

REDES INALÁMBRICAS DE SENSORES

MÁSTER EN INGENIERÍA ELECTRÓNICA, ROBÓTICA Y AUTOMÁTICA

PRÁCTICA 3. REDES EN CONTIKI

Eduardo Hidalgo Fort

ehidalgo@us.es

José María Hinojo Montero

jhinojo@us.es

Dpto. Ingeniería
Electrónica



ÍNDICE

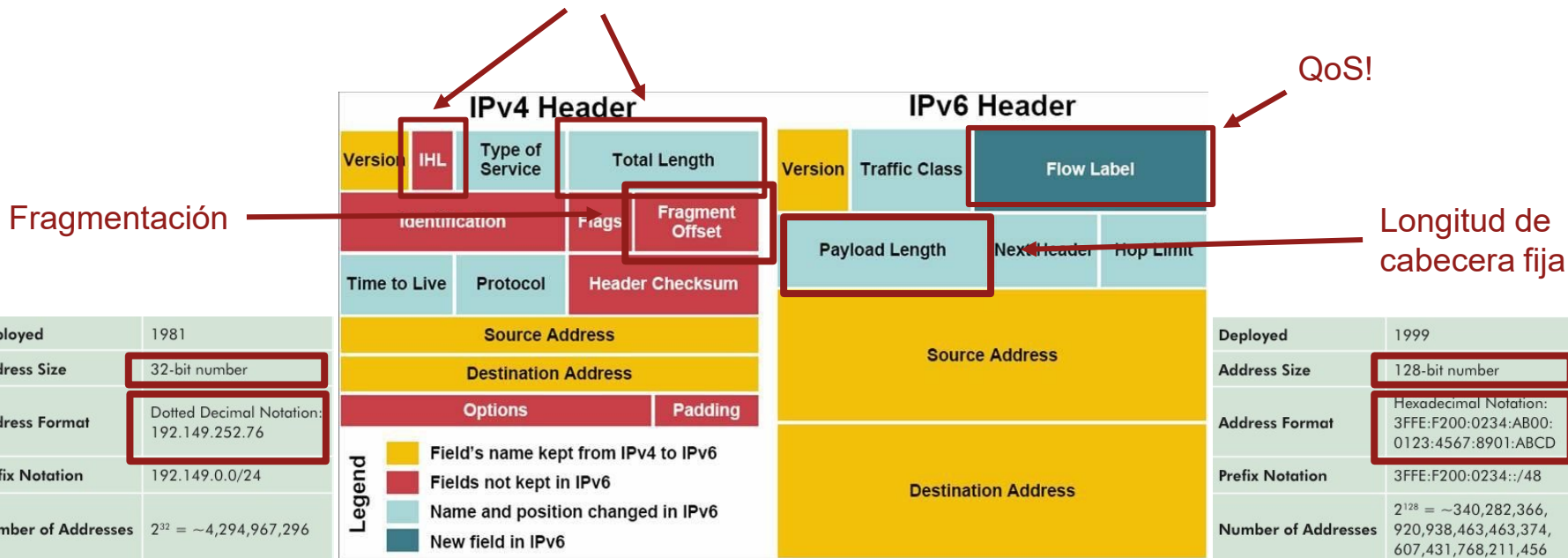
1. Introducción: IP4 vs IPv6
2. Protocolos incluidos
3. Redes en CONTIKI
4. Capas o niveles del modelo OSI
5. Ejercicios

I. INTRODUCCIÓN: IPv4 vs IPv6

INTRODUCCIÓN

- CONTIKI incorpora una pila de comunicaciones basada en IPv6, pero adaptada para redes de bajo consumo y con posibilidad de pérdidas de paquetes.
- Comparativa entre las versiones 4 y 6 de IP (Internet Protocol)

Longitud de cabecera variable

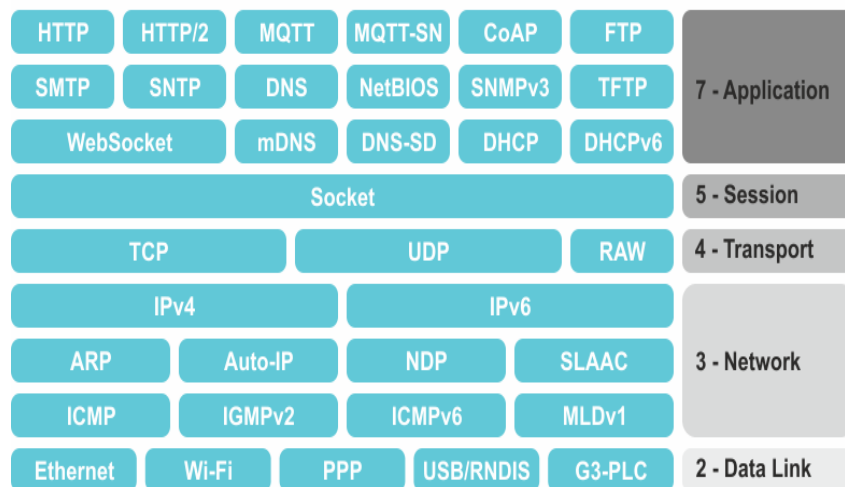


12 campos → 20 Bytes

8 campos → 40 Bytes

II. PROTOCOLOS SOPORTADOS

- La pila de CONTIKI tiene soporte para los protocolos:
 - **TCP:** Transfer Control Protocol
 - **UDP:** User Datagram Protocol
 - **DNS:** Domain Name System
 - **RPL:** Routing Protocol for Low-Power and Lossy Networks
 - **CoAP:** Constrained Application Protocol
 - **LWM2M:** LightWeight Machine-to-Machine
 - **Websockets:** canales de comunicación un protocolo de transporte



III. REDES EN CONTIKI

REDES EN CONTIKI (I)

- Los códigos para la gestión de redes se encuentran en

```
~/contiki-ng/os/net
```

- Los parámetros de configuración se encuentran en el archivo de cabecera *netstack.h*

- Modos de enrutamiento:

```
ROUTING_CONF_RPL_LITE  
ROUTING_CONF_RPL_CLASSIC  
ROUTING_CONF_NULLROUTING
```

- Capa de red:

```
NETSTACK_CONF_WITH_IPV6  
NETSTACK_CONF_WITH_NULLNET
```

III. REDES EN CONTIKI

REDES EN CONTIKI (II)

- Capa de acceso al medio (MAC):

```
MAC_CONF_WITH_NULLMAC  
MAC_CONF_WITH_CSMA  
MAC_CONF_WITH_TSCH  
MAC_CONF_WITH_BLE
```

- Construcción de las tramas (framer_802154):

```
NETSTACK_FRAMER
```

- Transceptor radio (medio físico):

```
NETSTACK_CONF_RADIO
```

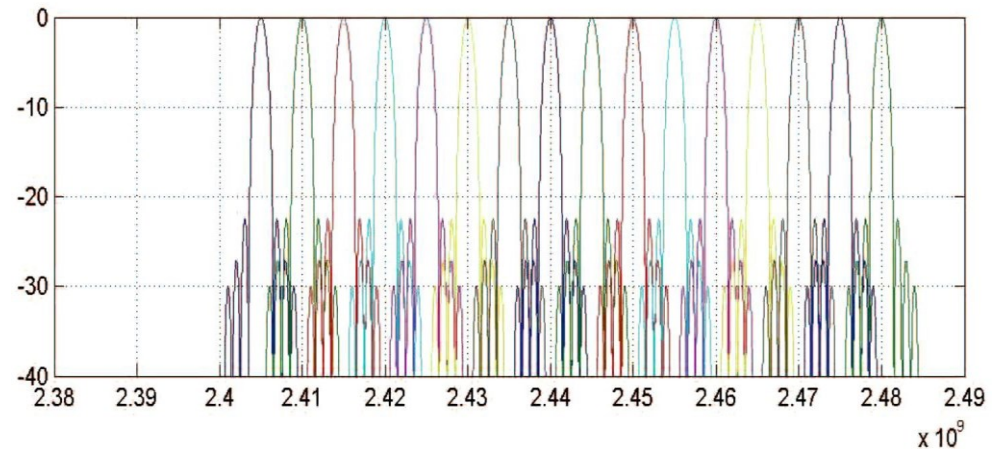
III. REDES EN CONTIKI

RADIO

- El archivo con los parámetros configurables, se encuentra en:
~/contiki-ng/os/contiki-default-conf.h
- Llamadas a librería desde:
~/contiki-ng/os/dev/radio.h

```

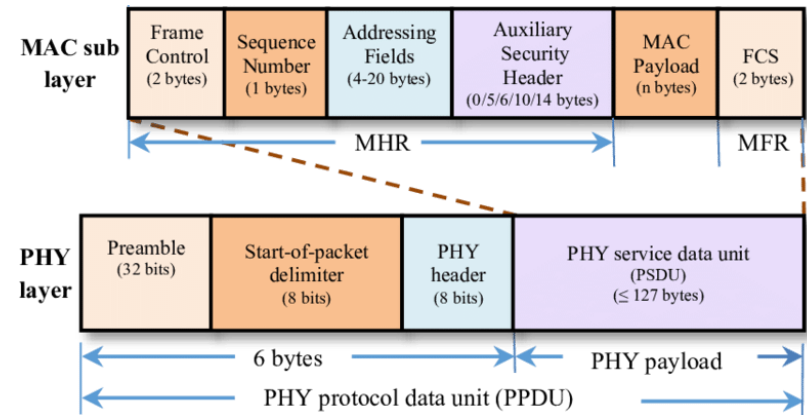
RADIO_PARAM_POWER_MODE
RADIO_PARAM_CHANNEL
RADIO_PARAM_PAN_ID
RADIO_PARAM_16BIT_ADDR
RADIO_PARAM_RX_MODE
RADIO_PARAM_TX_MODE
RADIO_PARAM_TXPOWER
RADIO_PARAM_CCA_THRESHOLD
RADIO_PARAM_RSSI
RADIO_PARAM_LAST_RSSI
RADIO_PARAM_LAST_LINK_QUALITY
RADIO_PARAM_64BIT_ADDR
RADIO_PARAM_LAST_PACKET_TIMESTAMP
RADIO_PARAM_SHR_SEARCH
    
```



III. REDES EN CONTIKI

ACCESO AL MEDIO (MAC)

- La configuración y los códigos fuente de C están en el directorio
`~/contiki-ng/os/net/mac`
 - Por ejemplo, en `mac.h` se definen los posibles estados de una transmisión radio: OK, colisión, no ACK, retrasada, error.
- En cada subdirectorio se encuentra la implementación de otros módulos de esta capa:
 - **BLE:** Bluetooth Low Energy
 - **CSMA:** protocolo de acceso al medio (*Carrier-Sense Multiple Access*)
 - **FRAMER:** composición de las tramas IEEE 802.15.4.
 - **TCSH:** Time Slotted Channel Hopping, incluyendo 6TiSCH "minimal config"



IEEE 802.15.4 Overview Packet Structure

III. REDES EN CONTIKI

ENRUTAMIENTO

- CONTIKI-ng define dos modos de enrutamiento RPL, en el directorio `~/contiki-ng/os/net/routing`
- RPL Clásico
 - La implementación original para Contiki (2009)
 - Se han ido añadido nuevas funcionalidades con el tiempo (múltiples instancias, modos de guardado y no-guardado, multicast...)
 - Esto lleva a una implementación más compleja y más consumo de ROM

<http://www.diva-portal.org/smash/get/diva2:1042739/FULLTEXT01.pdf>

- RPL Lite
 - Se utiliza por defecto en Contiki-ng (2017)
 - Funcionalidad básica y estable (se pierden algunas opciones → menos interoperabilidad)
 - Mejor rendimiento y menor *huella* en la memoria de programa (ROM)

IV. CAPAS O NIVELES DEL MODELO OSI

REDES EN CONTIKI: PROTOCOLO IPV6

- Todos los módulos relacionados con el direccionamiento IPv6 están en `~/contiki-ng/os/net/ipv6`
- Contiki-ng utiliza la implementación uIP (*micro-IP*), que permite utilizar TCP/IP para microcontroladores de 8 y 16 bit.
- Desarrollado inicialmente por *Adam Dunkels*
- La idea es reducir de forma considerable el tamaño necesario para guardar el fichero compilado en los nodos.
- En el archivo *uip.h* se definen varios parámetros de esta pila.

IV. CAPAS O NIVELES DEL MODELO OSI

REDES EN CONTIKI: NIVEL APLICACIÓN

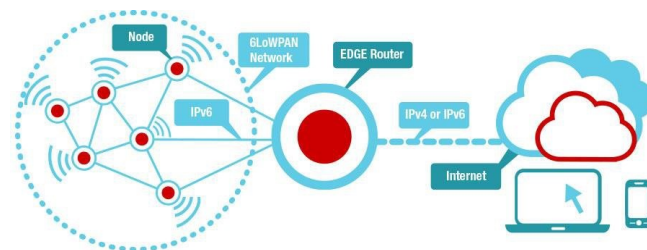
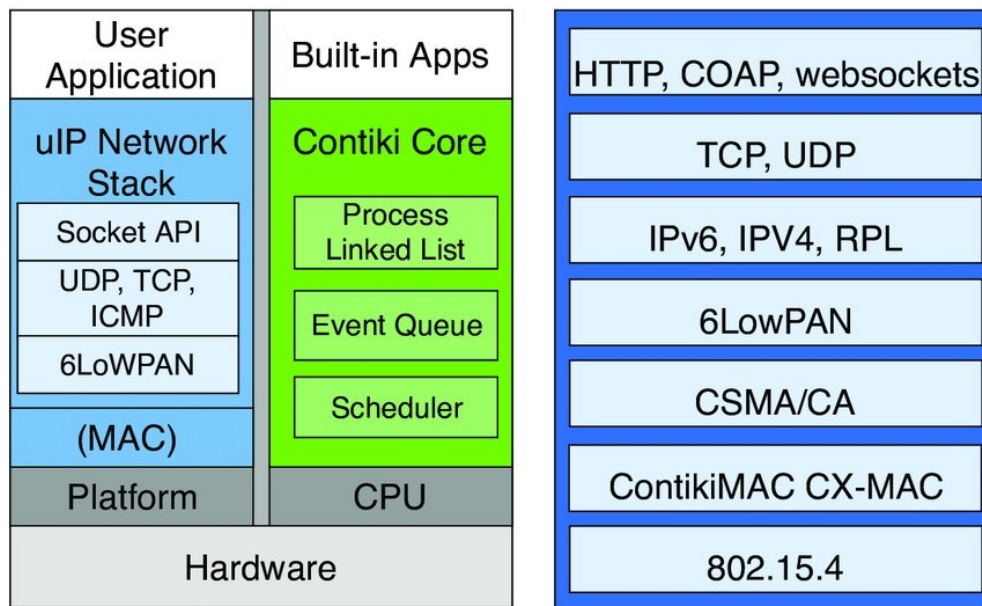
- Para completar la torre de protocolos, queda el nivel de aplicación

~/contiki-ng/os/net/app-layer

- En Contiki-ng tenemos algunos módulos listos para utilizar:
 - **SNMP:** *Simple Network Management Protocol*. Intercambio de información de administración entre dispositivos de una red).
 - **COAP:** *Constrained Application Protocol*. Permite proporcionar servicios de aplicación utilizando muy pocos recursos. Ejemplo: aplicación REST en Contiki de bajo consumo
<http://dunkels.com/adam/kovatsch11low-power.pdf>
 - **HTTDP-WS:** un servidor de páginas web (basado en HTTP) muy simplificado.
 - **HTTP-Socket:** conexión (canal) para transferencia de datos utilizando HTTP sobre TCP.
 - **MQTT:** *MQ Telemetry Transport*. Sistema de suscripción/publicación de mensajes ("data broker"), ideado para redes de bajo ancho de banda, con posibilidad de fallos o para dispositivos con procesador y/o memoria de baja capacidad.

IV. CAPAS O NIVELES DEL MODELO OSI

REDES EN CONTIKI: RESUMEN



V. EJERCICIOS

EJERCICIO 1 (4 PUNTOS)

- Se pide desplegar la red del ejercicio 3 de la práctica 2 en 1 nodo Servidor y 1 nodo Cliente **REALES**.
- Identificar en el código el PANID y el canal utilizado.
- Adjuntar tantas capturas/imágenes como sea necesario para demostrar la implementación y despliegue del ejercicio.

EJERCICIO 2 (2 PUNTOS)

- Se pide modificar el PANID y canal utilizado y desplegar nuevamente la red, en este caso en COOJA, con 3 nodos clientes y 1 servidor. Utilizar PANID distinto al del resto de compañeros.
- Explicar cómo se realiza dicha modificación, indicando los valores utilizados.
- Adjuntar tantas capturas como sea necesario para demostrar la implementación y despliegue del ejercicio.
- Archivos .c y aquellos archivos que hayan sido necesario modificar.

V. EJERCICIOS

EJERCICIO 3 (4 PUNTOS)

- Se pide desplegar en COOJA las redes diseñadas en los ejercicios 1 y 2 de esta práctica. Cada una de las redes contará con 3 nodos clientes y 1 servidor.
 - La red del ejercicio 1 se implementará haciendo uso de nodos Sky (servidor) y Z1 (clientes).
- Adjuntar tantas capturas como sea necesario para demostrar la implementación y despliegue del ejercicio.
- Archivos .c, Makefile (todos, si se utiliza más de uno) y aquellos archivos que hayan sido necesario modificar.
- *Se recomienda realizar la compilación de cada una de las redes desde la ventana de comandos, guardando los archivos .z1 y .sky en carpetas distintas*