Redes de comunicación deterministas

PID_00249432

Francisco Vázquez Gallego
Pere Tuset Peiró

Tiempo mínimo de dedicación recomendado: 4 horas

Universitat Oberta de Catalunya

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Índice

1. Introducción 5

2. Redes *time-sensitive* cableadas 10
2.1. Introducción 10
2.2. Estándares IEEE TSN 12
2.2.1. IEEE 802.1AS-Rev: Time Synchronization 12
2.2.2. IEEE 802.1Qbv: Time Aware Traffic Shaping 15
2.2.3. IEEE 802.1Qbu: Frame Preemption 17
2.2.4. IEEE 802.1CB: Frame Replication and Elimination for Reliability 18
2.2.5. IEEE 802.1Qcc: Stream Reservation Protocol (SRP) Enhancements 20
2.2.6. IEEE 802.1Qci: Ingress Policing 21
2.2.7. IEEE 802.1Qch: Cyclic Queuing and Forwarding 22
2.2.8. IEEE 802.1Qca - Path Control and Reservation 25
2.3. Implementaciones 25

3. Redes *time-sensitive* inalámbricas 28
3.1. Introducción 28
3.2. Estándares IEEE-IETF 6TiSCH 30
3.2.1. IEEE 802.15.4 PHY 31
3.2.2. IEEE 802.15.4e MAC (TSCH) 32
3.2.3. IETF 6top Protocol (6P) 37
3.2.4. IETF 6LowPAN 40
3.2.5. IETF RPL 42
3.2.6. IETF CoAP 42
3.3. Implementaciones 42

Bibliografía 45

1. Introducción

La conmutación de circuitos fue la primera técnica de conmutación utilizada en redes de comunicaciones para la transmisión de señales. En conmutación de circuitos, cuando dos terminales necesitan comunicarse, se establece un canal con un ancho de banda fijo, denominado circuito, entre la fuente y el destino. Este circuito se reserva exclusivamente para un único flujo de información, y ningún otro flujo puede utilizar dicho circuito.

Por tanto, una red basada en la técnica de conmutación de circuitos ofrece un servicio orientado a conexión, y si la capacidad del circuito es mayor o igual que la velocidad de transmisión máxima del flujo de información, entonces la calidad de servicio (en inglés, *quality of service*, QoS) es máxima en términos de retardo y ancho de banda.

Uno de los inconvenientes de la técnica de conmutación de circuitos es que se desperdicia el ancho de banda en los intervalos de tiempo en los que la fuente de información deja de transmitir. Es decir, el ancho de banda del canal se utiliza de forma ineficiente. Para resolver este y otros problemas, se introdujo la técnica de conmutación de paquetes, que es la base del protocolo de internet (en inglés, *internet protocol*, IP).

En la técnica de conmutación de paquetes, los flujos de información se dividen en paquetes de datos de tamaño variable. Cada paquete incorpora la dirección de origen y destino, y los paquetes se envían uno a uno hacia el *switch* (conmutador) más cercano. A su vez, cada *switch* chequea la dirección de destino de los paquetes y los reenvía al siguiente *switch* que corresponda dependiendo del valor de la dirección de destino de cada paquete, así como la topología y el estado de la red. Este proceso se repite hasta que el paquete llega a su destino.

El mecanismo de conmutación de paquetes realizado en cada *switch* se denomina *store-and-forward* (almacena y reenvía) porque los paquetes son recibidos completamente por el *switch*, almacenados y procesados, y finalmente transmitidos de nuevo. Si dos o más paquetes deben ser transmitidos hacia un mismo *switch* en el siguiente salto, entonces los paquetes son almacenados en *buffers* de memoria, pero si los *buffers* del *switch* están llenos, entonces los esos paquetes son descartados y se pierden.

Por tanto, la técnica conmutación de paquetes ofrece un servicio no orientado a conexión y *best effort* (o de mejor desempeño). Es decir, las redes basadas en la técnica conmutación de paquetes (como internet y las redes de área local

como Ethernet) no pueden garantizar los siguientes parámetros de calidad de servicio (*quality of service*, QoS) para un determinado flujo de información entre dos estaciones:

- Tasa de paquetes perdidos,
- ancho de banda,
- retardo máximo, y
- fluctuación máxima del retardo (en inglés, *jitter*).

A pesar de que las redes Ethernet no son la mejor solución para aplicaciones con requisitos de tiempo real, en la actualidad, los sistemas de comunicaciones industriales basados en redes Ethernet se utilizan en multitud de aplicaciones de automatización industrial y se pueden considerar como los sustitutos de los buses de campo tradicionales.

En los últimos años se han desarrollado diversos protocolos Industrial Ethernet propietarios y mutuamente incompatibles. Los sistemas Industrial Ethernet se pueden clasificar en tres categorías o clases dependiendo del tipo de implementación y de su capacidad de respuesta en tiempo real, es decir, de su capacidad para transmitir datos con bajo retardo. La figura 1 muestra las tres clases de sistemas Industrial Ethernet existentes.

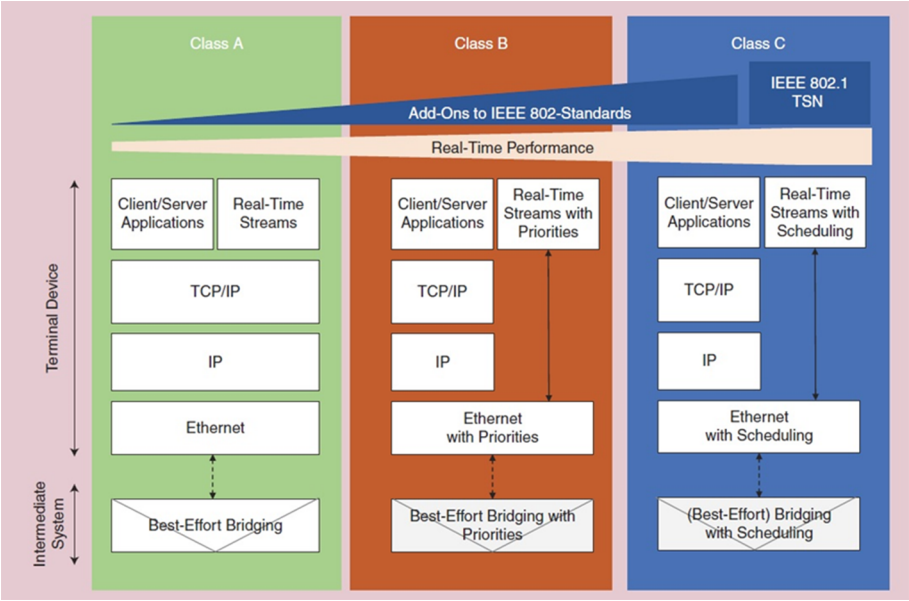
Los sistemas de clase A utilizan la pila de protocolos del estándar de Ethernet y TCP/IP, y añaden las funcionalidades específicas de tiempo real en la capa de aplicación. La principal limitación de los sistemas de clase A es que se basan en el modelo *best-effort*, utilizando *switches* de Ethernet que funcionan según el estándar IEEE 802.1D, lo cual introduce retardos no deterministas en la transmisión de datos y puede provocar pérdidas de paquetes en caso de congestión de la red. Por consiguiente, los sistemas de clase A ofrecen un rendimiento en tiempo real moderado, con valores de retardo no determinista en torno a los 100 ms. Entre los sistemas de clase A destacan Modbus/TCP, EtherNet/IP y PROFINET.

Los sistemas de clase B introducen el modelo de priorización de tráfico y el mecanismo Virtual Local Area Network (VLAN) *tagging*, definidos en el estándar IEEE 802.1Q, para separar el tráfico de tiempo real del tráfico *best-effort*. Los sistemas de clase B implementan las funcionalidades específicas de tiempo real encima de la capa MAC de Ethernet, por tanto, utilizan *switches* de Ethernet estándar. En los sistemas de clase B, la separación del tráfico de tiempo real y el tráfico *best-effort* permite conseguir valores de retardo entre 1 ms y 10 ms en el tráfico con la prioridad más alta. Entre los sistemas de clase B destaca PROFINET RT.

El principal problema de los sistemas de clase B es que una trama de tiempo real podría ser retrasada por una trama de menor prioridad en cualquier *switch* del sistema. Para resolver el problema, en los sistemas de clase C se modifica la capa MAC del estándar de Ethernet introduciendo mecanismos de *scheduling*

y de sincronización de los relojes de los *switches* y de los dispositivos de campo usando el estándar IEEE 1588. Los sistemas de clase C ofrecen valores de latencia inferiores a 1 ms, pero son incompatibles con el estándar de Ethernet. Entre los sistemas de clase C destacan POWERLINK, PROFINET IRT, EtherCAT y SERCOS III.

Figura 1. Clasificación de los sistemas Industrial Ethernet



Fuente: extraído de Wollschlaeger, Sauter y Jasperneite (2017)

En paralelo al desarrollo de los sistemas Industrial Ethernet propietarios, el desarrollo de un sistema Ethernet estándar para aplicaciones con requisitos de tiempo real se inició en el año 2004 con la aparición del IEEE 802.3 Residential Ethernet Study Group. El objetivo de este grupo de trabajo consistía en desarrollar una nueva especificación de Ethernet para aplicaciones residenciales, y contaba con la contribución de empresas como Broadcom, Nortel, Pioneer, Samsung, NEC y Gibson Brands. Esta actividad continuó realizándose a partir del año 2005 en el IEEE 802.1 Audio/Video Bridging (AVB) Task Group, que en 2012 culminó con el desarrollo del protocolo IEEE 802.1 Ethernet AVB, también denominado protocolo AVB.

El protocolo AVB introdujo mejoras significativas para aplicaciones de transmisión de flujos de datos sincronizados de audio y vídeo sobre redes Ethernet. Los principios básicos del protocolo AVB son los siguientes:

1) Conseguir que todas las estaciones de la red estén sincronizadas, ofreciendo una referencia de *clock* precisa en toda la red.

2) Limitar el retardo introducido por la red a un valor conocido y pequeño (*bounded latency*).

3) Permitir la transmisión de patrones de tráfico *delay-sensitive* (con requisitos de retardo reducido) y tráfico *delay-tolerant*.

4) Evitar que el tráfico *delay-tolerant* degrade el rendimiento de la red en la transmisión de tráfico *delay-sensitive*.

El objetivo del protocolo AVB consistía en permitir que las estaciones pudieran reservar un canal de comunicación entre una estación fuente y una estación destino, con un determinado ancho de banda, garantizando un retardo máximo en la transmisión de datos y asegurando un valor de *jitter* máximo. En definitiva, se pretendía hacer que el funcionamiento de una red de conmutación de paquetes fuera muy parecido al de las redes basadas en la técnica de conmutación de circuitos.

Los principales estándares que componen el protocolo AVB son los siguientes:

- IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications (gPTP),

- IEEE 802.1Qat: Stream Reservation Protocol (SRP), y

- IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams (FQTSS).

En una red basada en el protocolo AVB, todas las estaciones están sincronizadas de manera precisa usando gPTP (IEEE 802.1AS), lo cual permite que todas las estaciones tengan la misma referencia de tiempos.

El principio de funcionamiento del protocolo AVB se describe a continuación. En primer lugar, usando SRP (IEEE 802.1Qat), la estación fuente (*Talker*) realiza la petición de un canal de comunicación con una estación destino (*Listener*) indicando los requisitos de ancho de banda, retardo y *jitter*. La petición es reenviada a través de los *switches* de la red hacia el *Listener*, reservando el ancho de banda, acumulando los retardos de cada enlace entre *switches* y calculando el *jitter* hasta llegar al *Listener*. Una vez que se ha establecido el canal de comunicación y se inicia el flujo de información, los *switches* realizan el enrutamiento de paquetes y conforman el tráfico (en inglés, *traffic shaping*) usando el algoritmo *Credit Based Shaper* (CBS) definido en FQTSS (IEEE 802.1Qav).

En el protocolo AVB se definen dos clases de tráfico AVB *stream reservation* (SR): SR-Clase A y SR-Clase B. La diferencia entre ambas clases es el requisito de retardo máximo a través de una red de siete saltos. En una red Ethernet AVB con enlaces de 100 Mbps, para SR-Clase A, el retardo máximo es de 2 ms; y para SR-Clase B, el retardo máximo es de 50 ms. Además, el ancho de banda reservado para ambas clases SR está limitado a un 75 % del ancho de banda

total. Es decir, si los enlaces son de 100 Mbps, entonces se puede reservar un ancho de banda máximo de 75 Mbps para paquetes AVB, y el 25 % restante del ancho de banda se utiliza para paquetes que no son AVB.

Durante el desarrollo del protocolo AVB, un grupo de investigadores y empresas del segmento de la automatización industrial vio que el protocolo AVB podía ser útil en aplicaciones industriales con requisitos de tiempo real. Sin embargo, en la última fase del desarrollo de los estándares AVB se descubrió que el algoritmo CBS no era suficientemente robusto y que había casos en los que se podían producir pérdidas de paquetes de tráfico crítico (SR) por congestión. Esto fue visto como un problema grave por parte del sector industrial.

En el año 2012, para reflejar mejor el alcance del protocolo hacia aplicaciones industriales, el IEEE 802.1 AVB Task Group fue renombrado como IEEE Time-Sensitive Networking (TSN) Task Group, que empezó a trabajar en el desarrollo de un nuevo conjunto de estándares que resolviera los problemas del protocolo AVB.

Por otro lado, en la Internet Engineering Task Force (IETF) también se está trabajando en la definición de estándares para la implementación de *Deterministic Networks* (DetNet), con las mismas características que las de la tecnología TSN. Los estándares de IETF y IEEE TSN son complementarios, y buena parte de los del IETF se centran en DetNet inalámbricas.

La organización del resto del capítulo es la siguiente. La sección 2 está dedicada al estudio de las redes *time-sensitive* cableadas, también denominadas «Deterministic Ethernet», y la sección 3 está dedicada a las redes *time-sensitive* inalámbricas. En ambas secciones se describen los objetivos de la tecnología, las aplicaciones potenciales, algunos detalles sobre los estándares que las componen, y ejemplos de implementación.

2. Redes *time-sensitive* cableadas

2.1. Introducción

El IEEE Time-Sensitive Networking Task Group está trabajando desde 2012 en el desarrollo de un conjunto de estándares que permitan introducir el funcionamiento en tiempo real, requisito fundamental de las aplicaciones industriales, en redes Ethernet.

Los nuevos estándares de TSN describen diversos mecanismos que permiten asegurar el determinismo en los retardos introducidos por la red, asegurando la calidad de servicio en la transmisión de diferentes tipos de tráfico a través de la red Ethernet: tráfico *delay-sensitive* de control, tráfico *delay-tolerant* de control, y tráfico *best effort*.

El objetivo fundamental de la tecnología TSN consiste en resolver las siguientes necesidades de las aplicaciones industriales:

- Sincronizar los nodos de la red, de manera precisa, sencilla y robusta.

- Permitir la reserva de un ancho de banda para un determinado flujo de datos, garantizando que dicho flujo tenga un valor de latencia bajo y determinista, y sin pérdidas de datos por congestión.

- Evitar que el tráfico *best effort* introduzca retardos indeseados en el tráfico *delay-sensitive*.

- Permitir que un flujo de datos pueda tomar múltiples caminos a través de la red (redundancia) para asegurar que el fallo de un *switch* o de un enlace no provoque pérdidas de paquetes.

- Diagnosticar fallos en la red de manera rápida gracias a una información más precisa sobre el origen del fallo.

Los estándares de TSN empezaron a desarrollarse para introducir mejoras en el protocolo IEEE 802.1 Ethernet AVB descrito anteriormente. Dos de las diferencias clave de TSN respecto a AVB se describen a continuación:

- **Mecanismo de *scheduling* centralizado**. En el protocolo AVB se emplea un mecanismo descentralizado para crear y mantener un canal de comunicación para un flujo de datos con un determinado ancho de banda, asegurando un retardo y un *jitter* máximos. Cuando el número de flujos de datos es elevado, y al aumentar el tráfico en la red, entonces es necesario

modificar los enlaces asignados a los distintos flujos para evitar congestión en los *switches*. Este proceso es complejo, si no imposible, utilizando un mecanismo descentralizado. En TSN se introduce un mecanismo centralizado en el que un elemento de la red realiza las funciones de controlador y se encarga de establecer y mantener los canales según sea el estado de la red.

- **Algoritmo de *traffic shaping* temporal** (en inglés, ***time based shaping*, TBS**). En el protocolo AVB, las estaciones de la red realizan funciones de *trafficshaping* aprovechando que están sincronizadas. Por otro lado, los *switches* también realizan *traffic shaping* mediante el algoritmo CBS (*credit based shaper*), pero sin tener en cuenta el tiempo. En TSN, los *switches* están sincronizados e implementan un algoritmo de *traffic shaping* temporal, denominado TBS, en el que utilizan el tiempo para conformar el tráfico.

En TSN, los términos *switch* y *bridge* se utilizan indistintamente para referirse a los dispositivos que realizan las funciones de conmutación avanzada de paquetes.

En la tabla 1 se resumen los principales sectores de la industria para los que la tecnología TSN es fundamental y se detallan algunos de los casos de uso potenciales en cada sector.

Tabla 1. Ejemplos de aplicaciones de la tecnología TSN

Sector	Aplicaciones
Sistemas de control industrial (fabricación)	Comunicación entre máquinas, dispositivos de control, sensores y actuadores Aplicaciones de control de movimiento
Transporte	Aplicaciones de control en trenes Control y monitorización de infraestructuras de transporte (carreteras, ferrocarril, etc.)
Energía	Sistemas de control y de seguridad certificados Sistemas de protección en Smart Grids
Automoción	Aplicaciones de seguridad y control en automóviles que integran dispositivos sensores (cámaras, radar, etc.) con requisitos de gran ancho de banda
Aeronáutico y espacial	Red troncal de comunicaciones en aeronaves, satélites, lanzaderas, etc. Sistemas de seguridad y control robustos con tolerancia a fallos y redundancia

El resto de esta sección está dedicado al estudio y análisis de la tecnología TSN y se ha organizado de la forma siguiente. La sección 2.2 presenta los estándares que constituyen la tecnología TSN y describe sus características principales. Finalmente, en la sección 2.3 se presentan las soluciones existentes para implementar redes basadas en tecnología TSN.

2.2. Estándares IEEE TSN

En la tabla 2 se muestra el conjunto de estándares desarrollados por el IEEE Time-Sensitive Networking Task Group. Para cada estándar se indica la referencia, la funcionalidad asociada y el título del estándar.

Tabla 2. Estándares IEEE Time Sensitive Networking

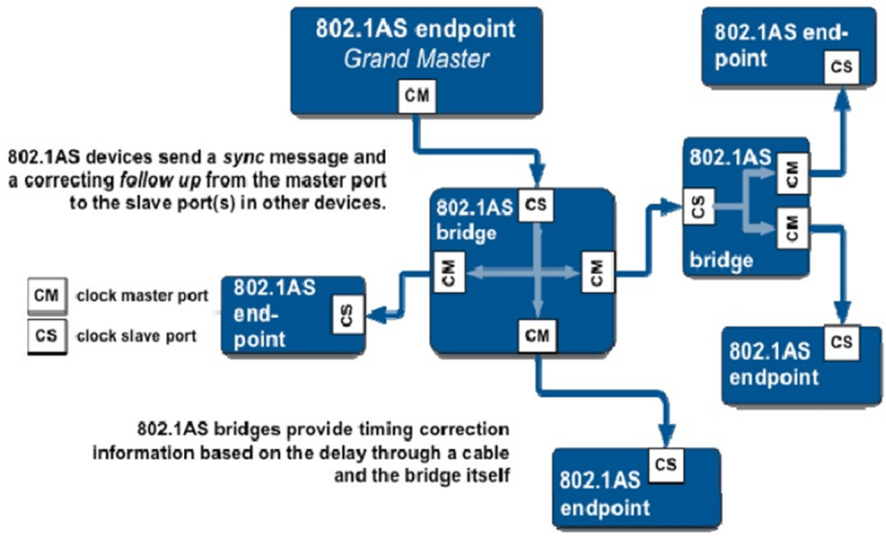
Estándar	Funcionalidad	Título
IEEE 802.1AS-Rev	Time Synchronization	Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks
IEEE 802.1Qbv	Forwarding and Queuing	Enhancements for Scheduled Traffic – Time-Aware Traffic Shaping
IEEE 802.1Qbu	Forwarding and Queuing	Frame preemption
IEEE 802.1Qca	Path Control and Seamless Redundancy	Path Control and Reservation
IEEE 802.1Qcc	Centralized Configuration	Stream Reservation Protocol (SRP) Enhancements and Performance Improvements
IEEE 802.1CB	Path Control and Seamless Redundancy	Frame Replication and Elimination for Reliability
IEEE 802.1Qch	Forwarding and Queuing	Cyclic Queuing and Forwarding
IEEE 802.1Qci	Time Based Ingress Policing	Per-Stream Filtering and Policing

En los apartados siguientes se describen las principales características de los estándares IEEE TSN detallados en la tabla anterior.

2.2.1. IEEE 802.1AS-Rev: Time Synchronization

El estándar IEEE 802.1AS-Rev define el protocolo y los procedimientos utilizados para asegurar que se cumplen los requisitos de sincronización para aplicaciones *time-sensitive* en redes de área local en las que los retardos de transmisión en los enlaces entre dos *switches* (o *bridges*) son constantes y simétricos. El estándar incluye el mantenimiento de la sincronización durante el funcionamiento de la red en condiciones normales, y después de añadir, extraer o fallar algún componente de la red. Este estándar está basado en las especificaciones del IEEE 1588 Precision Time Protocol (PTP).

Figura 2. Sincronización de relojes en TSN



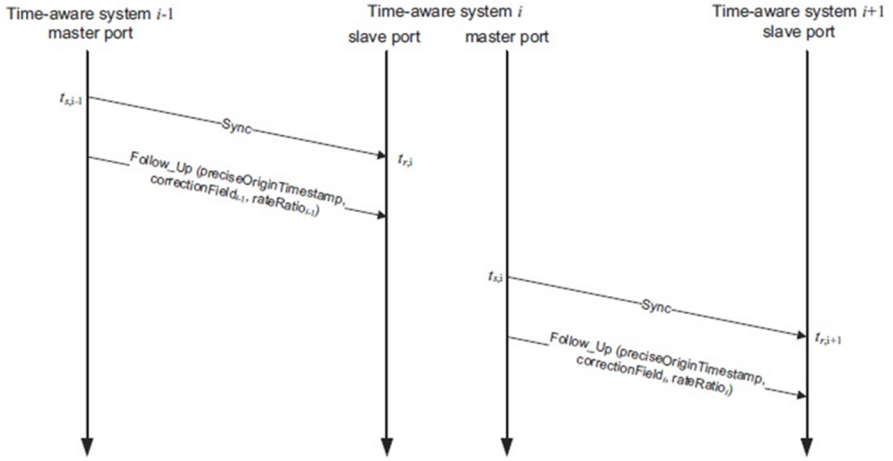
Fuente: extraído de Michael D. Johas Teener, «Time-Sensitive Networks, a primer». TSNA (05/02/2012)

Tal como muestra la figura 2, el estándar IEEE 802.1AS-Rev utiliza un protocolo *master-slave* para la sincronización de los relojes de los dispositivos. En la figura, los terminales o estaciones de la red se denominan *end-points*. La estación que contiene el reloj más preciso es la que realiza las funciones de *GrandMaster*. Se utiliza un método de selección de *GrandMaster* denominado *Best Master Clock Algorithm* (BMCA). En caso de fallo de conexión del *GrandMaster* actual, el BMCA permite reasignar el rol de *GrandMaster* a cualquier otra estación de la red.

El procedimiento de sincronización de relojes se representa en la figura 3. El *GrandMaster* envía un mensaje *Sync* con el valor de tiempo actual al *switch* al cual está conectado a través de su *clock master port* (CM). El *switch* recibe al mensaje a través de su *clock slave port* (CS) y corrige el valor de tiempo actual añadiendo el retardo de propagación necesario para transmitir los mensajes a través del enlace entre el *GrandMaster* y el *switch*. A continuación, el *switch* envía un mensaje *Sync* con el valor de tiempo actual corregido hacia todos los *switches* o *end-devices* que tenga conectados en sus CMs. El proceso de corrección del tiempo actual y reenvío se repite en cada *switch* hasta que los mensajes llegan a los *end-points*.

Para poder realizar la corrección de tiempos en los *switches* y en los *end-points* es necesario conocer el retardo de retransmisión de mensajes en el *switch* y el tiempo de propagación del mensaje. El retardo de retransmisión de mensajes es un parámetro interno del *switch* fácil de medir y pequeño respecto al tiempo de propagación. Por el contrario, el tiempo de propagación depende de las propiedades del medio y de la longitud del enlace.

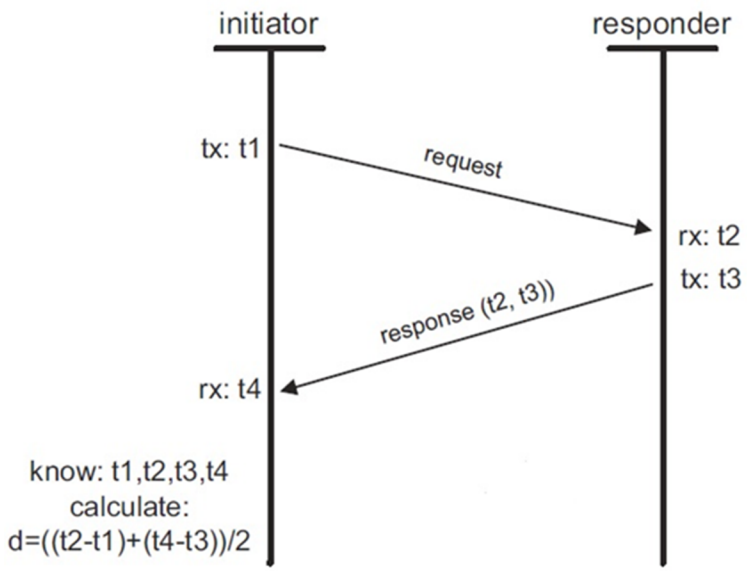
Figura 3. Procedimiento de sincronización de relojes



Fuente: extraído del estándar IEEE 802.1AS-Rev

El procedimiento de medida del retardo de propagación del mensaje a través de un enlace de la red se representa en la figura 4. El dispositivo que inicia la medida de retardo envía un mensaje de tipo *request* en el instante t1, el dispositivo *responder* recibe el *request* en el instante t2 y envía un mensaje de tipo *response* en el instante t3. El dispositivo *initiator* recibe el mensaje *response* en el instante t4 y calcula el valor del retardo de propagación, d = ((t2 – t1) + (t4 – t3))/2.

Figura 4. Medida del retardo de propagación



Fuente: extraído del estándar IEEE 802.1AS-Rev

El procedimiento de sincronización de relojes depende de la precisión de la medida del retardo de propagación en el enlace y del retardo de retransmisión de mensajes. Si el reloj utilizado en los dispositivos para realizar estas medidas de tiempo estuviera sintonizado con el del *GrandMaster*, entonces todas las medidas temporales tendrían la misma base de tiempos. Sin embargo, debido a

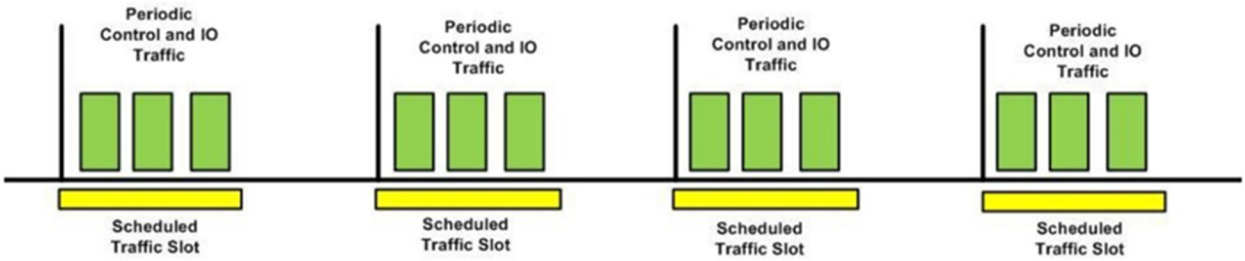
que el ajuste de frecuencia de los osciladores es un proceso lento, la corrección temporal se realiza utilizando los ratios entre la frecuencia del reloj de cada dispositivo y el reloj del *GrandMaster*.

La sincronización temporal de todos los dispositivos que componen la red es fundamental para que puedan implementarse el resto de procedimientos utilizados en TSN, que se describen en los apartados siguientes.

2.2.2. IEEE 802.1Qbv: Time Aware Traffic Shaping

Las aplicaciones en las que se implementan lazos de control (por ejemplo, robótica, automatización industrial, sistemas de control en el automóvil, etc.) necesitan que los datos de control se reciban en los dispositivos en instantes precisos, con un retardo y un *jitter* mínimos. Los paquetes que transportan los datos de control suelen transmitirse dentro de un ciclo (o *schedule*) que se repite de forma periódica, y un retardo elevado puede provocar la inestabilidad del sistema o un error en la operación del lazo de control. En la figura 5 se muestra un ejemplo de *schedule* para el tráfico de control y de sensores enviado periódicamente.

Figura 5. Ejemplo de *schedule* para la transmisión de tráfico periódico de control y de sensores

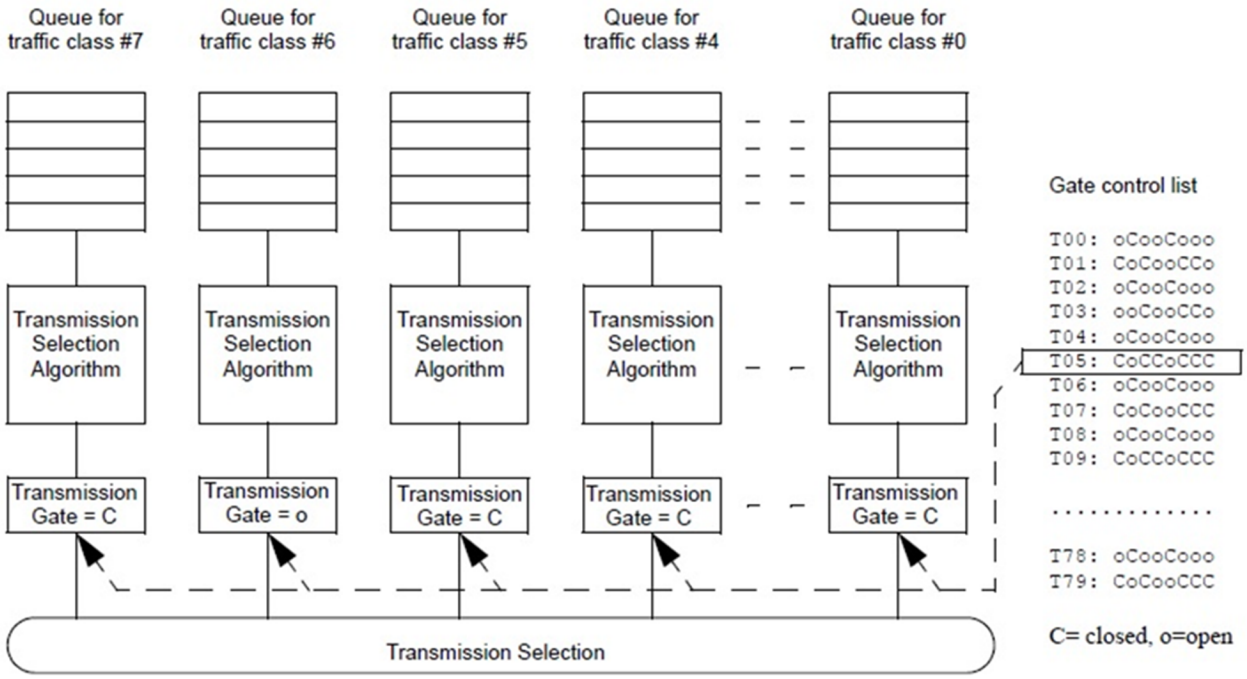


Fuente: extraído de «2015_ODVA_Conference_Ditzel-Didier_TSN.pdf»

En las redes Ethernet convencionales, la asignación de prioridades a los diferentes tipos de tráfico, propuesta por el estándar IEEE 802.1Q, no es suficiente para resolver las necesidades del tráfico de control. Por ejemplo, si un paquete de baja prioridad está siendo transmitido por un *switch*, dicha transmisión deberá finalizar antes de que otro paquete de mayor prioridad pueda enviarse, lo cual introduce un retardo que podría llegar al del tiempo de transmisión de un paquete de longitud máxima. Si estos retardos ocurren en cada *switch*, entonces el retardo acumulado en la red podría ser inaceptable. Además, el tráfico interferente también podría ser de alta prioridad; por ejemplo, cuando existen múltiples flujos de datos de diferentes sistemas de control. En ese caso, es importante poder coordinar los diferentes flujos para que no interfieran entre ellos.

Para resolver los problemas anteriores, el estándar IEEE 802.1Qbv divide el tráfico en diversas clases o prioridades, y asegura que en determinados *slots* (ranuras temporales) de un *schedule* solo pueda transmitirse una o varias clases de tráfico específicas.

Figura 6. Estructura de colas y puertas de transmisión



Fuente: extraído del estándar IEEE 802.1Qbv

En el estándar IEEE 802.1Qbv, cada estación y *switch* de la red incluye un conjunto de colas, y cada cola se utiliza para almacenar paquetes recibidos de una clase de tráfico. En la figura 6 se muestra la estructura de colas propuesta en el estándar IEEE 802.1Qbv. La transmisión de los paquetes almacenados en cada cola se realiza dentro de un *slot* del *schedule*. Cada cola tiene asociada una puerta de transmisión (en inglés, *transmission gate*), cuyo estado determina si los paquetes que hay en la cola pueden ser seleccionados para ser transmitidos. Si la cola se encuentra en estado abierto (*open*), sus paquetes son seleccionados para transmisión. Si la cola se encuentra en estado cerrado (*closed*), sus paquetes no son seleccionados para transmisión. Además, un paquete de una cola determinada tampoco está disponible para transmisión si no hubiera tiempo suficiente para transmitir el paquete completo dentro de un *slot* antes de que la puerta pase a estado cerrado.

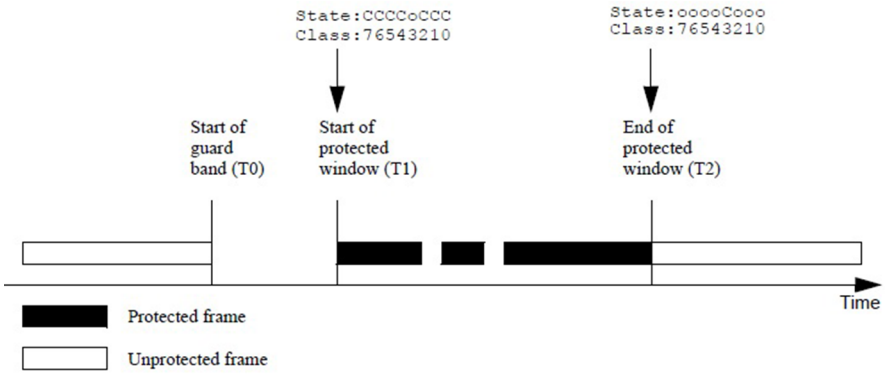
Cada uno de los puertos de un *switch* tiene asociada una lista de control de puerta (en inglés, *gate control list*) que contiene una secuencia de operaciones a realizar en las puertas. Cada operación de puerta cambia el estado de una puerta de transmisión asociada a cada cola del puerto. En una implementación que no soportara la mejora de tráfico en *schedule*, todas las puertas estarían permanentemente en estado abierto.

La secuencia de operaciones especificada por lista de control de puerta se repite cíclicamente. Al llegar al último elemento de la lista, el proceso se reinicia en el primer elemento de la lista de control de puerta, implementando un *schedule* que permite la apertura y cierre de puertas de transmisión en instantes de tiempo predefinidos.

Cuando varias puertas de transmisión están abiertas simultáneamente durante un intervalo de tiempo, y si hay paquetes almacenados en diversas colas asociadas a dichas puertas, entonces el algoritmo de selección de prioridades del estándar IEEE 802.1Q funciona normalmente.

La figura 7 ilustra un ejemplo de secuencia de operaciones de puerta en la que se incluye una ventana temporal solo para el tráfico de clase 3. En el instante T1 se cierran las puertas de los tráficos de clase 7, 6, 5, 4, 2, 1 y 0, y se abre la puerta de transmisión del tráfico de clase 3. En el instante T2 se cierra la puerta de clase 3 y se abren el resto de puertas.

Figura 7. Ejemplo de operaciones de la lista de control de puertas



Fuente: extraído del estándar IEEE 802.1Qbv

2.2.3. IEEE 802.1Qbu: Frame Preemption

En las redes Ethernet convencionales, los *switches* funcionan de forma no preemptiva. Es decir, si llega un paquete de prioridad alta (tráfico crítico de control) mientras se está transmitiendo un paquete de menor prioridad (tráfico no crítico) por uno de los puertos de un *switch*, la transmisión del paquete no crítico debe finalizar por completo antes de enviar el paquete crítico, introduciendo un retardo en la transmisión del paquete crítico.

La figura 8 muestra un ejemplo de transmisión de un paquete (o trama, *frame* en inglés) no crítico (en azul) en un *slot* reservado para el envío periódico de tráfico crítico (en verde). Mientras se está transmitiendo el paquete no crítico, se recibe un paquete crítico. Al tratarse de un *switch* funcionando en modo no preemptivo, el paquete crítico se envía cuando finaliza la transmisión del

paquete no crítico. Por tanto, el paquete crítico (en verde) se envía con un retardo excesivo y fuera del *slot* reservado para tráfico crítico, lo cual podría provocar errores en la operación del lazo de control y la inestabilidad del sistema.

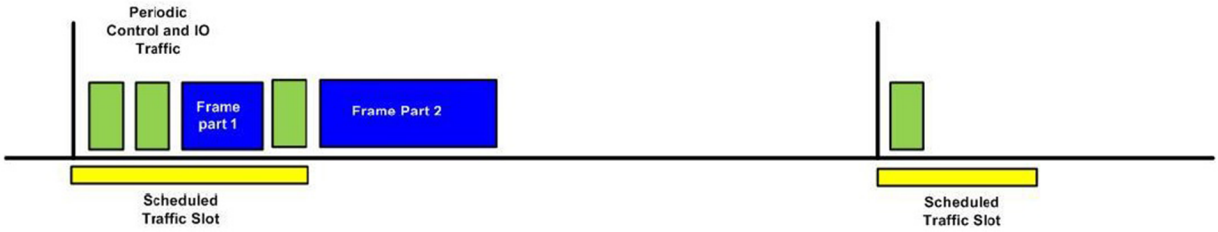
Figura 8. Ejemplo de funcionamiento no preemptivo

Fuente: extraído de «2015_ODVA_Conference_Ditzel-Didier_TSN.pdf»

El estándar IEEE 802.1Qbu propone el mecanismo Frame Preemption, que permite reducir el retardo, introducido por paquetes interferentes de baja prioridad, en la transmisión de tráfico crítico (de alta prioridad). El mecanismo Frame Preemption consiste en interrumpir la transmisión de paquetes no críticos, al recibir un paquete crítico, y retomar y finalizar la transmisión del paquete no crítico una vez que el paquete crítico ha sido transmitido.

En la figura 9 se muestra un ejemplo del funcionamiento del mecanismo Frame Preemption. El paquete no crítico (en azul) es interrumpido al llegar el paquete crítico (en verde), y la transmisión del paquete no crítico continúa una vez que el paquete crítico ha sido transmitido. El paquete no crítico se ha dividido en dos partes para evitar un retardo excesivo en la transmisión del paquete crítico, permitiendo que este pueda transmitirse en el *slot* reservado para el envío periódico de tráfico crítico.

Figura 9. Ejemplo de funcionamiento del mecanismo Frame Preemption



Fuente: extraído de «2015_ODVA_Conference_Ditzel-Didier_TSN.pdf»

2.2.4. IEEE 802.1CB: Frame Replication and Elimination for Reliability

Un sistema de comunicaciones para aplicaciones industriales debe ser robusto y fiable, y capaz de tolerar pérdidas de paquetes debidas a congestión, fallos de enlace, roturas de cableado, fallos de alimentación, etc.

El estándar IEEE 802.1CB especifica un conjunto de técnicas y procedimientos que permite el envío de copias duplicadas (réplicas) de paquetes de tráfico crítico a través de caminos diversos a través de la red. De este modo, si los dos paquetes llegan a la estación destino, la réplica se elimina, y si se perdiera uno de los paquetes, todavía existe la posibilidad de que el paquete duplicado llegue a su destino. Es decir, se introduce redundancia para conseguir mayor fiabilidad en la transferencia de datos.

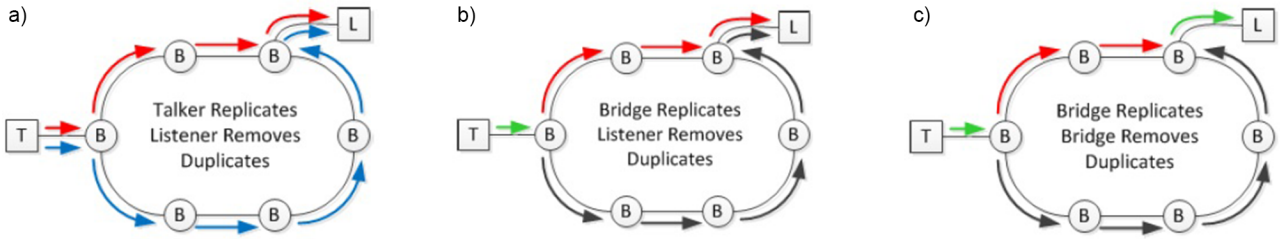
En particular, el estándar IEEE 802.1CB permite la creación y eliminación de copias duplicadas de paquetes en los *switches* y en cualquier estación de la red.

En la figura 10.a se muestra un ejemplo en el que la copia del paquete se crea en la estación fuente (*talker*). El paquete original y la copia viajan por dos caminos diferentes hasta llegar a la estación de destino, donde se elimina la copia del paquete.

En la figura 10.b se muestra un ejemplo en el que la copia del paquete se crea en el *switch* más cercano a la estación fuente. El paquete original y la copia viajan por dos caminos diferentes hasta llegar a la estación de destino, donde se elimina la copia del paquete.

En la figura 10.c se muestra un ejemplo en el que la copia del paquete se crea en el *switch* más cercano a la estación fuente y la copia se elimina en el *switch* más cercano a la estación destino.

Figura 10. Ejemplos de Frame Replication and Elimination



Fuente: extraído de «03_IEEE_TSN_Standards_Overview_and_Update_v4.pdf»

Si se creara una copia de todos los paquetes transmitidos, se produciría la congestión de la red. Para evitarlo, el proceso de replicado de paquetes se realiza teniendo en cuenta la clase de tráfico y la dirección de la estación de destino. Del mismo modo, para la eliminación de paquetes duplicados se tiene en cuenta la clase de tráfico, la dirección de destino y los instantes de tiempo en los que se reciben los paquetes. Solo es imprescindible replicar el tráfico crítico. Las clases de tráfico no crítico y *best effort* pueden transmitirse normalmente sin necesidad de replicado.

2.2.5. IEEE 802.1Qcc: Stream Reservation Protocol (SRP) Enhancements

El Stream Reservation Protocol (SRP) definido en el estándar IEEE 802.1Qat del protocolo AVB (Audio Video Bridging) permite reservar un canal de comunicación para un flujo de datos, entre una estación fuente (*Talker*) y una estación destino (*Listener*), con un determinado ancho de banda, un retardo y un *jitter* máximos.

En SRP, cuando un *Talker* necesita transmitir una determinada clase de tráfico hacia un *Listener*, el *Talker* solicita un canal de comunicación con el *Listener* indicando los parámetros de QoS del canal (ancho de banda, retardo y *jitter*). Los parámetros de QoS se envían a través de los *switches* de la red, y cada *switch* reserva los recursos necesarios para cumplir con los requisitos de QoS, acumulando el retardo de los enlaces entre *switches* y el *jitter* hasta llegar al *Listener*.

El principal problema de SRP es que funciona de forma descentralizada. Cuando existen muchos flujos de datos entre diversos *Talkers* y *Listeners*, si aumenta el tráfico, entonces puede ser necesario reconfigurar los *switches* para cambiar enlaces y evitar la congestión de la red. Esto puede ser complicado, si no imposible, utilizando un mecanismo descentralizado como el de SRP.

El estándar IEEE 802.1Qcc introduce mejoras significativas respecto a SRP utilizando un mecanismo de configuración centralizado. En la figura 11 se muestra la arquitectura del sistema propuesto por el estándar IEEE 802.1Qcc para configurar una red TSN. El sistema está compuesto por los siguientes elementos:

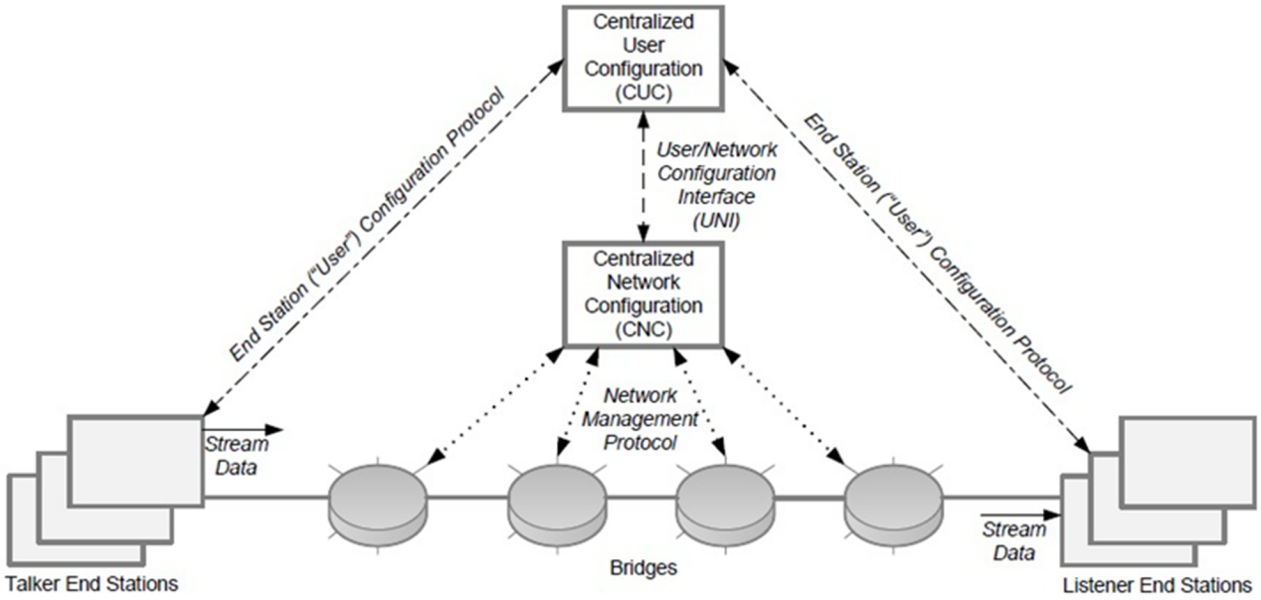
- **Centralized User Configuration** (CUC): es una herramienta software utilizada para configurar el sistema de control y automatización industrial. Permite enviar parámetros de configuración y programar los dispositivos de control (PLC), sensores, actuadores, etc. que son los *Talkers* y *Listeners* del sistema.

- **Centralized Network Configuration** (CNC): es el elemento que realiza la configuración de la red TSN. El CNC descubre la topología de la red e identifica las características relevantes de cada componente de la red TSN y de la red en su conjunto: latencias de los *switches*, velocidades y retardos de propagación de los enlaces, precisión de la sincronización de relojes, etc.

El CUC se comunica con el CNC para enviarle peticiones de canales de comunicación para flujos de datos entre *Talkers* y *Listeners*. La comunicación entre CUC y CNC se realiza a través de la User/Network Configuration Interface (UNI) mediante una RESTful service API.

En cada petición de canal enviada al CNC se incluyen los requisitos del canal solicitado: direcciones IP de *Talker* y *Listener*, tiempos de ciclo, ancho de banda, retardo, *jitter*, etc. El CNC selecciona los *switches*, los enlaces entre *switches*, y calcula los *schedules* necesarios para cumplir con los requisitos solicitados por el CUC. A continuación, el CNC distribuye los *schedules* a los diversos elementos de la red TSN. La comunicación entre el CNC y los elementos de la red TSN se realiza a través del protocolo NETCONF o RESTCONF.

Figura 11. Configuración centralizada en TSN (IEEE 802.1Qcc)



Fuente: extraído de «Time-Sensitive-Networking-for-Industrial-Applications-Volker-Goller.pdf»

Entre los parámetros enviados desde el CNC a los elementos de la red TSN se incluyen los siguientes: la configuración de las listas de control de puertas de cada *switch* (Gate Control List), según el estándar IEEE 802.1Qbv; la configuración del mecanismo de Frame Preemption, según el estándar IEEE 802.1Qbu; y la configuración del mecanismo Frame Replication and Elimination, según el estándar IEEE 802.1CB.

La configuración centralizada definida en el estándar IEEE 802.1Qcc permite aumentar el número de canales y flujos de datos, y reducir los tiempos de configuración de canal respecto a SRP.

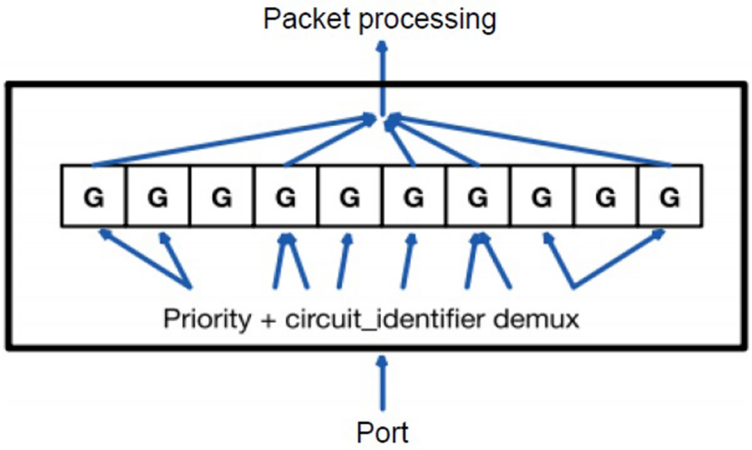
2.2.6. IEEE 802.1Qci: Ingress Policing

El estándar IEEE 802.1Qi define un mecanismo de Ingress Policing cuyo objetivo es evitar que se produzcan sobrecargas de tráfico en los nodos de la red. Para ello, se realiza un filtrado del tráfico de cada flujo de datos (*stream*) utilizando una puerta de entrada en cada *stream*. Cada puerta de entrada se encarga de dejar pasar o bloquear un flujo de datos de acuerdo con una función de *policing* específica. Entre las funciones de *policing* destacan las siguientes:

- Dejar pasar *streams* solo en intervalos de tiempo predefinidos.
- Dejar pasar *streams* solo a través de puertos de entrada específicos.
- Dejar pasar ráfagas de datos con un tamaño máximo.
- Algoritmo Leaky Bucket; chequea que los *streams* están dentro de unos límites de ancho de banda y de tamaño máximo de ráfaga de datos.

En la figura 12 se representan las puertas de entrada utilizadas para implementar el mecanismo de Ingress Policing.

Figura 12. Ingress Policing: filtrado del tráfico mediante puertas de entrada



De manera simplificada se podría decir que cada *Talker* tiene un contrato con un *Listener*. El contrato está definido por las funciones de *policing*, y las puertas de entrada se encargan de hacer que el contrato se cumpla. El mecanismo de Ingress Policing es especialmente útil para asegurar la robustez y fiabilidad de la red frente a ataques maliciosos en los flujos de datos.

2.2.7. IEEE 802.1Qch: Cyclic Queuing and Forwarding

El estándar IEEE 802.1Qch define un método de *traffic shapping*, denominado Cyclic Queuing and Forwarding (CQF), que permite la transmisión de tráfico con retardos deterministas. Como su nombre indica, el principio de funcionamiento de CQF es que los paquetes de datos se transmiten y se encolan en los *switches* de la red de manera cíclica.

El tiempo se divide en intervalos de tiempo sucesivos *i*, *i+1*, *i+2*, ..., *i+N*, siendo cada intervalo de duración *d*. En un flujo de datos que atraviesa los enlaces entre 3 *switches* adyacentes (A, B y C), los paquetes de datos se transmiten del *switch* A al B durante el intervalo i, del *switch* B al C en el intervalo i+1, y así sucesivamente hasta llegar a la estación de destino. Se asume que todos los *switches* y estaciones de la red están sincronizados.

Los paquetes transmitidos por el *switch* A durante el intervalo i son transmitidos por el *switch* B en el intervalo i+1, y por el *switch* C en el intervalo i+2. Por tanto, si la estación destino estuviera conectada al *switch* C, el retardo máximo en la transmisión de paquetes sería de 3 intervalos de tiempo de duración d.

El retardo introducido por los *switches* de la red es determinista y está definido por el tiempo de ciclo y el número de satos entre estación fuente y destino. Además, el retardo no varía al introducir interferencias de otros tipos de tráfico no crítico (*best effort*).

CQF se implementa configurando una combinación de los siguientes mecanismos introducidos en las secciones anteriores:

- Mecanismo de control de las puertas de entrada de los flujos de datos, definido en el estándar IEEE 802.1Qci (Per-Stream Filtering and Policing).

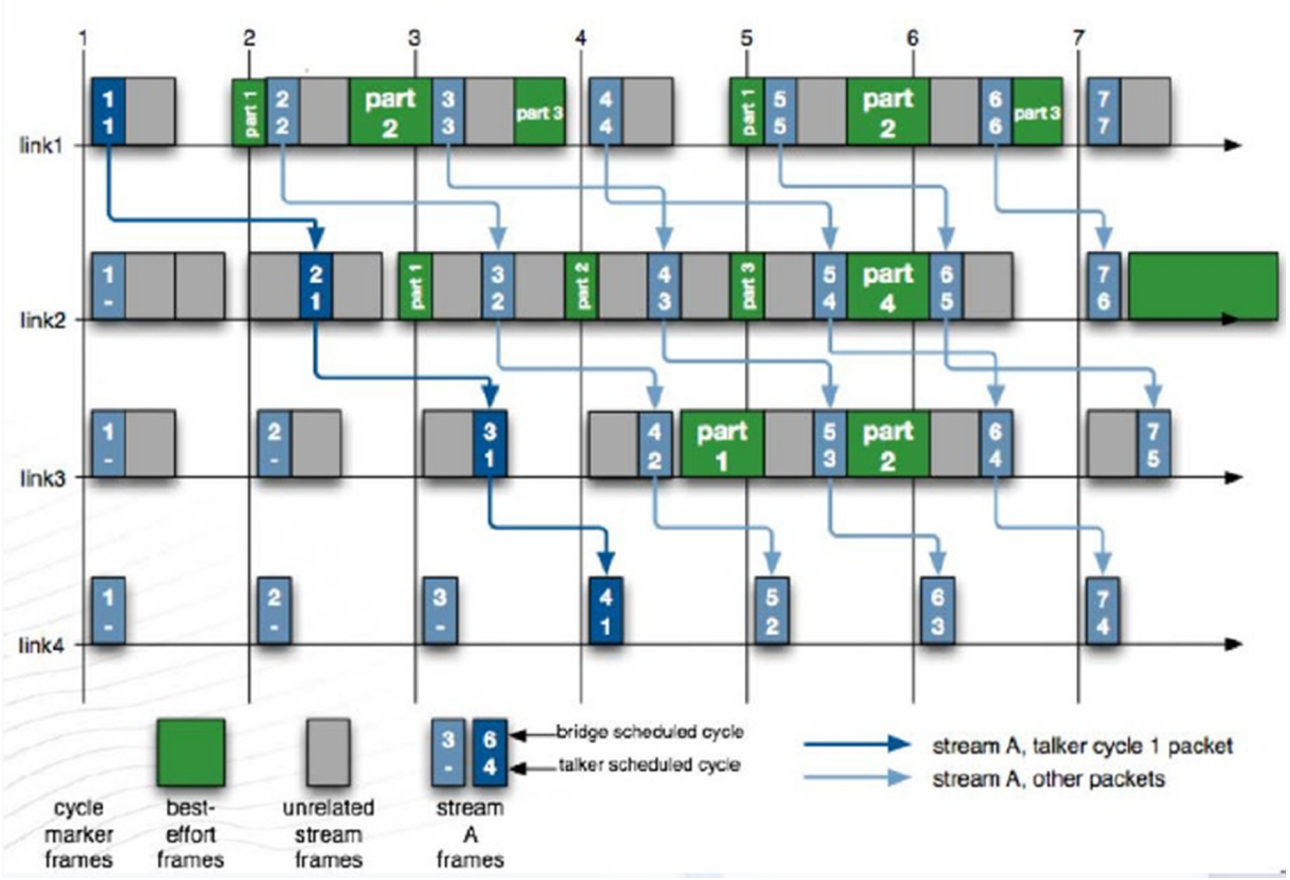
- Mecanismo de *scheduling* de las diferentes clases de tráfico que atraviesan cada *switch*, definido en el estándar IEEE 802.1Qbv (Time-Aware Traffic Shaping).

- Mecanismo de Frame Preemption definido en el estándar IEEE 802.1Qbu.

En la figura 13 se muestra un ejemplo de funcionamiento del mecanismo CQF combinado con el algoritmo de Frame Preemption en una red con 4 saltos (o enlaces) de un flujo de datos (*stream*) entre una estación fuente (*Talker*) y una estación destino (*Listener*). En el eje horizontal se muestra el tiempo, dividido en 7 ciclos o intervalos iguales, y en el eje vertical se muestran los 4 enlaces (*links*).

Los paquetes interferentes que pertenecen a otro flujo de datos están marcados en color gris. Los paquetes pertenecientes al flujo de datos entre la estación fuente y la estación destino están marcados de la forma siguiente. Los paquetes de tráfico crítico están marcados en color azul y los paquetes no críticos (*best effort*) están marcados en color verde.

Figura 13. Ejemplo de funcionamiento del mecanismo Cyclic Queuing and Forwarding (IEEE 802.1Qch) combinado con Frame Preemption



Fuente: IEEE 802.1Qbu

En cada paquete de tráfico crítico representado en la figura 13 se incluyen dos números. El número inferior indica el número de ciclo (del 1 al 7) en el que el *Talker* envía el paquete. El número superior indica el número de ciclo en el que un *switch* reenvía el paquete.

Como puede observarse en la figura 13, todos los paquetes de tráfico crítico llegan al *Listener* con un retardo determinista de 4 ciclos. Por ejemplo, el paquete que sale del *Talker* en el ciclo 1 llega al *Listener* en el ciclo 4, el que sale en el ciclo 2 llega en el ciclo 5, etc.

Por otro lado, los paquetes *best effort* son divididos en paquetes más cortos, según el mecanismo de Frame Preemption (IEEE 802.1Qbu) para evitar introducir retardos indeseados y aleatorios en los paquetes de tráfico crítico.

Además, en los enlaces intermedios (2 y 3), en los ciclos 2 y 3, se puede observar que los paquetes de datos sufren un pequeño retardo debido a la existencia de paquetes de otros flujos (en gris), pero dicho retardo finalmente se elimina en los ciclos del último enlace (4). Esto se realiza gracias a la aplicación del mecanismo de Time-Aware Traffic Shaping definido en el estándar IEEE 802.1Qbv.

2.2.8. IEEE 802.1Qca - Path Control and Reservation

El estándar IEEE 802.1Qca define los mecanismos que permiten la reserva de ancho de banda, la redundancia y el descubrimiento de caminos (*paths*) para los flujos de datos a partir de diversas topologías de red.

La selección de caminos se realiza aplicando reglas como la velocidad de transmisión de los enlaces entre *switches*, la fiabilidad de los enlaces y los retardos acumulados entre *Talker* y *Listener*.

Los mecanismos definidos en el estándar IEEE 802.1Qca permiten el cálculo y asignación de múltiples caminos simultáneos para cada flujo de datos, facilitando la implementación del mecanismo de Frame Replication and Elimination definido en el estándar IEEE 802.1CB.

2.3. Implementaciones

Mientras que el proceso de estandarización de la tecnología TSN avanza, diversos fabricantes ya han realizado las primeras implementaciones de las nuevas funcionalidades introducidas por TSN. Se estima que el conjunto de estándares del IEEE TSN estará acabado a finales de 2017. De momento, los primeros prototipos y el primer *switch* TSN comercial han demostrado que es posible conseguir un grado de determinismo en el retardo muy superior al de los sistemas Industrial Ethernet propietarios de clase C.

En la actualidad, existen 11 empresas líderes en el sector de la automatización industrial y en el sector IT que han anunciado sus nuevos productos con OPC UA sobre TSN como solución de comunicación industrial. La mayoría de fabricantes han anunciado que lanzarán productos basados en TSN a lo largo de 2017.

En el sector de la automoción, ya existen automóviles equipados con el primer *switch* con funcionalidades TSN. Además, los primeros vehículos autónomos equipados con tecnología TSN empezarán a comercializarse a lo largo de 2017.

El Industrial Internet Consortium (IIC) es una organización internacional que promueve el crecimiento del Industrial Internet of Things (IIoT). El IIC está formado por un conjunto de empresas líderes en automatización, entre las cuales han desarrollado un testbed TSN equipado con sus productos TSN comerciales. El objetivo del testbed es evaluar el funcionamiento de la tecnología TSN en todo tipo de aplicaciones industriales.

En la figura 14 se muestra la topología del testbed TSN del IIC, indicando el nombre de los fabricantes de los productos que forman el testbed, y las funciones que realiza cada uno de ellos. En la figura 15 se muestra una fotografía de una parte del testbed TSN del IIC.

Entre los productos TSN que constituyen el testbed del IIC, destacan: controlador de robot Kuka; controlador de movimiento Rexroth; generador de tráfico Ixia; controlador Schneider; controlador de movimiento BR Automation; controlador, módulo de visión y módulos de entrada/salida National Instruments; dispositivos de campo (entradas/salidas) Innovasic; *switches* CISCO y TTTech.

Figura 14. Topología del testbed TSN del Industrial Internet Consortium

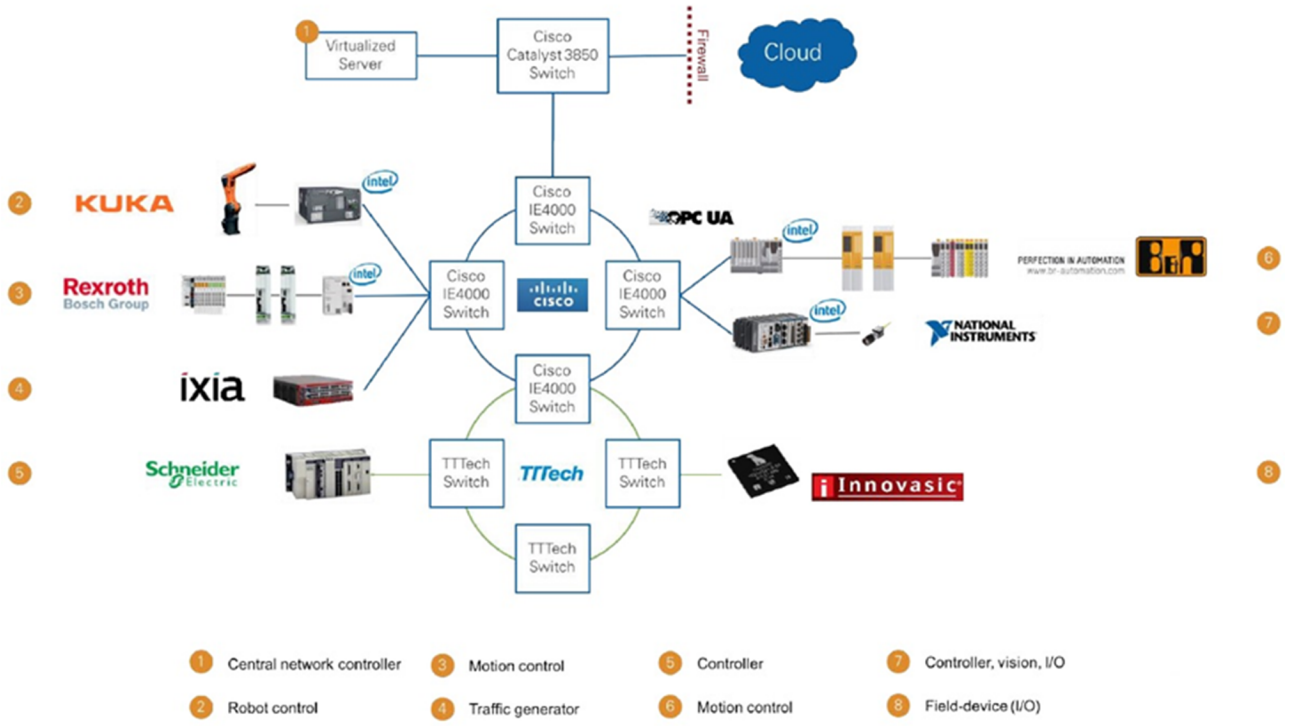
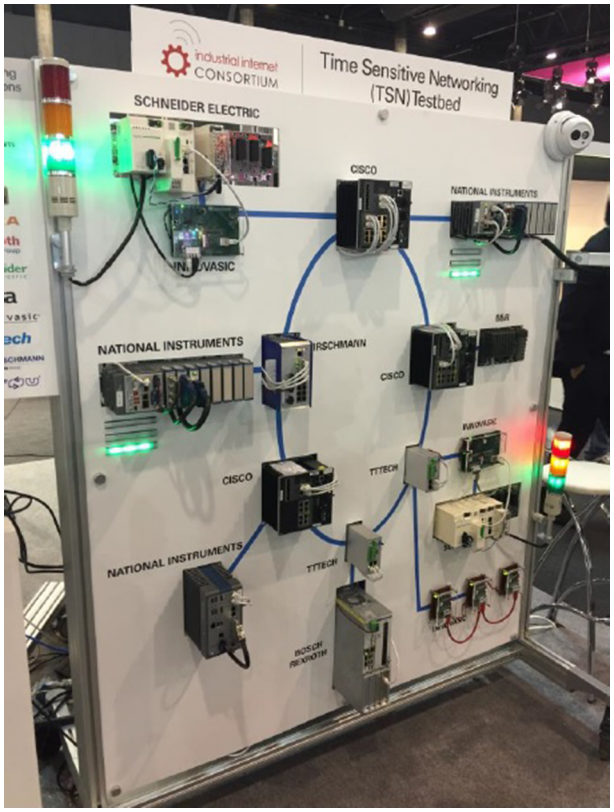


Figura 15. Parte del testbed TSN del Industrial Internet Consortium



3. Redes *time-sensitive* inalámbricas

3.1. Introducción

La utilización de las redes inalámbricas en los procesos de automatización industrial facilita la instalación y el movimiento de las máquinas y los dispositivos de campo, y reduce considerablemente los costes de cableado, puesta en marcha y de mantenimiento. Por ejemplo, en la industria del gas y del petróleo, suelen utilizarse miles de sensores de temperatura, presión, etc. para monitorizar las tuberías de combustible. La instalación y el mantenimiento del cableado de comunicaciones y alimentación de los sensores pueden costar entre 250 y 2.500 euros por metro, y constituye una gran parte del coste total del sistema de monitorización. Por tanto, el uso de las redes inalámbricas en este tipo de aplicaciones industriales está perfectamente justificado.

Para que las redes inalámbricas puedan utilizarse en los procesos de automatización industrial, deben cumplir los siguientes requisitos:

- Bajo retardo y bajo *jitter*.
- Retardo determinista.
- Alta fiabilidad en la transmisión de datos.
- Bajo consumo de los nodos alimentados con baterías y tiempo de vida de la red largo.

El primer protocolo para redes inalámbricas diseñado para cumplir los requisitos anteriores fue el Time Synchronized Mesh Protocol (TSMP). El TSMP fue desarrollado y comercializado por Dust Networks en 2006. Es un protocolo para redes con topología en malla (*mesh*) que establece un *schedule* para la transferencia de paquetes entre los nodos de la red, asegurando un valor de retardo determinista extremo a extremo. En TSMP, los nodos de la red están sincronizados, y reducen el consumo energético mediante la activación y desactivación de la radio durante los *slots* que corresponda según sea el *schedule* establecido.

Además, TSMP utiliza el mecanismo de Frequency Channel Hopping (FCH, saltos de canal frecuencial) para reducir las pérdidas de paquetes debido a interferencias y a los efectos del *fading* (o atenuación) producido por la propagación multi-camino. Este tipo de atenuación es muy habitual en comunicaciones radio y se produce cuando al nodo receptor llegan múltiples «ecos» de la señal deseada. La señal transmitida vía radio por el nodo receptor puede

atravesar diferentes caminos y rebotar en diferentes elementos del entorno hasta llegar al receptor. Esto suele introducir distintos desfases en los «ecos» recibidos y puede provocar la atenuación de la señal deseada. Los efectos del multi-camino son muy sensibles a la frecuencia y a pequeños cambios en el entorno, que pueden hacer variar notablemente los niveles en la señal recibida, degradando la calidad del enlace radio. Al utilizar diversos canales frecuenciales, la técnica FCH permite combatir los efectos de la propagación multi-camino ya que los patrones de los «ecos» que pueden atenuar o amplificar la señal recibida son distintos para diferentes frecuencias.

Con TSMP se consiguió una fiabilidad del 99,999 % en la entrega de paquetes de extremo a extremo, y diez años de tiempo de vida de la red, convirtiéndolo en una alternativa real frente a las soluciones basadas en redes cableadas.

Más adelante, los conceptos de TSMP fueron introducidos en los estándares WirelessHART (2008) e ISA100.11A (2011), que fueron diseñados específicamente para aplicaciones de monitorización industrial y procesos de control. En 2012, los conceptos del protocolo TSMP se incluyeron en el estándar IEEE802.15.4e.

Mientras que los estándares WirelessHART e ISA100.11A definen una pila de protocolos completa, el estándar IEEE802.15.4e solo define la capa física y tres capas de control de acceso al medio (MAC) distintas:

- Low Latency Deterministic Network (LLDN),
- Time-Slotted Channel Hopping (TSCH), y
- Deterministic and Synchronous Multi-channel Extension (DSME).

De las tres capas MAC definidas en el estándar IEEE802.15.4e, TSCH es la que facilita la operación en una red *multi-hop* (multi-salto), y además, mediante la técnica de FCH, proporciona robustez frente a interferencias y ante la atenuación producida por la propagación multi-camino.

En 2013 se creó el IETF 6TiSCH Working Group, cuyo objetivo era desarrollar una arquitectura, denominada 6TiSCH, que incluyera IPv6 encima de la capa de control de acceso TSCH definida en el estándar IEEE 802.15.4e.

Igual que las TSN cableadas, la arquitectura 6TiSCH permite combinar la transmisión de tráfico no crítico (*best effort*) con tráfico crítico (*time-sensitive*), sin degradar el retardo determinista necesario en el tráfico crítico. Para ello, 6TiSCH permite incorporar dos mecanismos que permiten realizar la reserva de ancho de banda para los flujos de datos. Para los flujos de datos con tráfico crítico, se utiliza un mecanismo de *routing* y *scheduling* centralizado, similar al empleado en TSN y en las Software Defined Networks (SDN). Por otro lado,

para los flujos de datos con tráfico no crítico, se utiliza un mecanismo de *routing* y *scheduling* distribuido (o descentralizado) basado en el protocolo RPL (en inglés, Routing Protocol for Low-Power and Lossy Networks).

Esta sección está dedicada al estudio y análisis de la arquitectura 6TiSCH como tecnología para redes inalámbricas deterministas en aplicaciones industriales. La sección 3.2 presenta los estándares que constituyen la tecnología 6TiSCH y describe sus características principales. Finalmente, en la sección 3.3 se presentan las soluciones comerciales existentes para implementar redes basadas en la tecnología 6TiSCH.

3.2. Estándares IEEE-IETF 6TiSCH

La pila de protocolos 6TiSCH se muestra en la figura 16. La arquitectura 6TiSCH está compuesta por un conjunto de estándares desarrollados por el IEEE y el IETF, que proporcionan conectividad IPv6 de extremo a extremo, asegurando valores de retardo y *jitter* deterministas, robustez y bajo consumo de los nodos de la red. Estas características han convertido a 6TiSCH en una buena alternativa para aplicaciones de control y monitorización de procesos industriales.

Como puede observarse, entre las aplicaciones y la capa MAC (TSCH) se encuentran los siguientes estándares:

- **IETF CoAP:** es un protocolo de la capa de aplicación que permite que nodos de bajo consumo de una red inalámbrica funcionen como pequeños servidores web con los que cualquier máquina puede interactuar a través de internet.

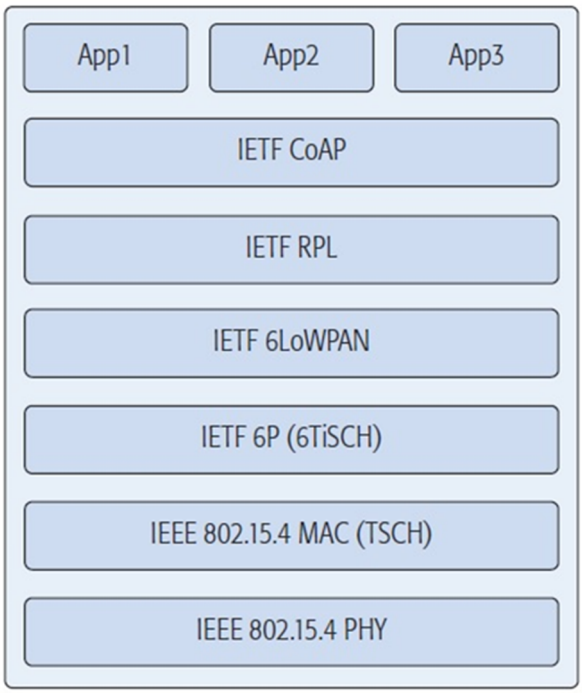
- **IETF RPL (Routing Protocol for Low-Power and Lossy Networks):** es un protocolo de *routing* que permite enrutar paquetes a través de una red *multi-hop*.

- **IETF 6LoWPAN:** es una capa de adaptación que comprime y fragmenta las largas cabeceras de los paquetes IPv6 para que quepan en paquetes IEEE 802.15.4.

- **IETF 6top Protocol (6P):** es una capa de adaptación que permite la integración entre la capa 6LoWPAN y la capa IEEE 802.15.4e MAC (TSCH) a través de interfaces de datos y gestión, y organiza la transmisión de paquetes IPv6 sobre el protocolo TSCH.

En los apartados siguientes se describen las principales características de todos estos estándares IEEE y IETF que componen la arquitectura 6TiSCH.

Figura 16. Pila del protocolo 6TiSCH



Fuente: extraído de Watteyne, Tuset-Peiro, Vilajosana, Pollin y Krishnamachari (2017)

3.2.1. IEEE 802.15.4 PHY

El estándar IEEE 802.15.4 define la capa física para comunicaciones inalámbricas de bajo consumo entre dispositivos alimentados con baterías en redes denominadas Low Rate Wireless Personal Area Networks (LR-WPAN).

El estándar IEEE 802.15.4 ofrece tres bandas frecuencias libres de licencia (868 MHz, 915 MHz y 2.4 GHz) con un total de 27 canales frecuenciales repartidos de la forma siguiente: 16 canales en la banda de 2.4 GHz, 10 canales en la banda de 915 MHz, y un canal en la banda de 868 MHz.

En la banda de 2.4 GHz se utiliza Direct Sequence Spread Spectrum (DSSS) y modulación O-QPSK (Orthogonal Quadrature Phase Shift Keying). En las bandas de 915 MHz y 868 MHz se utiliza DSSS y modulación BPSK (Binary Phase Shift Keying). La arquitectura IETF 6TiSCH funciona en la banda de 2.4 GHz.

La capa física del estándar IEEE 802.15.4 permite configurar potencias de transmisión entre 0 y 10 dBm, una velocidad de transmisión de datos de hasta 250 kbps en la banda de 2.4 GHz, y un tamaño máximo de *payload* de 127 bytes.

Una de las principales limitaciones del estándar IEEE 802.15.4 es que el tamaño de la *payload* está limitado 127 bytes, que es mucho menor que el tamaño máximo de los paquetes Ethernet (1500 bytes), y que el tamaño de la Maximum Transmission Unit (MTU) en IPv6 (de 1280 bytes a 64 kbytes).

En 2007, el IETF creó el 6LowPAN Working Group, que definió un mecanismo de fragmentación y compactación para transportar paquetes IPv6 sobre paquetes IEEE 802.15.4. El funcionamiento de la capa IETF 6LowPAN se describe más adelante en esta sección.

3.2.2. IEEE 802.15.4e MAC (TSCH)

El estándar IEEE 802.15.4e define el protocolo de control de acceso al medio TSCH utilizado en la arquitectura 6TiSCH. En TSCH, el tiempo está divido en ranuras temporales o *slots*. Los *slots* se agrupan en tramas o *frames* (típicamente de 5 a 1.000 *slots* por trama), que se repiten de forma periódica. La duración de un *slot* suele ser de 10 ms, tiempo suficiente para transmitir el paquete de datos de tamaño máximo y para que el nodo receptor envíe un paquete de *Acknowledgment* (ACK). En un *slot*, un nodo puede transmitir un paquete de datos, recibir, o dormir (*sleep*). Cuando un nodo está en modo *sleep*, su radio está apagada.

En una red TSCH se utiliza un *schedule* para coordinar la comunicación entre los nodos de la red. El *schedule* indica a cada nodo la operación que debe realizar en cada *slot*: transmitir, recibir o dormir. Además, en aquellos *slots* en los que se realice la transmisión o recepción de datos, el *schedule* determina el nodo vecino con el que debe comunicarse el nodo, y el número de canal frecuencial (o *channel offset*). El *schedule* debe asegurar que existen suficientes oportunidades de comunicación entre los nodos de la red para satisfacer los requisitos de retardo, fiabilidad y *throughput*.

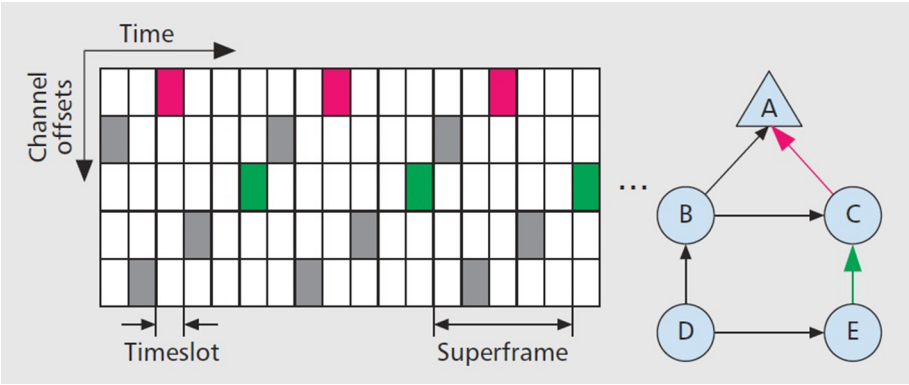
Cuando llega un paquete a la capa MAC procedente de las capas superiores, el paquete se almacena en una cola de transmisión. En cada *slot*, la capa MAC chequea si hay un paquete en la cola con destino a un nodo vecino asociado con ese *slot*. Si lo hay, transmite el paquete y espera un ACK. Al recibir el ACK, elimina el paquete de la cola; en caso contrario, el paquete se mantiene en la cola para una retransmisión futura.

En cada *slot* de recepción, el nodo pone en marcha la radio y espera a recibir un nuevo paquete. Si recibe un paquete destinado a él, envía un ACK, apaga la radio y pasa el paquete a la capa superior. Si no recibe ningún paquete pasado un tiempo, el nodo vuelve a modo *sleep*.

La figura 17 muestra un ejemplo de *schedule* de comunicaciones en una red TSCH. El tiempo está divido en tramas de 5 *slots* que se repiten periódicamente. Cada fila del *schedule* representa un canal frecuencial (*channel offset*). En un mismo *slot* podrían producirse comunicaciones simultáneas entre múltiples parejas de nodos de la red en canales frecuenciales diferentes. En el ejemplo, cuando el nodo E necesita transmitir un paquete al nodo A, el nodo E espera al *slot* con una celda de color verde para enviar el paquete al nodo C. Finalmente, el nodo C espera al *slot* con una celda en color rojo para enviar el paquete al

nodo A. Si alguna de las dos transmisiones falla, es decir, el nodo transmisor no recibe un ACK, entonces el transmisor vuelve a intentarlo en el mismo *slot* de la trama siguiente.

Figura 17. Ejemplo de *schedule* TSCH en una red basada en 6TiSCH



Fuente: extraído de Dujovne, Watteyne, Vilajosana y Thubert (2014)

A continuación, se describen los mecanismos utilizados para crear y mantener el *schedule*, sincronizar los nodos de la red, realizar los saltos de canal frecuencial, y formar la topología de la red mediante el descubrimiento de nodos vecinos.

Scheduling

El *schedule* del protocolo TSCH debe crearse para asegurar que cuando un nodo A transmite a un nodo B vecino, B está escuchando el canal para recibir paquetes procedentes de A. De forma similar, si A deja de ser vecino de B (porque A ha sido desplazado o apagado), entonces B no debería escuchar más para recibir paquetes de A.

Por tanto, el *schedule* debe ser creado en la puesta en marcha de la red y actualizado periódicamente para incluir o eliminar los enlaces que aparecen y desaparecen. El procedimiento de creación y mantenimiento del *schedule* se denomina *scheduling*, que se ejecuta en la capa 6top descrita en la sección 3.2.3.

Cada celda del *schedule* está definida por un número de *slot* en la trama TSCH y un número de canal frecuencial, y constituye una oportunidad de comunicación entre dos nodos vecinos. Por tanto, al añadir más celdas al *schedule*, aumenta el ancho de banda y reduce el retardo extremo a extremo, pero incrementa el consumo energético de los nodos porque deben permanecer activos (transmitiendo o recibiendo) durante más tiempo.

Sincronización

La sincronización de los nodos en una red TSCH es necesaria para mantener la conectividad con los nodos vecinos en una red basada en tramas divididas en *slots* y que no utiliza paquetes de *beacon*. Se definen dos mecanismos para permitir que un nodo se sincronice con la red: la sincronización basada en ACK, y la sincronización basada en trama.

- En la **sincronización basada en ACK**, el nodo receptor calcula la diferencia entre el tiempo de llegada esperado de un paquete y el tiempo de llegada real, y envía esa información en el ACK al nodo transmisor. Esto permite al nodo transmisor sincronizarse con el reloj del nodo receptor.

- En la **sincronización basada en paquete**, el nodo receptor calcula la diferencia temporal entre el tiempo de llegada esperado de un paquete y el tiempo de llegada real, y ajusta su propio reloj con la diferencia. Esto permite al nodo receptor sincronizar su reloj con el del nodo transmisor.

En funcionamiento normal, los nodos se resincronizan utilizando los paquetes que intercambian con otros nodos. Sin embargo, cuando no se comunican con otros nodos durante un intervalo de tiempo largo (30 segundos), entonces los nodos intercambian un paquete de datos vacío (mensajes «keep-alive») para resincronizarse. En una red TSCH, el tiempo se propaga desde el nodo coordinador. Cada nodo sincroniza su reloj con al menos un nodo vecino, y ofrece su valor de tiempo de red a todos sus nodos vecinos.

Frequency Channel Hopping

TSCH incorpora la técnica de Frequency Channel Hopping (FCH) en el *schedule* de la red. Tal como se comentó antes, FCH añade diversidad frecuencial para mitigar los efectos de las interferencias y de la atenuación producida por la propagación multi-camino. Además, el uso de diversos canales frecuenciales simultáneamente aumenta la capacidad de la red porque permite que varios nodos puedan transmitir a la vez en el mismo *slot* utilizando canales diferentes.

En el estándar IEEE 802.15.4e se definen un total de 16 canales frecuenciales en la banda de 2.4 GHz. En algunos casos se restringe el uso de determinados canales por cuestiones de coexistencia con otras redes que operen en la banda de 2.4 GHz. Si el número de *slots* por trama y el número de canales disponibles son pequeños, el FCH de TSCH permite utilizar canales distintos para el mismo *slot* en tramas sucesivas.

Formación de la red

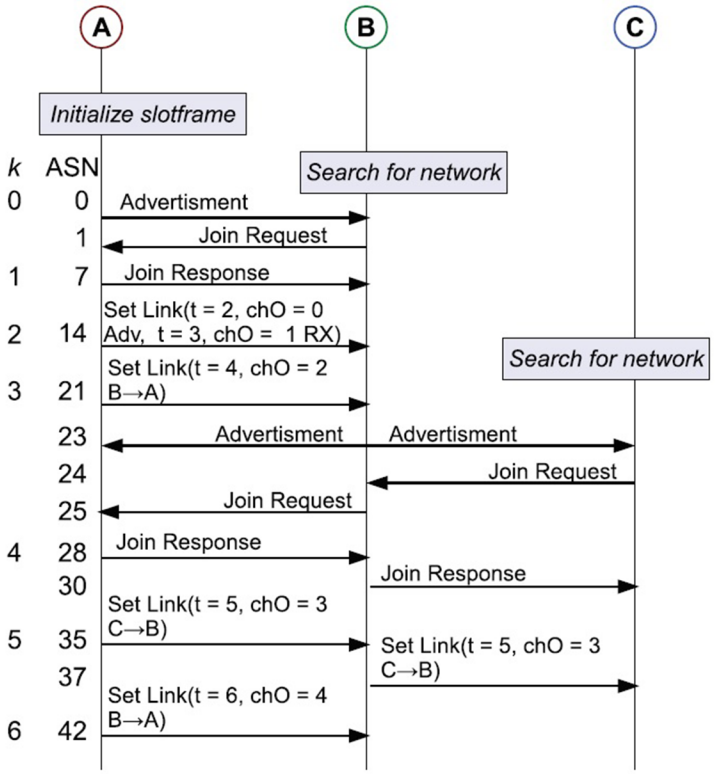
El proceso de formación de una red TSCH consta de dos fases: *advertising* y *joining*. Un nuevo nodo que intenta unirse a la red escucha el canal para recibir mensajes de comando *Advertisement*. Cuando el nodo recibe un mensaje de *Advertisement*, el nodo se une a la red mediante el envío de un mensaje de comando *Join Request* hacia el nodo que envió el mensaje de *Advertisement*.

En una red TSCH basada en un sistema de gestión centralizado, los mensajes *Join Request* se enrutan hacia el nodo «manager». Sin embargo, si se utiliza un sistema de gestión distribuido, los mensajes *Join Request* se procesan localmente en el nodo que los recibe. Cuando un nuevo nodo es aceptado en la red, el nodo que envió el mensaje de *Advertisement* activa al nuevo nodo mediante la configuración de las tramas y de los enlaces entre el nuevo nodo y el resto de nodos vecinos. Evidentemente, las tramas y los enlaces también pueden eliminarse y/o modificarse después de que un nodo se haya unido a la red.

La figura 18 muestra los mensajes intercambiados durante la fase de creación de una red TSCH en un escenario sencillo con tres nodos (A, B y C), usando tramas de siete *slots*, y un sistema de gestión centralizado. El nodo A realiza funciona como «manager» de la red. Es decir, define los enlaces en cada uno de los *slots* de las tramas para permitir la comunicación dedicada entre nodos y la comunicación *broadcast*.

En el ejemplo de la figura 18, para cada mensaje intercambiado entre los nodos de la red se indica el número de *slot* (ASN, Absolute Slot Number) en el cual se envió el mensaje. Además, se indica el inicio de cada trama con el número de trama k.

Figura 18. Mensajes intercambiados durante la creación de una red TSCH formada por tres nodos



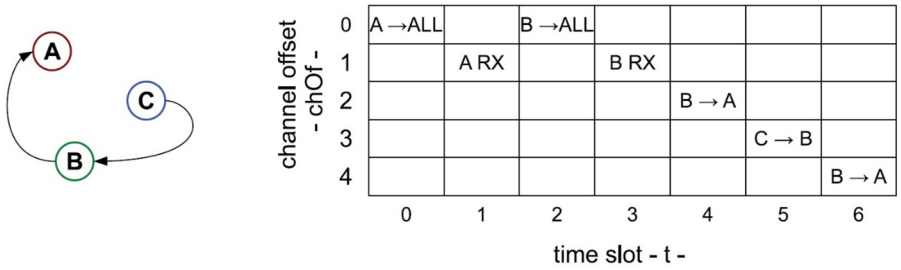
Fuente: extraído de Palattella y otros (2013)

El «manager» de la red (nodo A) inicia la primera trama (k = 0), con la que los otros nodos A y B deben sincronizarse. El nodo A se reserva los dos primeros *slots* para el envío de mensajes de *Advertisement* y recepción de mensajes *Join Request*. Cuando el nodo B recibe el mensaje de *Advertisement*, responde con un *Join Request*, y el nodo A responde con un *Join Response* y le asigna número de *slot* y de canal en la trama. Una vez que el nodo B ha entrado en la red, envía mensajes de *Advertisement*. Cuando el nodo C recibe el mensaje de *Advertisement* procedente del nodo B, le responde con un mensaje *Join Request*, que el nodo B reenvía hacia el nodo A, y este responde con un *Join Response* y le asigna número de *slot* y de canal en la trama. El nodo B reenvía los mensajes enviados por el nodo A al nodo C.

La figura 19 muestra los enlaces y los canales frecuenciales asignados en cada *slot* de las tramas TSCH configuradas durante el proceso de creación de la red. Como puede observarse, los *slots* 0 y 2 se utilizan para los enlaces de *broadcast* a través del canal 0, y los *slots* 1 y 3 se utilizan para recibir mensajes *Join Request* a través del canal 1. Para permitir el envío de datos desde los nodos B y C hacia el nodo A, los *slots* 4, 5 y 6 se reservan para los enlaces B→A, C→B y B→A, respectivamente, en los canales frecuenciales 2, 3 y 4.

Una vez que todos los nodos se han unido a la red, el «manager» deja de recibir mensajes *Join Request* durante un determinado intervalo de tiempo, a partir del cual puede decidir finalizar el procedimiento de *advertising*, que posteriormente puede volver a activar periódicamente para comprobar si existen nuevos nodos que esperan para conectarse a la red.

Figura 19. *Schedule* de la trama de 7 *slots* definido durante el proceso de creación de la red



Fuente: extraído de Palattella y otros (2013)

3.2.3. IETF 6top Protocol (6P)

El protocolo TSCH definido en el estándar IEEE 802.15.4e indica cómo debe ejecutarse el *schedule* TSCH, pero no especifica cómo crear y mantener el *schedule*. Esta operación se realiza en la capa superior, IETF 6top Protocol.

El estándar IETF 6TiSCH Operation Sublayer (6top) Protocol (6P) ha sido diseñado para combinar mecanismos de *scheduling* de tipo centralizado y distribuido.

- En ***scheduling* centralizado**, un nodo «manager», o Path Computation Element (PCE), es responsable de crear y mantener el *schedule* de la red. Cada nodo envía periódicamente al «manager» información con la lista de nodos que puede escuchar y del tráfico que genera. A partir de la información de los nodos vecinos, el «manager» construye un gráfico de conectividad. Y a partir de del volumen de tráfico generado en cada nodo, el «manager» asigna *slots* a los diferentes enlaces existentes. Una vez creado el *schedule*, el «manager» se lo envía a cada nodo, que siguen el *schedule*. Cuando hay un cambio en los enlaces, el «manager» actualiza el *schedule* e informa a los nodos afectados. El *scheduling* centralizado es muy eficiente con redes estáticas. Sin embargo, cuando la topología de la red cambia constantemente, entonces es preferible el *scheduling* distribuido.

- En ***scheduling* distribuido**, los nodos deciden localmente, negociando con los nodos vecinos, los enlaces que deben ser asignados en cada *slot* del *schedule*, sin necesidad de un nodo que realice las funciones de «manager». Es decir, los nodos vecinos se encargan de negociar la incorporación y la eliminación de una o múltiples celdas en el *schedule* a partir de información local o utilizando un protocolo de reserva de recursos distri-

buido similar al Resource Reservation Protocol (RSVP) o al Next Steps In Signaling (NSIS).

En 6top, cada una de las celdas del *schedule* puede ser considerada de dos tipos: «hard» o «soft». Las celdas de tipo «hard» no pueden ser reasignadas por 6top, es decir, 6top no les puede asignar otro número de *slot* ni de canal frecuencial. Las celdas de tipo «hard» son típicamente configuradas por un PCE, y una vez añadidas, solo el PCE puede moverlas o eliminarlas dentro del *schedule* TSCH.

Las celdas de tipo «soft» pueden ser reasignadas por 6top dinámicamente según se lo indique un módulo o función de *scheduling* distribuido, que es quien determina el número de celdas a asignar a un nodo vecino específico según sea su requisito de ancho de banda.

Un ejemplo de función de *scheduling* distribuido es el denominado On-The-Fly Bandwidth Reservation (OTF), que se encarga de determinar el número de celdas que hay que añadir o eliminar, dependiendo de la carga de tráfico real en la red, y 6top es quien realiza los cambios en el *schedule*. Es decir, 6top selecciona los números de *slot* y los números de canal de cada celda a través de un procedimiento de negociación de celdas «soft» definido en el estándar IETF 6top Protocol.

Además, 6top ejecuta un procedimiento de monitorización del rendimiento de las celdas asignadas a cada nodo y, si alguna celda tiene un rendimiento peor que las otras celdas del mismo nodo vecino, entonces 6top reasigna dicha celda a otro número de *slot* y número de canal frecuencial dentro del *schedule* TSCH.

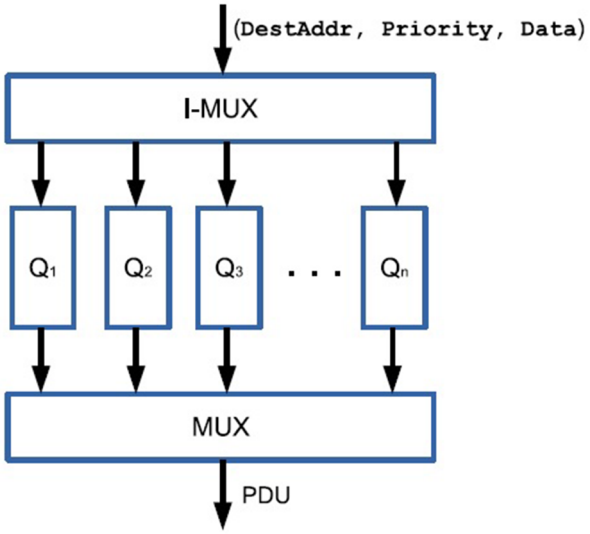
Una vez que se ha establecido el *schedule* TSCH, 6top se encarga de enviar a la capa TSCH el tráfico procedente de la capa superior (6LowPAN). 6top incluye un modelo de colas que permite satisfacer los requisitos de QoS de los diferentes tipos de tráfico procedentes de la capa 6LowPAN. El modelo de colas de 6top está representado en la figura 20. El número de colas es configurable. Cada cola puede ser para tráfico *broadcast* o *unicast*, y tiene una prioridad asignada.

Cuando 6top recibe un paquete, el módulo I-MUX (Inverse Multiplexer) selecciona una cola para almacenar el paquete según sea la prioridad del paquete. El módulo MUX (Multiplexer) se ejecuta en cada celda del *schedule* TSCH en la que se haya previsto una transmisión. MUX recorre las colas utilizando la dirección del nodo vecino al cual hay que transmitir en cada celda, para comprobar si existe algún paquete para transmitir a ese nodo vecino. Si encuentra un paquete, se lo pasa a la capa TSCH para que realice la transmisión.

6top mantiene información con las estadísticas del rendimiento de cada celda. Cuando se usa un *schedule* centralizado, esta información se envía periódicamente al PCE, que continuamente adapta el *schedule* y envía actualizaciones a los nodos cuando sea necesario.

6top proporciona un conjunto de comandos para soportar mecanismos de *scheduling* centralizado, distribuido e híbridos. Estos comandos pueden ser utilizados por las capas superiores, por ejemplo, para añadir o eliminar celdas «soft». Además, las estadísticas de rendimiento de los enlaces también son accesibles desde las capas superiores a 6top, lo cual permite, por ejemplo, optimizar el enrutado de paquetes según sean los requisitos de QoS de las aplicaciones.

Figura 20. Modelo de colas de 6top



Fuente: extraído de Palattella y otros (2016)

El estándar IETF 6top define un protocolo, denominado 6top Protocol (6P), que permite la negociación entre nodos vecinos para la reasignación de celdas «soft» en el *schedule*. El protocolo 6P define un conjunto de mensajes para añadir o eliminar *slots* (mensajes ADD, DELETE, CLEAR) y para intercambiar información sobre el *schedule* (mensajes COUNT y LIST). Por ejemplo, cuando la función de *scheduling* solicita a 6top añadir un determinado número de celdas en el *schedule* para satisfacer los requisitos de ancho de banda, la capa 6top ejecuta los siguientes pasos:

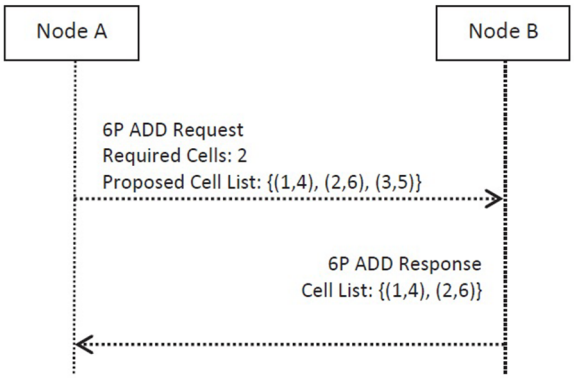
1) Comprueba si en el *schedule* TSCH hay suficientes celdas libres para satisfacer la petición de la función de *scheduling*.

2) Si hay suficientes celdas libres, entonces 6top envía un mensaje *ADD Request* al nodo vecino con la lista de celdas propuestas tal como se representa en la figura 21.

3) El nodo vecino recibe el mensaje *ADD Request* y comprueba si en su *schedule* TSCH están libres las celdas propuestas.

4) El nodo vecino envía un mensaje *ADD Response* indicando las celdas seleccionadas de la lista de celdas propuestas y añade las nuevas celdas a su *schedule*.

Figura 21. Intercambio de mensajes 6P intercambiados entre dos nodos vecinos para añadir nuevas celdas en su *schedule* TSCH



Fuente: extraído de Thomas, Shalu, Daniel, Anand y Hegde (2017)

3.2.4. IETF 6LowPAN

En el estándar IETF IPv6 over Low Power WPAN (6LoWPAN) se define un mecanismo que permite transmitir paquetes IPv6 sobre redes Low Power Wireless Pico Area Networks (LPWAN) IEEE 802.15.4. Las redes LPWAN se caracterizan por tener un tamaño de paquetes pequeño (*payload* de 127 bytes), dispositivos alimentados con baterías que pasan largos periodos de tiempo en modo *sleep* para reducir consumo energético, topologías en estrella y malla, etc.

Por tanto, la adopción de IPv6 sobre una red LPWAN no es inmediata y requiere la inclusión de una capa de adaptación. Por ejemplo, un paquete IPV6 tiene un tamaño de MTU mínimo de 1.280 bytes, que es demasiado grande para meterlo en un paquete IEEE 802.15.4. Además, el *overhead* introducido por las cabeceras IPv6 (40 bytes) es muy grande y supondría un desperdicio del ancho de banda proporcionado por la capa física del estándar IEEE 802.15.4.

El estándar IETF 6LoWPAN define una capa de adaptación entre IPv6 y la capa MAC del estándar IEEE 802.15.4 que realiza la compresión y la fragmentación de las largas cabeceras de los paquetes IPv6 para que quepan en paquetes IEEE 802.15.4, suprimiendo información redundante que pueda ser deducida de las otras capas de la pila de protocolos.

Los datagramas 6LoWPAN que deben ser transportados sobre la capa MAC del estándar IEEE 802.15.4 incluyen un prefijo formado por conjunto de cabeceras de diversos tipos. En particular, los tipos de cabecera se agrupan en cuatro categorías dependiendo de su función en la capa de adaptación 6LoWPAN. Las

cabeceras pueden estar presentes o no en los paquetes dependiendo de si son necesarias. En la figura 22 se muestran los cuatro tipos de cabecera 6LoWPAN que se resumen a continuación:

- **Cabecera «NO 6LoWPAN»:** se utiliza cuando el paquete no cumple con la especificación 6LoWPAN. Esto permite la coexistencia con nodos que no son 6LoWPAN en la misma red.

- **Cabecera «Dispatch»:** se utiliza para comprimir una cabecera IPv6 o para gestionar una transmisión *multicast/broadcast* en la capa de enlace. Se incluye un número de secuencia para detectar y suprimir paquetes duplicados. La cabecera «Dispatch» incluye diversos tipos (LOWPAN_IPHC, LOWPAN_HC1 y LOWPAN_BC0) que determinan el esquema de codificación usado para comprimir la cabecera IPv6.

- **Cabecera «Mesh Addressing»:** se utiliza para realizar el reenvío de paquetes IEEE 802.15.4 hacia la capa de enlace, convirtiendo una red *single-hop* en *multi-hop*. Es decir, esta cabecera se coloca en los paquetes que deben dar varios saltos hasta llegar al nodo destino. Además de las direcciones de los nodos fuente y destino, se incluye la dirección del siguiente nodo en cada uno de los saltos.

- **Cabecera «Fragmentation»:** se utiliza cuando un datagrama 6LoWPAN no cabe en un paquete IEEE 802.15.4 y debe fragmentarse. Esta cabecera incluye el tamaño del datagrama que debe ser fragmentado, un identificador (*tag*) del datagrama IP original, y el número de fragmento del datagrama.

Figura 22. Tipos de cabeceras en paquetes 6LoWPAN

First 2 bits		Following bit combinations	
NO 6LoWPAN	00	xxxxxx	Any combination
Dispatch	01	000000	Additional Dispatch byte follows
		000001	Uncompressed IPv6 Addresses
		000010	LOWPAN_HC1 compressed IPv6
		010000	LOWPAN_BC0 broadcast
		1xxxxx	LOWPAN_IPHC compressed IPv6
Mesh Addressing	10	xxxxxx	Any combination
Fragmentation	11	000xxx	First Fragmentation Header
		100xxx	Subsequent Fragmentation Header

Fuente: extraído de Palattella y otros (2013)

3.2.5. IETF RPL

En las redes LPWAN, el enrutado de paquetes a través de los nodos de la red es un proceso complejo debido a que los nodos suelen estar alimentados con baterías; puede ser necesario alargar los tiempos en modo *sleep*, y los cambios de topología de la red pueden ser muy frecuentes, especialmente si los nodos se mueven.

El grupo de trabajo IETF «Routing over Low Power and Lossy (ROLL) networks» ha desarrollado el protocolo RPL (Routing Protocol for Low power and Lossy Networks) para redes LPWAN con topología en malla. El protocolo RPL puede soportar una gran variedad de capas de enlace, es capaz de construir tablas de enrutamiento de forma rápida, distribuir el conocimiento de las rutas entre los nodos de la red y adaptarse a los cambios de topología de forma eficiente.

RPL organiza la red mediante un Directed Acyclic Graph (DAG) creado a partir de un dispositivo Gateway que se encarga de conectar la LPWAN a internet. En RPL, los nodos informan periódicamente de su distancia al Gateway, permitiendo a los nodos vecinos obtener su propia distancia.

3.2.6. IETF CoAP

Una red LPWAN que soporte IPv6 permite la conexión de sus nodos a escala mundial a través de internet. El estándar IETF Constrained Application Protocol (CoAP) especifica un protocolo HTTP para nodos que presentan limitaciones de energía y de capacidad de procesamiento. El protocolo CoAP permite convertir cualquier nodo de la red en un servidor web, de forma que un nodo puede publicar las lecturas de sus sensores en un servidor en internet, o un dispositivo móvil puede realizar consultas directas.

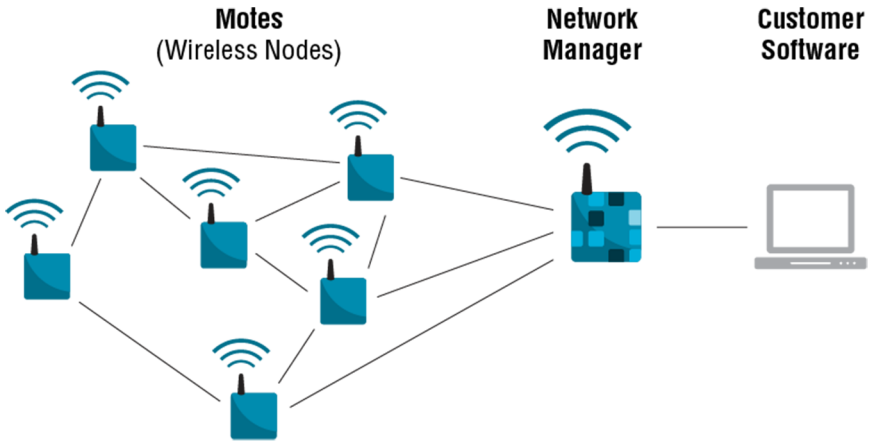
3.3. Implementaciones

Las primeras soluciones de redes inalámbricas deterministas para aplicaciones industriales fueron desarrolladas por la empresa Dust Networks, que en la actualidad está integrada en Linear Technology. El producto de Dust Networks se denomina SmartMesh y está basado en el Time Synchronized Mesh Protocol (TSMP). El protocolo TSMP fue diseñado por Dust para redes con topología en malla e incluye una capa MAC basada en la idea del Time Slotted Channel Hopping (TSCH) sobre la capa física del estándar IEEE 802.15.4 en la banda de 2.4 GHz.

La figura 23 muestra un ejemplo de red SmartMesh de Dust Networks. Una red SmartMesh está formada por un nodo «manager» que monitoriza y controla el rendimiento de la red, y transfiere datos entre los nodos de la red y una aplicación de usuario que corre en un ordenador. Los nodos inalámbricos se

denominan «motas», que están conectadas a los sensores y actuadores del sistema, puede enviar o recibir datos, e intercambian paquetes con otras motas y con el «manager» de la red.

Figura 23. Red SmartMesh de Dust Networks



Fuente: extraído de http://www.linear.com/products/smartmesh_ip

Dust Networks ha desarrollado dos categorías de producto a partir de su tecnología de red SmartMesh:

- SmartMesh WirelessHART: cumple el estándar WirelessHART (IEC 62591) utilizado en aplicaciones industriales.

- SmartMesh IP: combina el protocolo TSMP y una implementación de la capa 6LoWPAN con un consumo de potencia muy reducido.

Una de las principales características de la tecnología SmartMesh de Dust Networks es el bajo consumo energético de las motas, que están basadas en el System-on-Chip Eterna desarrollado por Dust. Las motas pueden alimentarse con baterías o mediante dispositivos de captura de energía (*energy harvesters*). Además, a partir de las tasas de generación de tráfico en cada mota, el «manager» gestiona el proceso de enrutamiento de tráfico para minimizar el consumo energético y extender el tiempo de vida de la red.

En paralelo al desarrollo de productos comerciales y al proceso de estandarización llevado a cabo por el IETF 6TiSCH Working Group, existen diversos grupos de investigación que han realizado implementaciones de redes inalámbricas deterministas basadas en la arquitectura 6TiSCH. Entre ellas, destacan las implementaciones realizadas sobre sistemas operativos *open-source*, como OpenWSN (Watteyne y otros, 2012), Contiki (Dunkels, Gronvall y Voigt, 2004), RIOT (Baccelli, Hahm, Gunes, Wahlisch y Schmidt, 2013) y TinyOS (Levis y otros, 2005), utilizando nodos de red (motas) formados por un microcontrolador, un transceptor de radio-frecuencia y sensores.

Para más información sobre las características (plataformas hardware, protocolos, consumo energético, etc.) de las diferentes implementaciones de protocolos para redes inalámbricas deterministas sobre sistemas operativos *open-source*, se recomienda consultar Watteyne y otros (2016).

Bibliografía

Baccelli, E.; Hahm, O.; Gunes, M.; Wahlisch, M.: Schmidt, T. C. (2013). «RIOT OS: Towards an OS for the internet of things». En: *Actas de la 2013 IEEE Conference on Computer Communications Workshops* (págs. 79-80). Turín, 14-19 de abril.

Bello, L. L. (2014). «Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging». En: *Actas de la 2014 IEEE Emerging Technology and Factory Automation (ETFA)* (págs. 1-8). Barcelona.

Dujovne, D.; Watteyne, T.; Vilajosana, X.; Thubert, P. (2014, diciembre). «6TiSCH: deterministic IP-enabled industrial internet (of things)». *IEEE Communications Magazine* (vol. 52, núm. 12, págs. 36-41).

Dunkels, A.; Gronvall, B.; Voigt, T. (2004). «Contiki—A lightweight and flexible operating system for tiny networked sensors». En: *Actas de la International Conference on Local Computer Networks* (págs. 455-462).

IEEE 802.1AS - Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

IEEE 802.1Qat - Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Amendment: 9: Stream Reservation Protocol (SRP).

IEEE 802.1Qav - IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams.

IEEE 1722 - IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks.

IEEE ASbt - Timing and Synchronization for Time-Sensitive Applications.

IEEE Qca - Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Path Control and Reservation.

IEEE 802.1Qbv - Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Enhancements for Scheduled Traffic.

IEEE 802.1Qbu - Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks - Amendment: Frame Preemption.

IEEE 802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements.

IEEE 802.1CB - Standard for Local and Metropolitan Area Networks-Frame Replication and Elimination for Reliability.

IEEE 802.1BA- IEEE standard for Local and metropolitan area networks-Audio Video Bridging (AVB) Systems.

IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks.

IETF 6TiSCH - IPv6 over the TSCH mode of IEEE 802.15.4e.

Imtiaz, J.; Jasperneite, J.; Han, L. (2009). «A performance study of Ethernet audio video bridging (AVB) for industrial real-time communication». En: *Actas de la 14th Int. Conf. Emerging Technologies and Factory Automation (EFTA)* (págs. 1-8).

Levis, P. y otros (2005). «TinyOS: An operating system for sensor networks». En: *Ambient Intelligence* (págs. 115-148). Berlín: Springer Verlag.

Palattella, M. R. y otros (2013, 3er. trimestre). «Standardized Protocol Stack for the Internet of (Important) Things». *IEEE Communications Surveys & Tutorials* (vol. 15, núm. 3, págs. 1389-1406).

Palattella, M. R. y otros (2016, 15 de enero). «On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks». *IEEE Sensors Journal* (vol. 16, núm. 2, págs. 550-560).

Steiner, W. y otros (2016). «Next generation real-time networks based on IT technologies». En: *Actas de la 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (págs. 1-8). Berlín.

Thomas, L.; Shalu, R.; Daniel, J. J.; Anand, S. V. R.; Hegde, M. (2017). «6TiSCH operation sublayer (6top) implementation on Contiki OS». En: *2017 9th International Conference on Communication Systems and Networks (COMSNETS)* (págs. 423-424). Bangalore.

Watteyne, T.; Tuset-Peiro, P.; Vilajosana, X.; Pollin, S.; Krishnamachari, B. (2017, mayo). «Teaching Communication Technologies and Standards for the Industrial IoT? Use 6TiSCH!». *IEEE Communications Magazine* (vol. 55, núm. 5, págs. 132-137).

Watteyne, T. y otros (2012). «OpenWSN: A standards based low-power wireless development environment». *Trans. Emerging Telecommun. Technol.* (vol. 23, núm. 5, págs. 480-493).

Watteyne, T. y otros (2016, mayo). «Industrial Wireless IP-Based Cyber – Physical Systems». En: *Actas del IEEE* (vol. 104, núm. 5, págs. 1.025-1.038).

Wollschlaeger, M.; Sauter, T.; Jasperneite, J. (2017). «The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0». *IEEE Industrial Electronic Magazine* (vol. 11, núm. 1, págs. 17-27). DOI: 10.1109/MIE.2017.2649104.