de los casos estudiados [27], el coste total de la planificación del sistema distribuido objeto de estudio es la suma de tres variables: el coste fijo anual incluido el coste de las líneas (ramas) y de las subestaciones (nodos), el coste por las pérdidas de energía y el coste de interrupción del servicio.

7.2.1.4 Interfaz virtual iOSPF-SG

Los parámetros y el estado de las líneas de transmisión deben de ser tratadas como fuentes de información (datos) por **OSPF** pero sin modificar la arquitectura y la implementación del protocolo **OSPF**. Para resolver este dilema, proponemos el desarrollo de una sencilla API que gestione esa información que obtiene del Agente, y que hemos denominado interfaz OSPG para Smart Grids, **iOSPF-SG**, el cual hemos esbozado en la *Figura 28*.

Los casos por los que puede cambiar de estado cualquiera de los enlaces son:

1. Falta en la línea de transmisión eléctrica.
2. Cambios importantes en las características de la red, pérdidas, cargas o requerimientos del servicio.
3. Errores o caída línea de datos
4. Caída de un nodo que afecta a la transmisión de datos o a la parte eléctrica.

Los casos 3 y 4 son detectados mediante el protocolo *Hello*, sin embargo el 1 y 2 solamente pueden tratarse a partir de la información de la que dispone el agente.

Una operativa de funcionamiento que debería incluir este interfaz virtual, consiste en prohibir la recepción de paquetes de control enviados por el enlace de datos asociado a la línea de transmisión que sufriera los casos 1 o 2 (datos proporcionados por el Agente), simulando de esta forma lo que ocurriría por la ausencia de mensajes *Hello* en ese enlace.

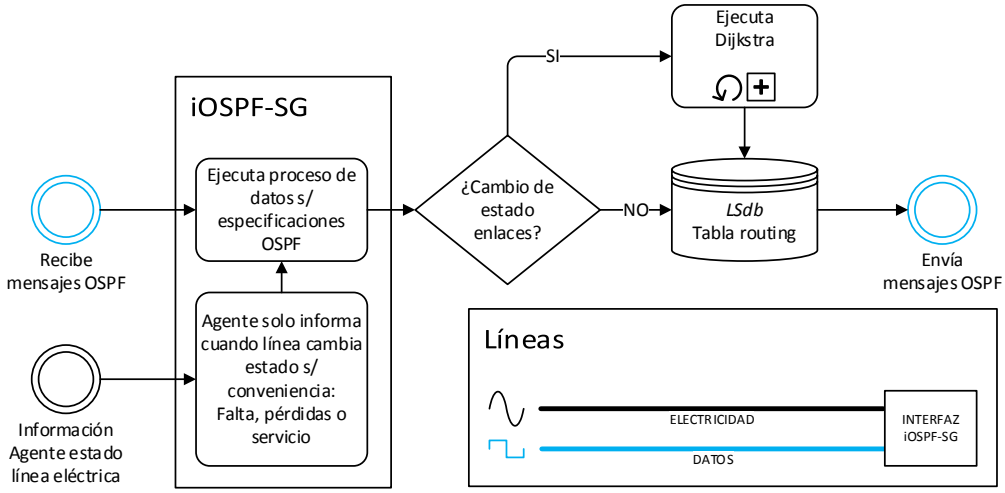


*Figura 25 - Planteamiento esquema i**OSPF**-SG*

7.2.2 Red de distribución

Nuestra red está formada por enlaces punto a punto, lo que nos permite la implementación más sencilla y robusta de OPSF.

Hemos supuesto para nuestro planteamiento que los Centros de Transformación cuentan con un Agente virtual y dispositivos inteligentes que permiten medir y controlar sus componentes (ver *Figura 29*): conmutadores, disruptores, transformador y/o protecciones. En función de las necesidades del servicio el Agente puede conmutar su alimentación a cualquiera de las líneas. La línea de comunicación le permite compartir información con cualquier otro elemento de la red y sobre todo con sus pares.

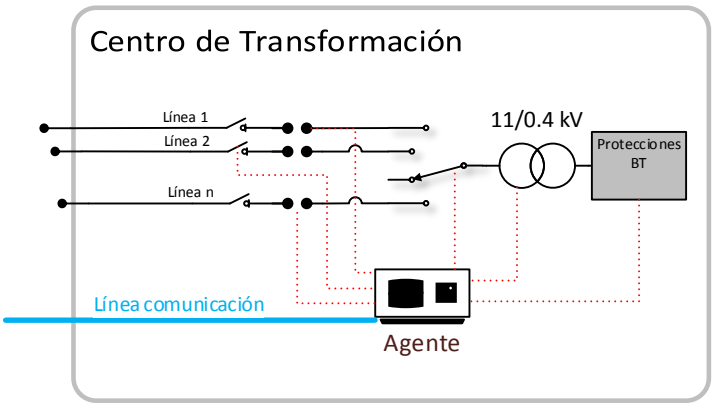


Figura 26 - Esquema Centro de Transformación Inteligente

En cuanto a las líneas de comunicaciones nos es indiferente la tecnología de transmisión, radio, Ethernet, *FDDI*, pero suponemos que todas las transmisiones se establecen como una red IP. En cuanto al ancho de banda, suponemos que las comunicaciones entre *CTs* no se van a ver afectadas por el intercambio de mensajes del protocolo *Hello* de **OSPF**, ya que el

tamaño de los mensajes intercambiados es mínimo.

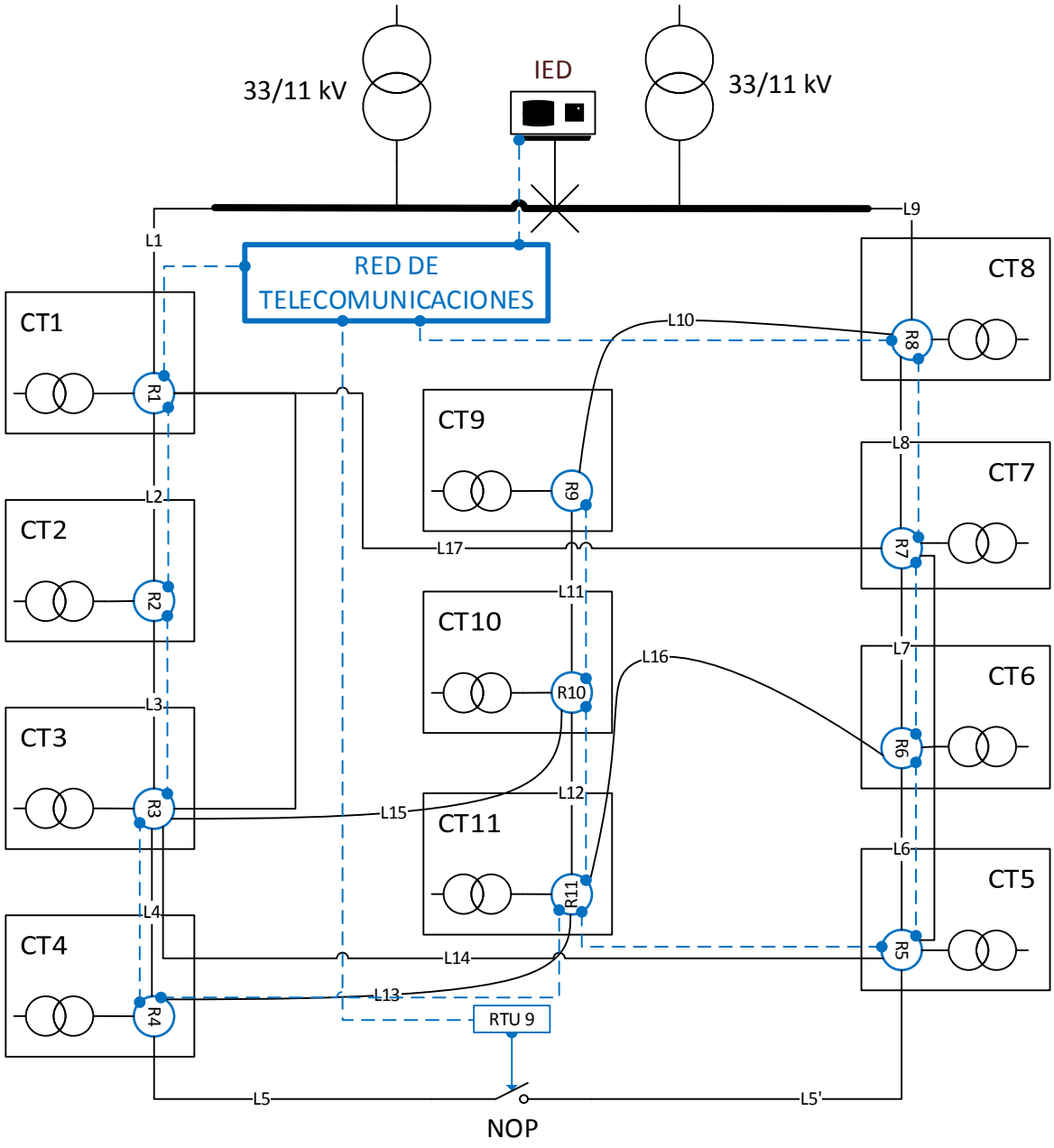


Figura 27 - Esquema eléctrico de la red mallada de CTs

La Red de Distribución Mallada objeto de estudio representada en *Figura 30*, está formada por 11 Centros de Transformación Inteligentes, con las mismas características de diseño y con un Agente tipo PGDIN implementado que ejecuta nuestro protocolo **OSPF**. La red de distribución cuenta con estas características:

- Urbana, situada en el Corredor del Henares. Una red urbana tiene posibilidades reales de contar con enlaces adicionales y convertirse en una red mallada como la nuestra, en una red rural por su extensión no suele ser lo habitual.

- Inteligente. Esto es, los PGDIN disponen de información en tiempo real de los valores de tensión, corriente y potencia de las líneas de transmisión.
- Identificamos para cada uno de los nodos de la red un Centro de Transformación.
- Cada una de las líneas de transmisión cuenta con unos valores de carga que definirán las pérdidas de estos enlaces entre nodos de la red.

Para simplificar la red mallada anterior hemos obviado la representación en el esquema las cargas, los puntos de conexión, las comunicaciones, los buses y conexiones y hemos identificado cada una de las rutas. De esta forma obtenemos la representación gráfica de la *Figura 31*.

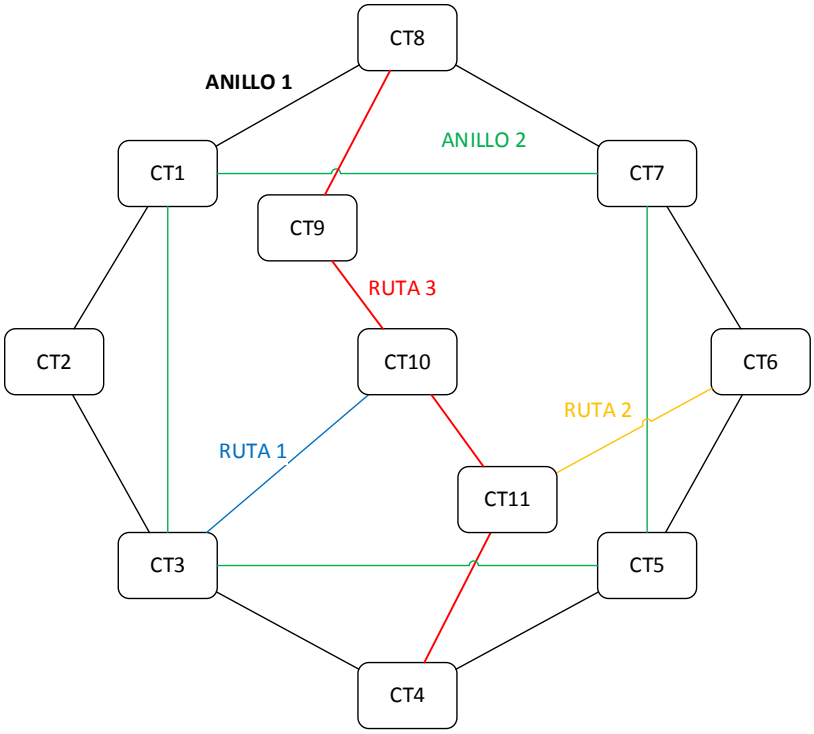


Figura 28 - Nodos y enlaces de la red de distribución propuesta.

Los datos de las métricas deberían ser generados de acuerdo con los criterios definidos en el apartado *8.3.2 Criterio de aplicación de valores a los enlaces.* Sin embargo, al no disponer de una red real con cargas reales a partir de la cual podríamos generar los valores de las pérdidas en los enlaces y no siendo ese cálculo objeto de este estudio, hemos tomado valores generados de forma aleatoria, prescindiendo de cálculos reales de los circuitos o de las cargas.

No obstante, como ya hemos mencionado, el criterio podría ser modificado para adaptarlo a

las necesidades reales. En este esquema de conexión los valores (métricas) de las conexiones hemos supuesto las siguientes:

- Anillo 1. Conecta todos los CTs de la red, del 1 al 8. Los valores de sus métricas son los indicados en la tabla *Tabla 4*.

Tabla 5 - Métrica de enlaces en Anillo1

CT	1	2	3	4	5	6	7	8
1		1818						9106
2	1818		4018					
3		4018		3377				
4			3377		5752			
5				5752		2348		
6					2348		4868	
7						4868		3507
8	9106						3507	

- Anillo 2. Conecta los CTs interiores, en concreto los 1, 3, 5 y 7. Los valores de sus métricas son los indicados en la tabla *Tabla 5*.

Tabla 6 - Métrica de enlaces en Anillo2

CT	1	3	5	7
1		844		3998
3	844		1233	
5		1233		7757
7	3998		7757	

- Ruta 1. Conecta el CT 3 con el 10.

Tabla 7 - Métrica de la Ruta 1

CT	3	10
3		9027
10	9027	

- Ruta 2. Conecta el CT 6 con el 11.

Tabla 8 - Métrica de la Ruta 2

CT	6	11
6		3685
11	3685	

- Ruta 3. Conecta secuencialmente el CT8-CT9-CT10-CT11 y CT4

Tabla 9 - Métrica de la Ruta 3

CT	4	8	9	10	11
4					2217
8			8759		
9		8759		2581	
10			2581		855
11	2217			855	

7.3 Cálculos efectuados

Nuestro ensayo parte de una situación de reposo en la que la *LSdb* (base de datos de encaminamiento) del protocolo **OSPF** instalado en cada uno de los Agentes de los Centros de Transformación de la red de distribución propuesta, ver *Figura 28*, la *LSdb* es distribuida y está replicada en todos los miembros, todos han llegado a converger.

En nuestra demostración, en vez de hacer los cálculos para todos los CTs del segmento, hemos elegido el CT1 como ejemplo de funcionamiento de esta red, sabiendo que todos los demás, por compartir la *LSdb* a partir de la cual se generan las rutas, obtendrían unos resultados similares, pero personalizados con respecto a su ubicación en ella.

7.3.1 Tratamiento de mensajes OSPF

La verdadera innovación de nuestro planteamiento es el intercambio de información entre nodos de la red para conocer el estado real de los nodos y enlaces antes de que cambien.

En este estado la *LSdb* de todos los CTs es igual, cuando se produce una modificación, y hasta que se completa la convergencia, pueden ser diferentes algunos registros pero la mayor parte de la información contenida se mantiene invariable.

En detalle, cuando se produce un cambio en alguno de los enlaces, la secuencia de los acontecimientos es la siguiente:

1. Se hacen modificaciones en los *LSA*.
2. Se empaqueta en un *Link State Update*, que puede o no contener otros *LSAs*.
3. Se envía por todos sus interfaces produciendo la *inundación*.
4. Cuando uno de sus vecinos recibe el paquete *Link State Update*, examina cada uno de los *LSA* contenidos en la actualización. Suponiendo el *LSA* válido y comprobando que es más reciente que la última copia de la que dispone, instala el *LSA* nuevo en la

LSdb propia.

5. Envía un *LSACK* de vuelta al router que lo envió.
6. El receptor reencapsula el *LSA* en un nuevo paquete *Link State Update* y lo envía por todos sus interfaces, excepto por el que se lo envió.

Este procedimiento se repite hasta que todos los routers del dominio han actualizado el *LSA*. Mientras que no haya modificaciones en los enlaces de la red, los CTs únicamente envían mensajes *Hello* para verificar que todo funciona correctamente.

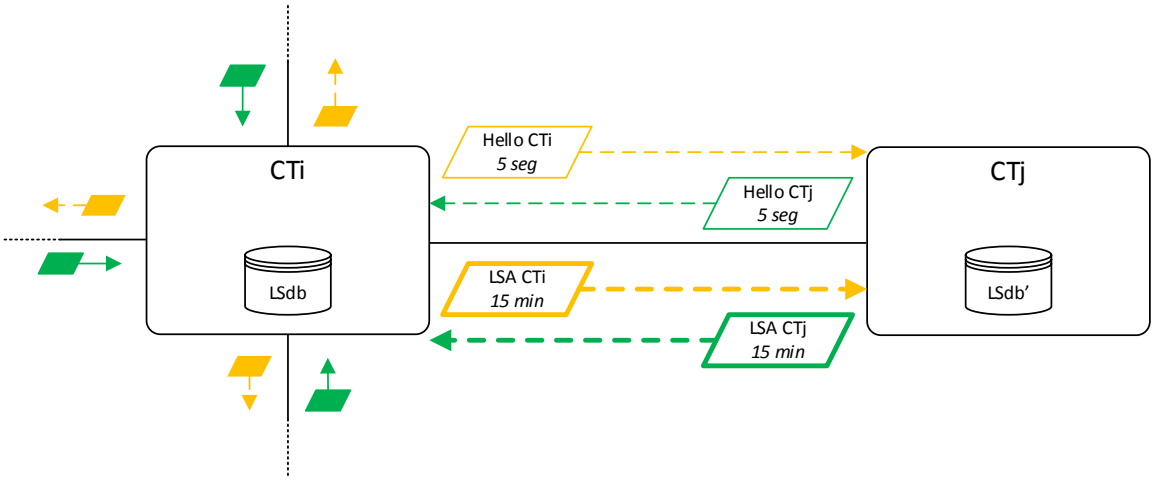


Figura 29 - Intercambio de mensajes entre CTs

En el caso de que un vecino CT aparezca de nuevo en la red, es decir que no tenga ninguna base de datos previa, solo en ese caso se realiza un *Database Exchange*: se envía una secuencia completa de paquetes *Database Description* al vecino, se recibe la secuencia completa de paquetes *Database Description* del vecino, y después de tener todos sus paquetes *Link State Request* contestados por *Link State Updates* del vecino; el router declara que la conexión se ha sincronizado y advierte que ya está dispuesto a cursar tráfico de datos.

7.3.2 Modelo estado inicial

Para generar la *LSdb* que contiene los valores de los enlaces, definimos la matriz de la *Tabla 3* con la métrica de los enlaces (líneas de transmisión) que hemos detallado a partir de la red definida en la *Figura 28*. De acuerdo con nuestros criterios, aquellos enlaces que tienen un mayor valor, tienen un peor comportamiento, sus pérdidas son mayores.

Tabla 10 - Matriz de valores inicial

CT	1	2	3	4	5	6	7	8	9	10	11
1	1	1818	844	Inf	Inf	Inf	3998	9106	Inf	Inf	Inf
2	1818	1	4018	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
3	844	4018	1	3377	1233	Inf	Inf	Inf	Inf	9027	Inf
4	Inf	Inf	3377	1	5752	Inf	Inf	Inf	Inf	Inf	2217
5	Inf	Inf	1233	5752	1	2348	7757	Inf	Inf	Inf	Inf
6	Inf	Inf	Inf	Inf	2348	1	4868	Inf	Inf	Inf	3685
7	3998	Inf	Inf	Inf	7757	4868	1	3507	Inf	Inf	Inf
8	9106	Inf	Inf	Inf	Inf	Inf	3507	1	8759	Inf	Inf
9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	8759	1	2581	Inf
10	Inf	Inf	9027	Inf	Inf	Inf	Inf	Inf	2581	1	855
11	Inf	Inf	Inf	2217	Inf	3685	Inf	Inf	Inf	855	1

De la matriz es necesario destacar lo siguiente:

- Es una matriz simétrica, lo que quiere decir que la métrica de cada uno de los enlaces es igual vista desde un extremo o desde el otro. Esto es una conveniencia, ya que en la realidad, las pérdidas del enlace de *CTi* a *CTj*, no son iguales a las de *CTj* a *CTi*, además, **OSPF** está configurado para permitir asignar valores diferentes a cada uno de los extremos de un enlace.
- El coste de los enlaces de cada CT consigo mismo es 1.
- El coste de los enlaces que no existen es Infinito (Inf), un circuito abierto.

Representamos[19] los Centros de Transformación (nodos) generados a partir de la *Tabla 11*, definiendo las posiciones en un eje de coordenadas cada uno de los nodos.

Tabla 11 - Posición en cuadrante.

pixeles	**CT1**	**CT2**	**CT3**	**CT4**	**CT5**	**CT6**	**CT7**	**CT8**	**CT9**	**CT10**	**CT11**
Posición X	300	100	300	500	700	900	700	500	450	500	550
Posición Y	800	500	200	100	200	500	800	900	700	500	300

A partir de esta matriz generamos la representación gráfica con la secuencia de la *Figura 33* y definiendo el esquema final según la *Figura 34*.

[19] MATLAB, gráfica de 1000x1000 pixels

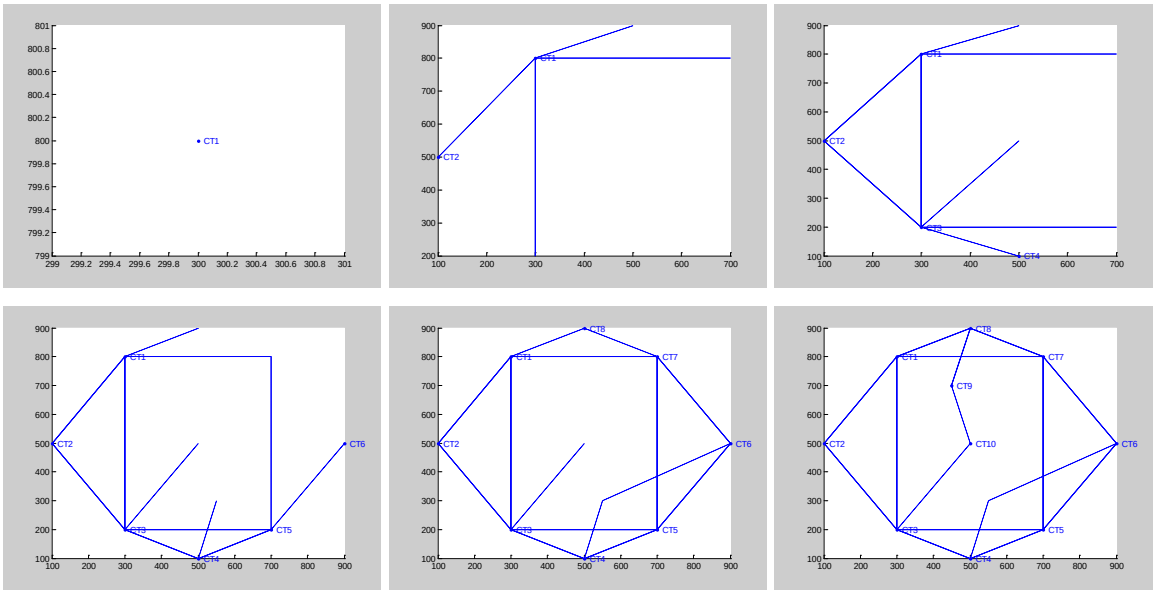


Figura 30 - Proceso de representación gráfica situación inicial

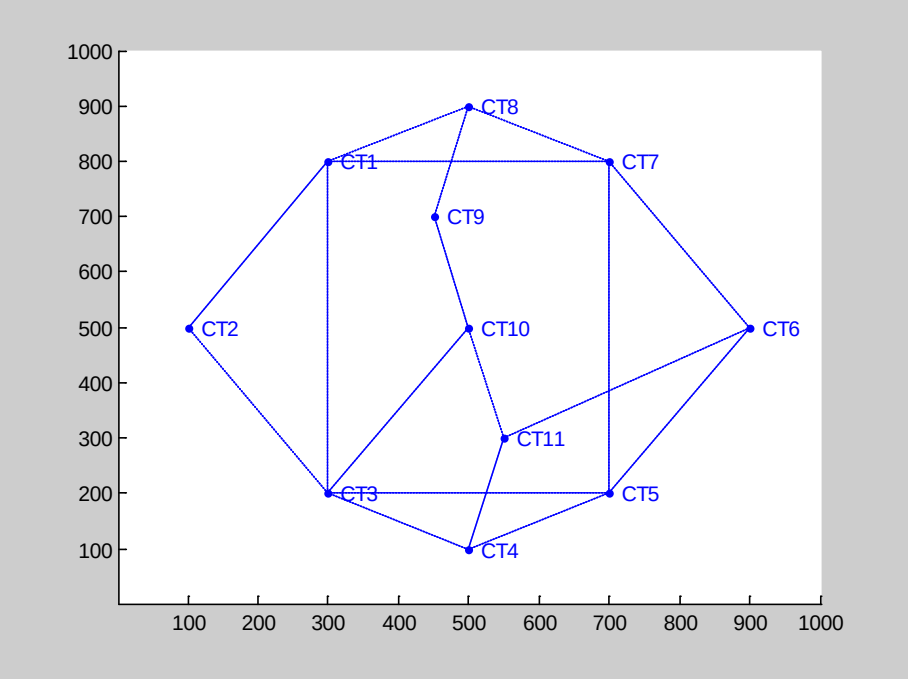


Figura 31 - Gráfico de la red completa situación inicial

7.3.2.1 Cálculo tabla de enrutamiento en CT1

Una vez disponemos de nuestra red y de la matriz de enlaces, aplicamos el algoritmo de *Dijkstra*[20], Cada rama desde un nodo cualquiera tiene un "valor positivo" asociado, en su funcionamiento, el algoritmo asigna a cada nodo una etiqueta temporal y permanente. Inicialmente todos los nodos, excepto el nodo raíz, son asignados con etiquetas temporales, una vez seleccionada la ruta, la etiqueta se hace permanente hasta el siguiente cambio de

[20] http://www.mathworks.com/matlabcentral/fileexchange/5550-dijkstra-shortest-path-routing

estado (hasta el siguiente cálculo).

Desde el CT1 conseguimos los mejores caminos a todos los CTs restantes, representados gráficamente en la *Figura 31* y habiendo generado la tabla de encaminamiento *Tabla 4* que queda almacenada en el CT1 como parte del ***OSPF***, hasta que no se produce un cambio de estado de alguno de los nodos o enlaces, esto es, que la *LSdb* permanece sin cambios.

*Tabla 12 - Tabla de encaminamiento **OSPF** en CT1 situación inicial*

	CT1	CT2	CT3	CT4	CT5	CT6	CT7	CT8	CT9	CT10	CT11
Ruta	-	1 2	1 3	1 3 4	1 3 5	1 3 5 6	1 7	1 7 8	1 3 4 11 10 9	1 3 4 11 10	1 3 4 11
Coste	0	1818	844	4222	2078	4425	3998	7505	9875	7294	6439

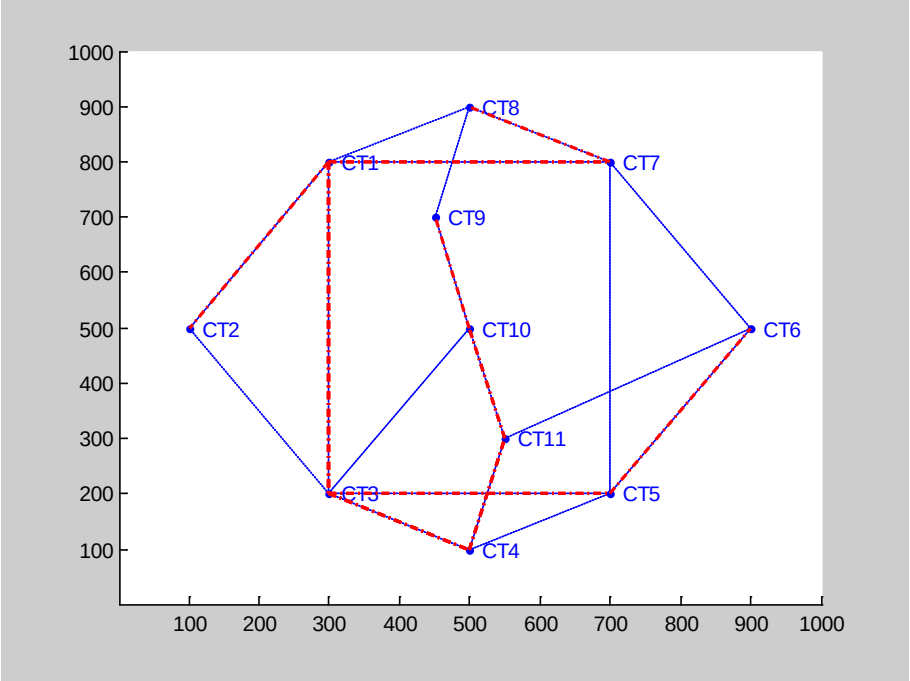


*Figura 32 - Representación rutas **OSPF** en CT1 situación inicial*

A la vista de los resultados se aprecia claramente que la ruta más costosa desde el CT1 es la que llega al CT9, y además el camino recorrido es mayor. Sin embargo, es interesante comparar la ruta al CT8 y al CT10, la primera pasa por menos nodos pero tiene un coste mayor que la segunda.

7.3.3 Modelo pasa a inestable.

De forma repentina, en nuestro modelo estable se produce alguna de las situaciones que hemos definido como posibles detonadores de una necesidad de reconfiguración en la red, falta en una de las líneas de transmisión, fallo de uno de los CTs o de las líneas de

comunicaciones[21], requerimientos del servicio, optimización de transmisión, etc.

Una vez se produce, los *CTs* tienen conocimiento de ello a través de sus interfaces i**OSPF**-SG, los cuales obtienen esa información de los mensajes *Hello* (o de su ausencia) y de la información que obtienen del Agente, como hemos mostrado anteriormente.

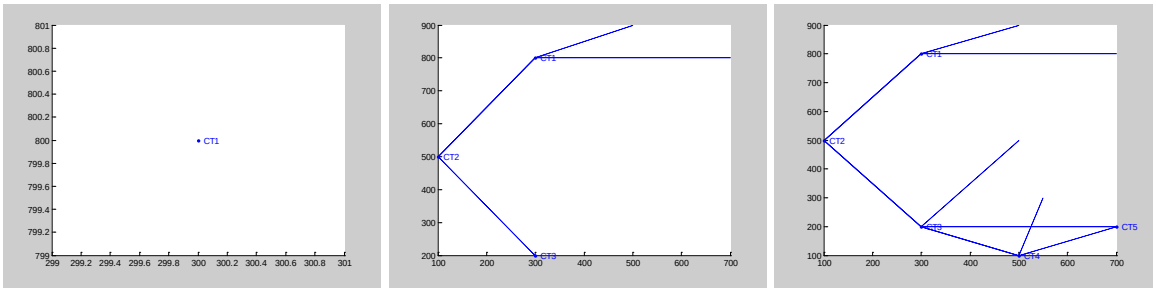
En nuestro cálculo suponemos un fallo en el enlace del CT1 con el CT3, lo que hace que todos los CTs de la red, empezando por aquellos que están conectados a ese enlace, se intercambien la información hasta converger en una nueva *LDdb* común. La matriz *Tabla 5* es la resultante después del intercambio de *LSAs* entre los diferentes nodos, destacando que la métrica del enlace CT1 - CT3 pase a Inf (Infinito), circuito abierto.

Tabla 13 - Matriz de valores tras la falta

CT	1	2	3	4	5	6	7	8	9	10	11
1	1	1818	Inf	Inf	Inf	Inf	3998	9106	Inf	Inf	Inf
2	1818	1	4018	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
3	Inf	4018	1	3377	1233	Inf	Inf	Inf	Inf	9027	Inf
4	Inf	Inf	3377	1	5752	Inf	Inf	Inf	Inf	Inf	2217
5	Inf	Inf	1233	5752	1	2348	7757	Inf	Inf	Inf	Inf
6	Inf	Inf	Inf	Inf	2348	1	4868	Inf	Inf	Inf	3685
7	3998	Inf	Inf	Inf	7757	4868	1	3507	Inf	Inf	Inf
8	9106	Inf	Inf	Inf	Inf	Inf	3507	1	8759	Inf	Inf
9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	8759	1	2581	Inf
10	Inf	Inf	9027	Inf	Inf	Inf	Inf	Inf	2581	1	855
11	Inf	Inf	Inf	2217	Inf	3685	Inf	Inf	Inf	855	1

7.3.3.1 Representación gráfica de la red en eje de coordenadas.

A partir de esta nueva matriz generamos la representación gráfica con la secuencia de la *Figura 32* y definiendo el esquema final según la Figura 33.



[21] Puede fallar la línea de comunicaciones pero no la de transmisión, sería un caso específico que habría que tratar.

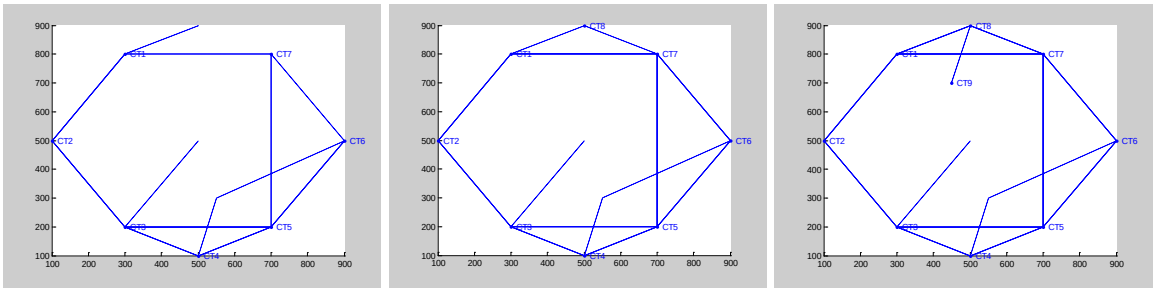


Figura 33 - Proceso de representación gráfica tras cambio de estado

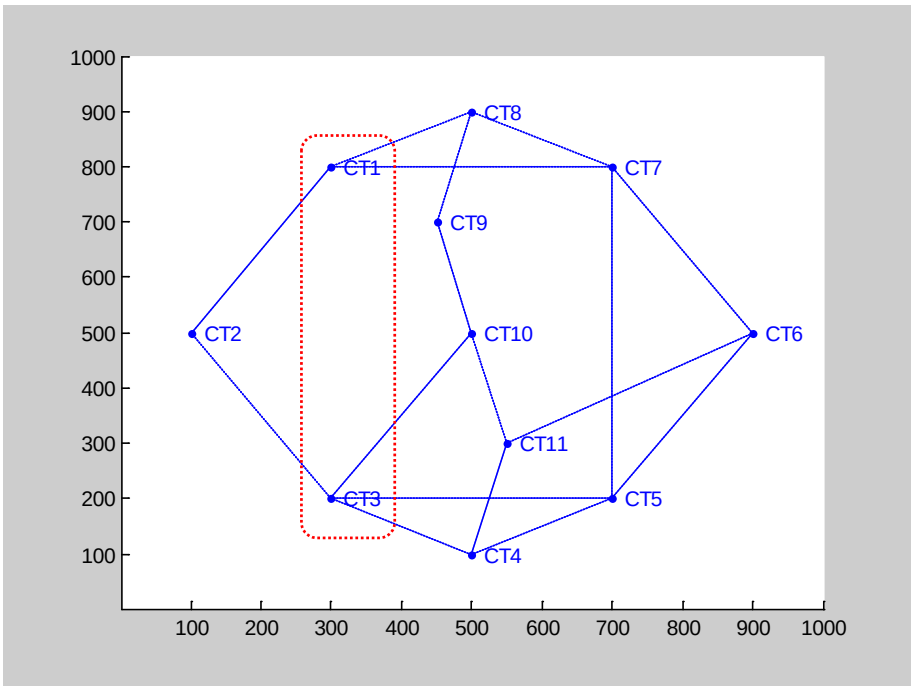


Figura 34 - Gráfico de la red completa tras cambio de estado

Hemos recuadrado el enlace ausente entre CT1 y CT2.

7.3.3.2 Cálculo nueva tabla de enrutamiento en CT1

Una vez disponemos de nuestra nueva red y de la matriz de enlaces, repetimos el proceso y conseguimos una nueva tabla de *Tabla 6*, representada gráficamente en la *Figura 33.* Hemos indicado en gris, las modificaciones en las rutas después del nuevo cálculo, como se puede apreciar todas ellas aumentan su coste.

*Tabla 14 - Tabla de encaminamiento **OSPF** en CT1 tras cambio de estado*

	CT1	CT2	CT3	CT4	CT5	CT6	CT7	CT8	CT9	CT10	CT11
Ruta	-	1 2	1 3	1 3 4	1 3 5	1 3 5 6	1 7	1 7 8	1 3 4 11 10 9	1 3 4 11 10	1 3 4 11
Coste	0	1818	5837	9214	7070	8866	3998	7505	14867	12286	11431

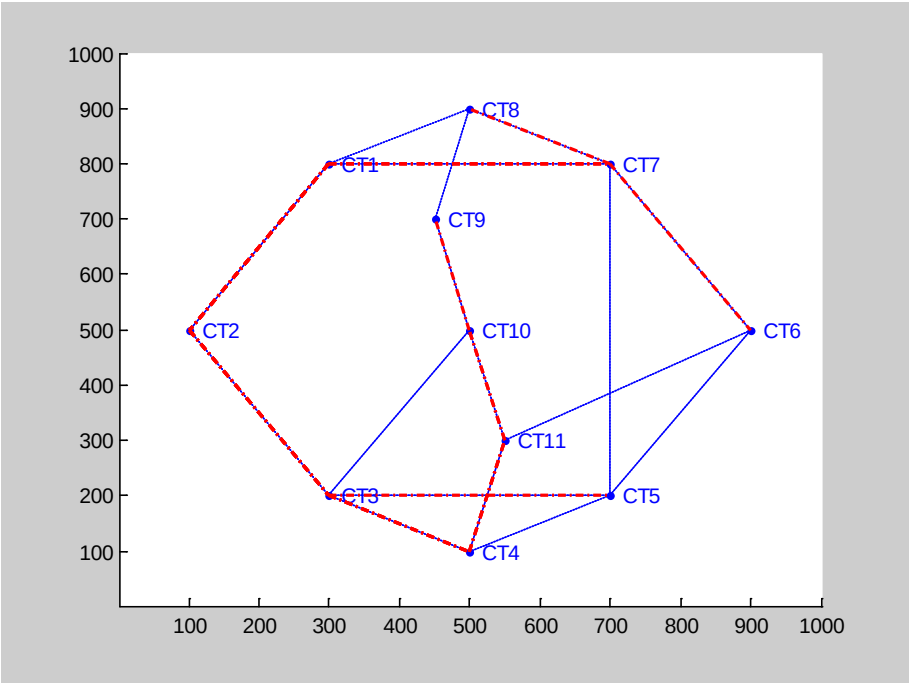


*Figura 35 - Representación rutas **OSPF** en CT1 tras cambio de estado*

De nuevo, hasta que no se produzca un cambio de estado esta *LSdb* permanecerá sin cambios.

Otra cuestión importante, que simplemente vamos a esbozar, es el coste máximo que pueden alcanzar las rutas y la utilidad derivada de ello. Recordamos que el mayor valor de todo el camino no puede exceder de 65.535[22].

Recordamos que cada uno de los CTs al compartir la misma *LSdb* generan la misma tabla de encaminamiento, por ejemplo, si obtuviéramos la tabla de encaminamiento del CT4, la ruta al CT9, sería igual (en pasos y coste) a la calculada por el CT1.

7.4 Conclusiones

El protocolo **OSPF** ha demostrado su utilidad como herramienta para la reconfiguración automática de una red de distribución inteligente, basada en el control de los CTs por los Agentes virtuales, compartiendo la gestión y toma de decisiones de forma descentralizada, los resultados son concluyentes. Todo el proceso es automático, sin la intervención de los operadores.

El **OSPF** no solo hace un cálculo robusto y rápido de la mejor ruta en función de las métricas de los enlaces, si no que mediante el intercambio de mensajes mantiene actualizada la base

[22] 2^{16}, tamaño del campo 16 bits.

de datos distribuida (común) de todos los enlaces de la red entre todos los Agentes.

Como hemos visto, existen muchos otros algoritmos y protocolos, pero a nuestro entender no aportan ni la robustez, ni la velocidad de cálculo que debe tener un sistema de reconfiguración automática de la red que sepa comportarse adecuadamente ante determinadas situaciones, no solo de faltas o interrupciones de las líneas de transmisión, sino también de la caída de las líneas de comunicaciones, de los propios CTs o simplemente por requerimientos del servicio.

En una red de mayor tamaño que la nuestra, el uso de un algoritmo más complejo (y más exacto) podría provocar que los retardos nunca permitan converger (estabilizar) la red con los riesgos que eso comporta.

Este Trabajo Fin de Grado es un primer paso que posteriormente permitirá profundizar en determinadas cuestiones que quizá no han quedado suficientemente resueltas, algunas de las cuales proponemos a continuación.

7.4.1 Trabajo futuro

Estas son algunas de las propuestas que durante la investigación y la posterior redacción de esta memoria a nuestro entender necesitarían un mayor análisis y estudio en detalle:

1. Implementación real del protocolo **OSPF** en un *PGDIN*.
    a. Estimación de los recursos consumidos por el protocolo **OSPF** ejecutándose en un PGDIN.
    b. Tiempos de respuesta ante un cambio de estado, incluyendo la recepción del mensaje de advertencia, el cálculo del nuevo árbol, y una vez conocida la nueva tabla, las actuaciones sobre los switches.
2. Profundizar en las características de i**OSPF**-SG y generar el código correspondiente.
3. Generar una función adecuada para asignar las métricas de los enlaces en función de los parámetros que mejor se ajusten a una red de distribución real.

8 CÓDIGO MATLAB

8.1 Función JMMCfix.m

Para nuestro TFG hemos desarrollado una función de MATLAB que hace un cálculo del Algoritmo Dijkstra para uno de los nodos seleccionados, disponiendo para ello de una serie de variables entre las que se definen el número de nodos de la red, los enlaces existentes entre cada uno de los nodos y el coste de los enlaces.

A continuación, hace una llamada a la función *dijkstra.m* previamente incluida en las librerías de MATLAB y una vez efectuados los cálculos genera el fichero ***OSPF**.mat* que contiene por un lado la tabla con las rutas desde el nodo elegido al resto de nodos (los lugares por donde pasa) y el coste de cada una de las rutas.

Para una mejor comprensión, la función efectúa previamente una representación gráfica de la red de nodos y enlaces en un eje de coordenadas y los caminos óptimos calculados también los representa.

8.1.1 Código

```
function [tRoute] = JMMCfixCT
% Variables
% Centro de Transformación para elcualse calculan las rutas
CT=1;
% Numero de nodos de la red
noOfNodes = 11;
% Ubicación de los nodos en eleje de coordenadas (pixeles)
netXloc = [300 100 300 500 700 900 700 500 450 500 550];
netYloc = [800 500 200 100 200 500 800 900 700 500 300];

% Carga de la matriz de enlaces y valores de los enlaces de un fichero
% loss.mat
S = load('lossf.mat','loss');
matrix = S.loss;
clear S;
%Inicialización representación gráfica
figure('Name', 'TFC JMMC');
clf;
hold on;
% Representación Grafica
for i = 1:noOfNodes
    plot(netXloc(i), netYloc(i), '.');
    text(netXloc(i), netYloc(i),  ['  CT', num2str(i)],'Color','b');
    for j = 1:noOfNodes
            if matrix(i,j) ~= Inf
                line([netXloc(i) netXloc(j)], [netYloc(i) netYloc(j)],
'LineStyle', ':');
            end;
    end;
end;
```

```
% Establecemos límites en la Gráfica en 1000 puntos para X e Y, para una
% mejor visualización
xlim ([1 1000]);
ylim ([1 1000]);

% Hacemos el cálculo del enrutamiento llamando a lafunción dijkstra.m y
% generando las tablas de enrutamiento en un fichero OSPF.mat
for k = CT
    for l = 1:noOfNodes
        d=l;
        [path, totalCost] = dijkstra(noOfNodes, matrix, k, d);
        tPath(l)={path};
        tRoute(l)=totalCost;
        if length(path) ~= 0
            for i = 1:(length(path)-1)
                line([netXloc(path(i)) netXloc(path(i+1))], [netY-
loc(path(i)) netYloc(path(i+1))], 'Color','r','LineWidth', 2, 'Lin-
eStyle', '-.');
            end;
        end;
    end;
 end;
 save ('OSPF.mat','tPath','tRoute');
return;
```

8.2 Función dijkstra.m[23]

Efectúa el cálculo del camino más corto en una matriz con una serie de valores dados. Para nuestros cálculos prescindimos de las variables *farthestPreviousHop* y *farthestNextHop*.

8.2.1 Código

```
function [path, totalCost, farthestPreviousHop, farthestNextHop] = dijks-
tra(n, netCostMatrix, s, d)
% path: the list of nodes in the path from source to destination;
% totalCost: the total cost of the path;
% farthestNode: the farthest node to reach for each node after performing
% the routing;
% n: the number of nodes in the network;
% s: source node index;
% d: destination node index;

for i = 1:n,
%    initialize the farthest node to be itself;
    farthestPreviousHop(i) = i;     % used to compute the RTS/CTS range;
    farthestNextHop(i) = i;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% all the nodes are un-visited;
visited(1:n) = 0;

distance(1:n) = inf;    % it stores the shortest distance between each
node and the source node;
```

[23] http://www.mathworks.com/matlabcentral/fileexchange/5550-dijkstra-shortest-path-routing

```
parent(1:n) = 0;

distance(s) = 0;
for i = 1:(n-1),
    temp = [];
    for h = 1:n,
         if visited(h) == 0   % in the tree;
             temp=[temp distance(h)];
         else
             temp=[temp inf];
         end
     end;
     [t, u] = min(temp);    % it starts from node with the shortest dis-
tance to the source;
     visited(u) = 1;       % mark it as visited;
     for v = 1:n,           % for each neighbors of node u;
         if ( ( netCostMatrix(u, v) + distance(u)) < distance(v) )
             distance(v) = distance(u) + netCostMatrix(u, v);   % update
the shortest distance when a shorter path is found;
             parent(v) = u;                                     % update
its parent;
         end;
     end;
end;

path = [];
if parent(d) ~= 0   % if there is a path!
    t = d;
    path = [d];
    while t ~= s
        p = parent(t);
        path = [p path];

        if netCostMatrix(t, farthestPreviousHop(t)) < netCostMatrix(t, p)
            farthestPreviousHop(t) = p;
        end;
        if netCostMatrix(p, farthestNextHop(p)) < netCostMatrix(p, t)
            farthestNextHop(p) = t;
        end;

        t = p;
    end;
end;

totalCost = distance(d);

return;
```

8.3 Función JMMCvar

Incluimos esta función como una variante de la primera que genera de forma aleatoria una red en la que únicamente se define el número de nodos que queremos en ella y a continuación ejecuta lo mencionado en *9.1 Función JMMCfix.m*, pero para todos los nodos de la red, generando una estructura que incluye todas las rutas y todos los costes de todos los nodos.

8.3.1 Código

```
function [tRoute] = JMMCvar(n)
% Variables
% clear;
noOfNodes = n;
% noOfNodes  = 30;

% maximum range;
L = 1000;
R = 500;

netXloc = rand(1,noOfNodes)*L;
    % [400 200 200 400 600 800 800 600];
netYloc = rand(1,noOfNodes)*L;
% [800 600 400 200 800 600 400 200];
% matrix = [1 1 1 Inf 45 1 Inf Inf;1 Inf 1 Inf Inf Inf Inf Inf;1 1 1 1
Inf 1 Inf 1;Inf Inf 1 1 Inf Inf 1 1;1 Inf Inf Inf 1 1 Inf Inf;1 Inf 1 Inf
1 1 Inf 1;Inf Inf Inf 1 Inf 1 1 1;Inf Inf 1 1 Inf 1 1 1;];

figure('Name', 'TFC JMMC');
clf;
hold on;

for i = 1:noOfNodes
    plot(netXloc(i), netYloc(i), '.');
    text(netXloc(i), netYloc(i),  ['  CT', num2str(i)],'Color','b');
     for j = 1:noOfNodes
        distance = sqrt((netXloc(i) - netXloc(j))^2 + (netYloc(i) - netY-
loc(j))^2);
        if distance <= R
            matrix(i, j) = 1;   % there is a link;
            line([netXloc(i) netXloc(j)], [netYloc(i) netYloc(j)], 'Lin-
eStyle', ':');
        else
        matrix(i, j) = inf;
        end;
    end;
    % for j = 1:noOfNodes
      % distance = sqrt((netXloc(i) - netXloc(j))^2 + (netYloc(i) - netY-
loc(j))^2);
        %if matrix(i,j) ~= Inf
         %       line([netXloc(i) netXloc(j)], [netYloc(i) netYloc(j)],
'LineStyle', ':');
        %end;
    %end;
end;
xlim ([1 1000]); %Gráfica con unos límites establecidos en 1000 puntos
para X e Y
ylim ([1 1000]);

for k = 1:noOfNodes
    s=k;
    for l = 1:noOfNodes
        d=l;
        [path, totalCost] = dijkstra(noOfNodes, matrix, s, d);
        tRoute(k,l).path=path;
        tRoute(k,l).totalCost=totalCost;
        if length(path) ~= 0
```

```
            for i = 1:(length(path)-1)
                line([netXloc(path(i)) netXloc(path(i+1))], [netY-
loc(path(i)) netYloc(path(i+1))], 'Color','r','LineWidth', 2, 'Lin-
eStyle', '-.');
            end;
        end;
    end;
 end;
hold off;
return;
```

9 BIBLIOGRAFÍA

[1] M. U. Molina, Supervisión y control de redes de distribución de energía, Alcalá de Henares: Universidad de Alcalá, 2010.

[2] N. Jenkins, J. Ekanayake y G. Strbac, Distributed Generation, Stevenage: IET - Institution of Engineering and Technology, 2010.

[3] IEEE Power Engineering Society, «Substation Automation Tutorial,» Document No: 03TP166, 2004.

[4] P. Anderson, Power System Protection, New York: IEEE and McGraw-Hill, 1999.

[5] J. Grainger y W. Stevenson, Elements of Power Systems Analysis, Maidenhead: McGraw-Hill, 1994.

[6] IEC - International Electrotechnical Commission, «Techical Report 60909-1 - Short-circuit currents in three-phase a.c. systems. Second edition,» www.iec.ch, Geneva, Switzerland, 2002.

[7] E. J. B. Peña, Optimización del comportamiento de un convertidor de tres niveles NPC coentado a la red eléctrica., Alcalá de Henares: Univerisdad de Alcala, 2005.

[8] N. Hatziargyriou, H. Asano, R. Iravani y C. Marnay, «An overview of ongoing research, development and demonstration projects,» *IEEE Power and Energy Magazine*, pp. 5(4), 78–94, 2007.

[9] C. U. U. Janaka Ekanayake, U. o. P. S. L. Kithsiri Liyanage, C. U. U. JianzhongWu, U. o. T. J. Akihiko Yokoyama y C. U. U. Nick Jenkins, Smart Grid: Technology and Applications., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom: John Wiley & Sons, Ltd - WILLEY, 2012.

[10] E. P. R. Institute, Techical Report - Common Information Model Primer, First Edition, Palo Alto, CA: EPRI, 2011.

[11] IEC, *IEC 61970 - Part 301: Common Information Model (CIM) Base,* 2009.

[12] IEC, *IEC 61968 - Part 11: Common Information Model (CIM) Extensions for Distribution,* 2010.

[13] IEC, *IEC 62325 - Framework for energy market communications,* International Electrotechnical Comission, 2005.

[14] M. I. Angelina Espinoza, S. M. I. Yoseba Penya, M. M. I. Juan Carlos Nieves y a. D. R.-A. Aitor Peña, «Supporting Business Workflows in Smart Grids: An Intelligent Nodes-Based Approach,» *IEEE.,* p. 14, 2013.

[15] J. C. Nieves, A. Espinoza, Y. K. Penya, M. O. d. Mues y A. Peña, «Intelligence distribution for data processing in smart grids: A semantic approach,» *Engineering Applications of Artificial Intelligence,* p. Elsevier, March 2013.

[16] J. Hughes., «Technical Report 1012393,» de *Harmonization of IEC 61970, 61968, and 61850 Models,* Electric Power Research Institute (EPRI), December 2006.

[17] D. Becker., «Technical Report 1020098,» de *Harmonizing the International Electrotechnical Commission Common Information Model (CIM) and 61850.,* Electric Power Research Institute (EPRI), May 2010.

[18] W3C, *OWL 2 Web Ontology Language,* W3C, 2008.

[19] J. T. Moy, OSPF: Anatomy of an Internet Routing Protocol, Upper Saddle River, New Jersey.: Pearson Education Corporate Sales Division, 1998.

[20] J. Reynolds y J. Postel., Assigned Numbers, October 1994.

[21] E. Dijkstra, «A note on Two Problems in Connexion with Graphs,» de *Numerische mathematik,* 1959, pp. 269-271.

[22] M. Jiménez y N. Hatziargyriou, «Research Activities in Europe on Integration of Distributed Energy Resources in the Electricity Networks of the Future,» *IEEE Power Engineering Society General Meeting,* pp. 1-4, 2006.

[23] D. Vallejo, J. Albusac, C. Glez-Morcillo, J. J. Castro-Schez y L. Jiménez, «A multi-

agent approach to intelligent monitoring in smart grids,» *International Journal of Systems Science,* p. http://dx.doi.org/10.1080/00207721.2013.783644, Apr 2013.

[24] S. M. I. Sridhar Chouhan, M. I. Hui. Wan, A. F. S. I. H.J.Lai y S. M. I. M. A. Choudhry, «Intelligent Reconfiguration of Smart Distribution Network using Multi-Agent Technology,» *IEEE,* pp. 978-1-4244-4241-6/09, 2009.

[25] M. Wooldridge, «Intelligent agents,» de *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.*, G.Weiss, 1999.

[26] A. S. Rao y M. P. Georgeff., «BDI-agents: From Theory to Practice,» *Proceedings of the First International Conference on Multiagent Systems - ICMAS'95,* 1995.

[27] P. Jha y M. S. Vidyasagar, «Dijkstra Algorithm for Feeder Routing of Radial Distribution System,» *IOSR Journal of Engineering (IOSRJEN),* pp. Vol. 3, Issue 1, Jan. 2013.

[28] a. Y. Y. S. M. I. D. o. E. E. T. U. Jianzhong Wu, «A New Method for Snapshot and Time-Varying Distribution Network Reconfiguration,» *Proceedings of the 5" World Congress on Intelligent Control and Automation, Hangzhou, P.R. China,* June 15-19, 2004.

[29] S. U. o. T. T. Niknam, J. Olamei, A. Arefi, A. H. Mazinan y S. T. B. Islamic Azad University, «A Novel Hybrid Evolutionary Algorithm Based on ACO and SA for Distribution Feeder Reconfiguration with Regard to DGs,» de *IEEE GCC Conference and Exhibition (GCC), Dubai, United Arab Emirates,* February 19-22, 2011.

[30] P. W. Z. L. a. Y. L. W. Li, «Reliability evaluation of complex radial distribution systems considering restoration sequence and network constraints.,» *IEEE Trans. Power Del.,* vol. 19, nº 2, p. 753–758, Apr. 2004..

[31] S. C. a. S. N. S. B. Amanulla, «Reconfiguration of Power Distribution Systems Considering Reliability and Power Loss,» *IEEE TRANSACTIONS ON POWER DELIVERY,* vol. VOL. 27, nº NO. 2, April 2012.

[32] B. Weedy y B. Cory, Electric Power Systems, New York: John Wiley and Sons, 2004.

[33] J. C. Nieves, M. O. d. Mues, A. Espinoza y D. Rodriguez-Alvarez., «Harmonization of semantic data models of electric data standards,» de *IEEE 9th International Conference on Industrial Informatics (INDIN 2011)*, 2011.

[34] British Standard Institute, BS EN/IEC 61850: Communication Networks and Systems for Power Utility Automation., 2010.

[35] Alstom Grid, Network Protection and Automation Guide: Protective Relays, Measurements and Control, Available from http://www.alstom.com/grid/NPAG/ on request., May 2011.

[36] IEC, *IEC 61970 - Part 501: CIM RDF Schema*, 2006.

[37] IEC, *IEC 61968 - Part 13: CIM RDF Model exchange format for distribution*, 2008.

[38] IEC, *IEC 61850 - General Requirements, Ed-1*, 2002.

[39] IEC, *IEC 61850 - Introduction and overview,* 2003.