



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de un dispositivo seguro y fiable de transmisión
unidireccional para redes de alta confidencialidad*

Grado en Ingeniería Mecánica

ALUMNO: Eloy Rafael Hidalgo Romero

DIRECTOR: Miguel Rodelgo Lacruz

CURSO ACADÉMICO: 2015-2016

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Desarrollo de un dispositivo seguro y fiable de transmisión
unidireccional para redes de alta confidencialidad*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Infantería de Marina

UniversidadeVigo

RESUMEN

En ocasiones la seguridad proporcionada por los dispositivos de protección tradicionales, como los firewall (cortafuegos) no es suficiente para determinadas redes clasificadas y de muy alta confidencialidad, lo que hace necesario aislar estas redes mediante *airgaps*, que desconectan físicamente una red de otra, lo que conlleva numerosos inconvenientes.

Para estos entornos se ha propuesto un nuevo dispositivo unidireccional no reprogramable, denominado diodo de datos, que es capaz de interconectar dos redes y garantizar físicamente que la información solo se transmite desde una de las redes a la otra pero no en sentido contrario, de manera que ningún tipo de información sale, de forma accidental o intencionada, de la red de alta confidencialidad.

En este proyecto se describe la realización de un diodo de datos, utilizando software de código abierto y hardware libre. En ausencia de un canal de retorno, el receptor no puede utilizar protocolos de transmisión fiables como TCP, por lo que el desarrollo incluye un método de corrección de errores, FEC, que garantice la fiabilidad de la transmisión. Respecto a un dispositivo comercial el prototipo tiene un coste muy inferior y la garantía de que funciona según lo especificado y se encuentra exento de puertas traseras.

PALABRAS CLAVE

Ciberseguridad, Red unidireccional, Diodo de datos, Confidencialidad, Fibra Óptica.

CONTENIDO

Contenido	3
Índice de Figuras	5
Índice de Tablas.....	8
1 Introducción.....	9
1.1 Motivación	9
1.2 Objetivos	11
1.3 Recursos utilizados	11
1.3.1 Hardware.....	11
1.3.2 Software	12
1.4 Organización de la memoria	12
2 Estado del arte	13
2.1 La Armada y la Ciberseguridad	13
2.1.1 Niveles de seguridad y tipos	13
2.1.2 Sistema de mando naval	14
2.1.3 Redes OTAN	16
2.2 Sistemas de protección de datos.....	17
2.2.1 Firewall	17
2.2.2 Diodo de datos	18
2.2.2.1 Funcionamiento	19
2.2.2.2 Protocolos y medidas.....	20
2.2.2.3 Productos y aplicaciones	22
3 Desarrollo del TFG.....	31
3.1 Componentes de la arquitectura	31
3.1.1 Proxies	31
3.1.2 Conversores a fibra óptica	33
3.1.3 UDP y UDPcast	33
3.1.4 FEC	34
3.1.5 FTP	34
3.2 Diseño de la solución	34
3.3 Configuración de los conversores y conexión mediante fibra óptica	35
3.4 Configuración de las RaspberryPi.....	38
3.4.1 Instalación Raspbian.....	38

3.4.2 Configuración de los interfaces de red	39
3.4.3 Instalación de programas	40
3.4.3.1 UDPcast.....	40
3.4.3.2 Instalación Iperf.....	41
3.4.3.3 Servidor FTP	41
3.4.3.4 Conexión de las Raspberries entre sí.....	42
3.4.3.5 Tabla de encaminamiento ARP	43
3.5 Transmisión de información de forma unidireccional	44
3.6 Limitación de tasa y FEC.....	45
3.7 Script.....	45
3.7.1 Transmisor	46
3.7.2 Receptor.....	47
4 Pruebas y Validación.....	49
4.1 Establecimiento de la tasa de transmisión.....	49
4.2 Simulación y corrección de errores.....	54
4.2.1 Traffic Control	54
4.2.2 Protección frente a errores, FEC.....	54
5 Conclusiones y líneas futuras	57
5.1 Conclusiones	57
5.1.1 Conclusiones del estudio	57
5.1.2 Conclusiones del desarrollo	59
5.2 Líneas futuras	60
6 Bibliografía.....	61
ANEXO I: PRESUPUESTO.....	65
ANEXO II: SCRIPT TRANSMISOR.....	67
ANEXO III: SCRIPT RECEPTOR.....	69
ANEXO IV: ACRÓNIMOS Y GLOSARIO.....	71

ÍNDICE DE FIGURAS

Figura 2-1. Organigrama de seguridad de la información en la Armada	14
Figura 2-2. Situación Táctica con Diodo de Datos.....	15
Figura 2-3. Solución de conexión en redes de la OTAN.....	16
Figura 2-4. Funcionamiento <i>Firewall</i>	17
Figura 2-5. Funcionamiento <i>Air gap</i>	18
Figura 2-6. Partes High/Low	19
Figura 2-7. Funcionamiento del diodo de datos a la inversa en entornos industriales	20
Figura 2-8- Triple “Hand-Shake” [2]	20
Figura 2-9. Diodo de datos	21
Figura 2-10. Ejemplo uso FEC.....	22
Figura 2-11. Funcionamiento	23
Figura 2-12. Enlace unidireccional con diodo de datos	23
Figura 2-13. Diodo de datos <i>OWL</i>	24
Figura 2-14. <i>Owl Miniaturized Cross Domain Solution</i> (MCDS)	24
Figura 2-15. Situación táctica.....	25
Figura 2-16. Diodo de datos <i>FOX IT</i>	25
Figura 2-17. <i>Sina One Way</i>	26
Figura 2-18. Clasificaciones Sima.....	26
Figura 2-19. HSDD	27
Figura 2-20. Montaje sobre protoboard.....	28
Figura 2-21. Esquema de diodo de datos.....	29
Figura 3-1. RaspberryPi B+.....	32
Figura 3-2. Proxy RaspberryPi	32
Figura 3-3. Convertidor Multimedia Multi-modo de 10/100Mbps.....	33
Figura 3-4. Montaje del diodo de datos.....	35
Figura 3-5. Diodo de datos <i>FOX IT</i> [18]	36
Figura 3-6. Configuración convertidores ópticos.....	36
Figura 3-7. Montaje de los convertidores.....	37
Figura 3-8. Configuración convertidor de fibra	37
Figura 3-9. Escritura de Raspbian en tarjeta SD.	38
Figura 3-10. Actualización de repositorios	38
Figura 3-11. Actualizar versión.....	38

Figura 3-12. Versión Raspbian Jessie	38
Figura 3-13. Interfaz de configuración raspi-config.....	39
Figura 3-14. Configuración de interfaces	40
Figura 3-15. Configuración IP del dispositivo	40
Figura 3-16. Instalación UDPcast	41
Figura 3-17. Instalación Iperf	41
Figura 3-18. Instalación vsftpd.....	41
Figura 3-19. Acceso a configuración vsftpd	41
Figura 3-20. Autorización de escritura en servidor FTP	42
Figura 3-21. Reinicio vsftpd.....	42
Figura 3-22. Unión de las RPi con cable cruzado	43
Figura 3-23. Ping.....	43
Figura 3-24. Envío de información por UDPcast.....	43
Figura 3-25. Recepción de información por UDPcast.....	43
Figura 3-26. Añadir MAC/IP a tabla ARP	44
Figura 3-27. Netcat en Rx	44
Figura 3-28. Netcat en Tx.....	44
Figura 3-29. Enlace correcto Tx-Rx.....	44
Figura 3-30. Transmisión unidireccional receptor	45
Figura 3-31. Transmisión unidireccional transmisor.....	45
Figura 3-32. Udp-sender limitación de la tasa	45
Figura 3-33. Udp-sender protección de errores.....	45
Figura 3-34. Creación script transmisor	46
Figura 3-35. Script init.d	46
Figura 3-36. Permisos de ejecución al script.....	47
Figura 3-37. Script con permisos	47
Figura 3-38 Convertir el script en Daemon.....	47
Figura 3-39. Creación script receptor	47
Figura 3-40. Script init.d	48
Figura 3-41. Permisos de ejecución al script.....	48
Figura 3-42. Script con permisos	48
Figura 3-43. Convertir el script en Daemon.....	48
Figura 4-1. Montaje del diodo	49
Figura 4-2. UDPcast sin límite de velocidad.....	50
Figura 4-3. Conexión bidireccional.....	50

Figura 4-4. Iperf transmisor.....	51
Figura 4-5. Iperf receptor	51
Figura 4-6. Resultado Iperf	51
Figura 4-7. Iperf 70 Mbits/segundo.....	51
Figura 4-8. Iperf 60 Mbits/segundo.....	52
Figura 4-9. Porcentaje bajo de pérdidas con 53 Mbits	52
Figura 4-10. 50 Mbits/s	53
Figura 4-11. UDPcast a 50 Mbits/s	53
Figura 4-12. UDPcast a 30 Mbits/segundo	53
Figura 4-13. Introducción pérdidas aleatorias	54
Figura 4-14. Modificación.....	54
Figura 4-15. Consulta	54
Figura 4-16. Introducción FEC.....	55
Figura AII-1. Script Transmisor	67
Figura AIII-1. Script Receptor	69

ÍNDICE DE TABLAS

Tabla 2-1. Servicios Armada.....	15
Tabla 4-1. Relación Pérdidas-Fec-Redundancia	55
Tabla 5-1. Fortalezas y debilidades de los sistemas <i>Firewall</i>	58
Tabla 5-2. Fortalezas y debilidades de los diodos de datos.....	58
Tabla AI-1. Presupuesto	65

1 INTRODUCCIÓN

1.1 Motivación

La seguridad a la hora de transmitir la información siempre ha sido una prioridad fundamental y estratégica en entornos militares. Ya desde la antigüedad, en diferentes batallas y guerras, la utilización de mensajeros ágiles y veloces capaces de entregar el mensaje a tiempo y sin ser interceptados (Maratón) o una codificación para proteger los mensajes (Julio César) para que la información fuera útil, reciente y fiable podía decantar el curso de una contienda.

En la denominada era de la comunicación, el vertiginoso desarrollo de los sistemas de comunicaciones permite la transmisión de grandes cantidades de información en poco tiempo. Con el incremento exponencial del volumen de datos a compartir (imágenes y videos de alta resolución, telemetría, posicionamiento, etc.) también han aumentado las amenazas a la seguridad de la información. Todo ello hace necesario el uso de diferentes niveles y políticas de seguridad para la difusión de la información de manera segura y confidencial. Esta seguridad no solo es clave en entornos o redes de defensa y agencias de inteligencia, que son los que manejan mayor cantidad de información sensible, sino también para gobiernos, grandes empresas, corporaciones e industrias.

A la seguridad de la información en el área informática se le denomina **ciberseguridad**[1], definida según la UIT, *Unión Internacional de Comunicaciones*, como el conjunto de herramientas, políticas, conceptos de seguridad, acciones y tecnologías que pueden utilizarse para proteger los activos de una organización y sus usuarios. La ciberseguridad garantiza que se alcancen y mantengan las propiedades de seguridad de los activos de la organización y de los usuarios contra los riesgos del entorno cibernético. Las propiedades de seguridad incluyen una o más de las siguientes:

- Disponibilidad: poder disponer en cualquier momento y lugar de aquellos recursos necesarios para las comunicaciones.
- Integridad: mantener los datos libres de modificaciones no autorizadas incluyendo la autenticidad, ser capaz de verificar la identidad de los comunicantes, y el no repudio, que impide negar que se ha mandado una información.
- **Confidencialidad**: la característica principal que vamos a tener en cuenta en este trabajo, es la que asegura la protección frente a la divulgación no deseada de información.

La confidencialidad en materia de seguridad de la información es uno de los requerimientos más importantes a la hora de manejar información clasificada o sensible, sobre todo al trabajar en un entorno militar donde mucha de esta información tiene carácter especialmente relevante para la seguridad nacional. A la hora de conectar dos redes de distinto nivel o clasificación de seguridad, como redes militares o de inteligencia, hay que establecer ciertos mecanismos para evitar el filtrado accidental o malintencionado de información almacenada en una red considerada confidencial a otra red menos segura donde las medidas de protección no son las adecuadas o directamente a Internet donde impera la filosofía de libre distribución. Considerando que actualmente esta defensa de la información es crucial en un entorno tan invadido por las nuevas tecnologías y la conexión a Internet, las Fuerzas Armadas deben contar con unos mecanismos de protección que eviten cualquier tipo de amenaza o ataque que pueda comprometer la seguridad de la información que se maneja.

Para conseguir esta confidencialidad y otras necesidades de seguridad se pueden implementar diferentes técnicas y procedimientos a la hora de tratar con esta información, como la codificación de archivos, la creación de redes privadas virtuales, sistemas *software*, *hardware* de protección, etc. A nivel *software* existen los denominados sistemas de prevención de filtrado de información, *Data Loss Prevention*, (DLP) que incluyen medidas como *firewalls*, sistemas de detección de intrusos, antivirus y sistemas de monitorización. Estos sistemas, imprescindibles para la protección ante ataques, están basados en *software* por lo que pueden ser sensibles a errores de configuración, ataques que modifiquen el comportamiento de los sistemas o puertas traseras. Por este motivo no proporcionan suficientes garantías para redes con muy altos requisitos de seguridad, como redes militares, gubernamentales o de defensa. Como resultado, para conseguir la deseada seguridad y confidencialidad, muchas de estas redes no pueden conectarse físicamente a la red y utilizan los denominados *air gaps*, que consisten en escribir los datos a transmitir en un medio físico, en el lado inseguro de la red, y posteriormente conectar el dispositivo físico al lado seguro para copiar estos datos. Este sistema, aunque eficaz desde el punto de vista de la seguridad, resulta ineficiente y engorroso.

Una solución más sofisticada se basa en el desarrollo de sistemas físicos o *hardware*, denominados **diodos de datos**, que permiten la transmisión de datos en una única dirección. Estos dispositivos utilizan elementos de transmisión y recepción no programables, como un diodo transmisor de luz y un diodo receptor, y garantizan la seguridad en el intercambio de información de una red insegura hacia una segura.

La información en un diodo de datos solo se puede transmitir en una única dirección lo que imposibilita que ningún tipo de información, ni siquiera mensajes de actualización o de asentimiento, *acknowledgement*, (ACK) salga de la red de alta seguridad. Este marcado carácter unidireccional impide el uso de protocolos de red “clásicos”, *Transmission Control Protocol* (TCP), que utilizan señalización en ambos sentidos aún para transmitir en un único sentido. Para ello se utilizan protocolos no orientados a la conexión, *User Datagram Protocol* (UDP). También es necesario el uso de ciertos mecanismos de protección para asegurar que los datos llegan sin errores al otro extremo, por ejemplo, los métodos de corrección de errores, *forward error correction*, (FEC)[2], que realizan un reenvío de la información, añadiendo redundancia y garantizando que la información llega íntegra al receptor.

Los diodos de datos, debido a su carácter físico son irreprogramables. Son prácticamente invulnerables, con la excepción de que se llegue de forma física a donde se encuentra instalado y se transmite la información entre las dos redes.

En la actualidad, existen varias empresas especializadas que ofrecen este tipo de dispositivos. No obstante, debido a su relativa reciente invención y la especificidad del mercado tienen un precio muy elevado por producto. Además, su utilización no está exenta de los riesgos derivados que supone adquirir estos productos a empresas externas al Ministerio de Defensa, como pueden ser las puertas traseras o la necesidad de confiar en que el producto opera como realmente asegura el fabricante.

Por ello, en este proyecto se **propone el desarrollo de un dispositivo de transmisión unidireccional** y seguro que permita conectar y realizar intercambio de información desde una red vulnerable conectada a Internet a otra red segura que carece de conexión, evitando al mismo tiempo cualquier tipo de amenaza sobre la red clasificada o segura. El dispositivo será de bajo coste, estará basado en *hardware* libre disponible comercialmente e incluirá los elementos *software* necesarios para transmitir la información de manera fiable a través de los elementos *hardware* unidireccionales.

1.2 Objetivos

El objetivo principal de este trabajo consiste en desarrollar un dispositivo de transmisión unidireccional basado en dispositivos disponibles en el mercado para la transmisión de información entre dos redes de diferente clasificación de seguridad, de manera que el enlace sea seguro y fiable. La transmisión de información se produce a través de un diodo de datos unidireccional *hardware* no reprogramable, de manera que la unidireccionalidad está garantizada a nivel físico y es invulnerable a errores de configuración, filtrados de información, deseados o involuntarios, ataques informáticos o puertas traseras. Las garantías proporcionadas por el dispositivo permiten su utilización en redes de alta confidencialidad que habitualmente se encuentran desconectadas físicamente mediante *air gaps*. El desarrollo incluye también la exposición de los medios *software* necesarios para la transmisión fiable de datos a través del diodo y el análisis de las diferentes opciones *hardware* y *software* disponibles para la implementación.

Para el desarrollo del dispositivo se ha utilizado *hardware* libre y de bajo coste: dos *RaspberryPi* (RPI) con adaptadores Ethernet que actúan como proxies a ambos extremos del diodo, conversores Ethernet-fibra óptica para implementar el diodo de datos propiamente dicho y diverso cableado Ethernet y de fibra óptica para la interconexión de los elementos.

Para el desarrollo *software*, siguiendo la política general de utilización medios libres y abiertos, se ha utilizado el sistema operativo de RPI: Raspbian, y diferentes programas y aplicaciones de Linux algunas de ellas incluidas en Raspbian: la pila de protocolos y los elementos de red, el servidor FTP *vsftpd*, una herramienta de medida de la velocidad del enlace, *Iperf*, y la aplicación *UDPCast* como sistema de transmisión unidireccional.

Para la consecución de estos objetivos generales se establecen los siguientes objetivos específicos:

1. Montaje del dispositivo *hardware* con conversores de fibra.
2. Configuración de los dos proxies con RPI y su sistema operativo, Raspbian.
3. Instalación de programas de Linux, *UDPCast*, *Iperf*, etc.
4. Configuración del enlace mediante datagramas (UDP)
5. Creación de scripts para la automatización de la transmisión.
6. Enlace de cada uno de los proxies mediante servidor FTP.
7. Funcionamiento completo del diodo de datos.
8. Pruebas y validación.

1.3 Recursos utilizados

En este apartado se enumeran los diferentes componentes *hardware* y *software* empleados.

1.3.1 Hardware

1. 2xRaspberry Pi

2. 3xConvertidores Ethernet-fibra óptica
3. 1xCable HDMI
4. 2xCargadores Mini-USB
5. 2x Cable Ethernet
6. 2xCable Fibra Óptica
7. 1xCable Ethernet cruzado

1.3.2 Software

1. Sistema operativo Linux
2. Sistema operativo Raspbian Jessie para RaspberryPi
3. *Software* libre UDPcast
4. *Software* libre vsftpd (FTP service)
5. *Software* libre Iperf
6. *Software* libre Traffic Control

1.4 Organización de la memoria

La memoria está estructurada en cuatro capítulos acompañados de cuatro anexos, según se indica a continuación:

En el presente Capítulo se realiza una introducción al proyecto, se expone la motivación, objetivos y estructura del mismo, así como una enumeración y descripción de los distintos medios *hardware* y *software* utilizados para el desarrollo.

En el Capítulo 2 se realiza una revisión del estado del arte dividida en las tres partes:

- En primer lugar, se expone la clasificación de seguridad en las Fuerzas Armadas, así como los requerimientos que se exigen para alcanzar los diferentes niveles de seguridad.
- En segundo lugar, se describen los sistemas tradicionales de protección de la información y el diodo de datos como solución de protección física de una red.
- En tercer lugar, se presentan varios desarrollos experimentales y de investigación, así como productos actuales de empresas que se dedican a la venta de este tipo de dispositivos.

En el Capítulo 3 se realiza todo el desarrollo llevado a cabo para la construcción del mecanismo de transmisión unidireccional y la configuración de los sistemas *hardware* y *software*.

En el Capítulo 4 se realizan y exponen las diferentes pruebas y validaciones, cuya finalidad es el establecimiento de la tasa de transmisión adecuada y protección contra errores.

En el Capítulo 5, se recogen las conclusiones del estudio llevado a cabo y de la realización del presente proyecto y, a continuación, se indican las posibles líneas de investigación futuras y mejoras aplicables.

Por último, se adjuntan cuatro anexos, el primero con el presupuesto total al que asciende el proyecto realizado, los dos siguientes con los scripts para configurar la automatización del intercambio de información del diodo de datos y en el último se exponen los acrónimos, con su traducción en los casos que corresponde.

2 ESTADO DEL ARTE

En este capítulo de la memoria se describen, en un primer lugar, las diferentes clasificaciones de seguridad de la información en las Fuerzas Armadas, así como las medidas llevadas a cabo en la Armada y especificaciones de la OTAN en materia de Ciberdefensa. En segundo lugar, las tecnologías existentes en la actualidad para realizar el intercambio unidireccional seguro de datos y, por último, algunos de los productos que ofrecen las principales empresas del sector y otros proyectos de investigación en la misma línea.

2.1 La Armada y la Ciberseguridad

La dependencia del ciberespacio dentro del entorno de las Fuerzas Armadas es cada vez mayor, hasta el punto que ya es considerado como el quinto dominio, junto con la tierra, la mar, el aire y el espacio. Es en este dominio donde se realizan los intercambios de información en cualquier formato (texto, imagen, vídeo, correo, mensaje, voz, etc.), que habilitan la capacidad de mando y control e información. En el futuro próximo es previsible un gran crecimiento de dispositivos móviles conectados en red con capacidad de proporcionar información en tiempo real.

2.1.1 Niveles de seguridad y tipos

En ambientes de alta seguridad como lo es el Ministerio de Defensa y las Fuerzas Armadas se establecen diferentes niveles de seguridad para cada tipo de información [3]:

- Secreto
- Reservado
- Confidencial
- Difusión limitada
- Sin clasificar

Las Fuerzas Armadas no son ajenas a establecer las medidas de protección que deben regir las comunicaciones internas para cada nivel de seguridad y, de esta manera, la Armada define las características de las Comunicaciones Navales, entre las que destacan como fundamentales la confianza y la seguridad:

- **Confianza:** Define la necesidad de que la información transmitida sea recibida, lo que se logra mediante la utilización de sistemas de transmisión y recepción adecuados o la utilización de equipos y sistemas de alta fiabilidad entre otros.
- **Seguridad:** Impide que las comunicaciones propias sirvan de fuente de información útil a personas no autorizadas o que un enemigo introduzca confusión en las mismas. Esta característica se enmarca dentro del amplio concepto de INFOSEC (Seguridad de la Información), cuyos principios se recogen en la publicación “Plan General de Seguridad en la Armada” [4]. Esta seguridad a su vez puede dividirse en física, criptográfica y de transmisión:
 - **Seguridad física:** Es el resultado de las medidas físicas necesarias para salvaguardar las instalaciones, equipos, material o documentos clasificados del posible uso que de ellos pudieran hacer personas no autorizadas. En esta ramificación de la definición de seguridad es en la que está centrado este proyecto, en garantizar la integridad y el mantenimiento de la confidencialidad de la información sensible mediante un dispositivo implementado a nivel físico, pero que asegure la seguridad en las capas superiores de la red.
 - **Seguridad criptográfica:** Es el resultado de la adopción de sistemas y procedimientos criptográficos que impidan la obtención de información útil al enemigo, por medio del descifrado de comunicaciones cifradas.
 - **Seguridad de la Transmisión:** Es el resultado de la adopción de medidas que permitan efectuar transmisiones libres del riesgo de poder ser interceptadas, analizadas o confundidas por el enemigo, presunto o real.

2.1.2 Sistema de mando naval

El sistema de mando naval, SMN [5] también denominado sistema de mando y control, es una de las redes clasificadas utilizadas por la Armada para llevar a cabo la transmisión y recepción de órdenes. Este sistema integra diferentes servicios como mensajería y acceso a la red interna de la Armada (Intranet), para tratar información clasificada de carácter operativo.

La seguridad de la información, **Figura 2-1**, requiere personal con alta especialización y adecuada formación en la normativa del Ministerio de Defensa y la Ciberdefensa para controlar las herramientas de seguridad.



Figura 2-1. Organigrama de seguridad de la información en la Armada

Los servicios que ofrece el SMN [6] para mantener las aplicaciones citadas son los expuestos en la **Tabla 2-1**.

Servicios Comunes	Servicios Específicos
Control de dominio Antivirus de red Navegación web Servicio de ficheros Impresión en red Ofimática Correo electrónico	Planeamiento Situación marítima Inteligencia Planeamiento actividades Mensajería instantánea

Tabla 2-1. Servicios Armada

El centro de operaciones de seguridad de la información del Ministerio de Defensa [4] tiene la misión de gestionar las actividades de operación de carácter proactivo, reactivo y de detección, relacionadas con la seguridad de la información y Ciberdefensa en los sistemas de información y telecomunicaciones corporativos de los tres ejércitos.

Una vez analizadas las necesidades y características de la seguridad en la Armada, y sus diferentes servicios de comunicaciones, se concluye que el desarrollo de un sistema de transmisión unidireccional, como el diodo de datos, ofrece grandes posibilidades sobre todo en seguridad física y de la transmisión. De manera que podría ser una forma de apoyar al Sistema de Mando Naval, por ejemplo, en la Jefatura CIS, donde las redes que manejan información sensible quedarían blindadas ante posibles filtraciones de cualquier tipo de datos hacia el exterior.

En un ambiente táctico de una fuerza establecida en tierra, **Figura 2-2**, se propone una situación donde el diodo de datos podría aislar una red de mando con información sensible y a la vez poder recibir información de las unidades subordinadas.

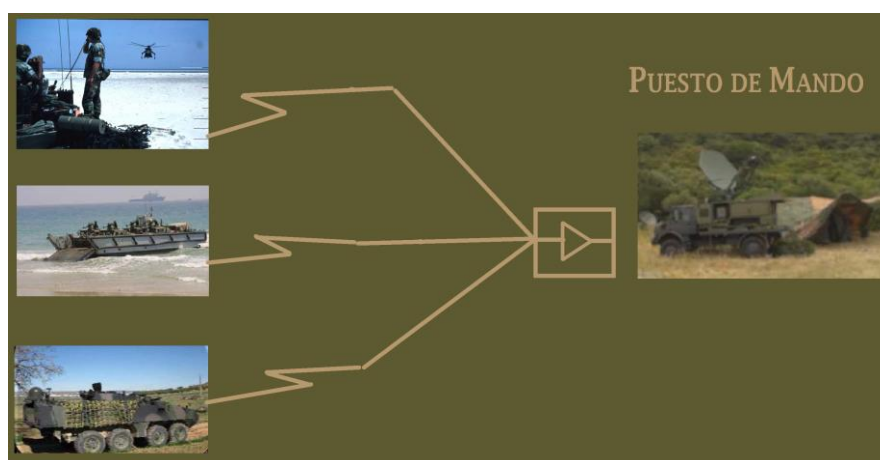


Figura 2-2. Situación Táctica con Diodo de Datos

2.1.3 Redes OTAN

La OTAN establece un Plan de Acción de Ciberdefensa para los países aliados con la finalidad de ayudar a conseguir las capacidades definidas en el concepto *Federated Mission Networking* (FMN) un componente clave de la Iniciativa de Fuerzas Conectadas de la OTAN[7] (*Connected Forces Initiative*, CFI) cuyo principal objetivo es mantener y mejorar las cotas de interoperabilidad, disponibilidad y eficacia alcanzadas por las fuerzas aliadas.

La iniciativa [8] consiste en realizar un gran esfuerzo en educación, entrenamiento y adiestramiento en Ciberdefensa, concentrando la seguridad en los sistemas de comunicación e información, *Communications and Information Systems* (CIS), de manera que abandonando la tradicional idea de protección de los dominios de seguridad, se propone un nuevo enfoque de autoprotección de los sistemas y redes de alta seguridad, que se traduce en una mayor conectividad.

La solución a estas necesidades de seguridad,[9] se establece para algunas configuraciones, como las denominadas MTWAN, conjunto de subredes, conectadas entre ellas a través de enrutadores, conectados a su vez a vías de comunicaciones disponibles a nivel táctico. En el contexto de redes IP, un cable de fibra óptica, junto con una interfaz asociada y el cifrado correspondiente forma una subred que conecta a dos o más rúters. Para interconectar estas MTWAN, **Figura 2-3**, se pueden utilizar los denominados *cross-domain solution* (CDS) o diodo de datos, en los que los datos se transfieren desde una red de baja seguridad a una de alta (*low to high*), desde una red sin clasificar/restringida a una secreta de la coalición o desde una red secreta de la coalición a una red secreta nacional.

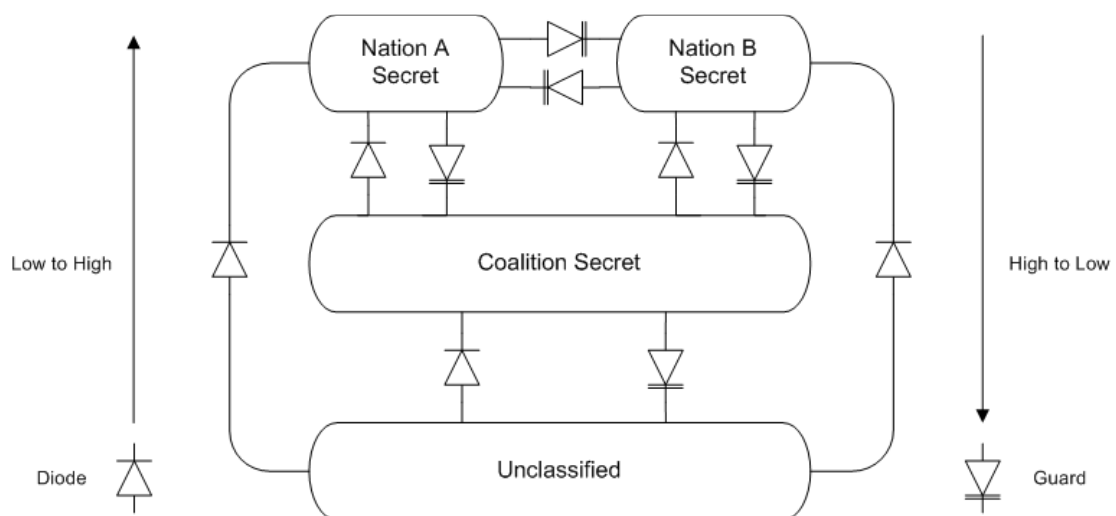


Figura 2-3. Solución de conexión en redes de la OTAN

Para facilitar el uso de este tipo de configuración se utilizan los diodos de datos, que mueven los datos de forma unidireccional, nunca en el sentido *high to low*, es decir, no hay flujo de información de las redes de mayor protección a las de menor.

2.2 Sistemas de protección de datos

Desde la aparición de lo que se considera el Internet moderno, la seguridad de los datos manejados por la red ha sido una de las principales necesidades, lo que ha provocado un permanente desarrollo de medidas de protección cuya función es la de proporcionar las competencias de seguridad descritas anteriormente. Se pueden establecer dos modos de proporcionar dicha seguridad, uno a nivel *software*, sobre las capas de enlace, transporte, red, etc. como son los *firewall* o los antivirus, y el otro modo es establecerlo sobre la base, el nivel físico, obteniendo un blindaje ante cualquier amenaza de *software* malicioso e incluso sobre acciones humanas.

2.2.1 Firewall

Tradicionalmente a nivel aplicación se han utilizado cortafuegos o *firewalls* (de red o de aplicación) que realizan el control de contenidos, *Data Leakage/ Loss Prevention* (DLP), [2] de manera que todos los datos que entran a una computadora o dispositivo lo hacen por una única vía y son analizados por el *firewall*, que actúa como un filtrador de paquetes, inspeccionando cada paquete entrante y saliente, verificando que cumplen con los criterios de seguridad y descartando aquellos que no los cumplen o resultan sospechosos (**Figura 2-4.**)

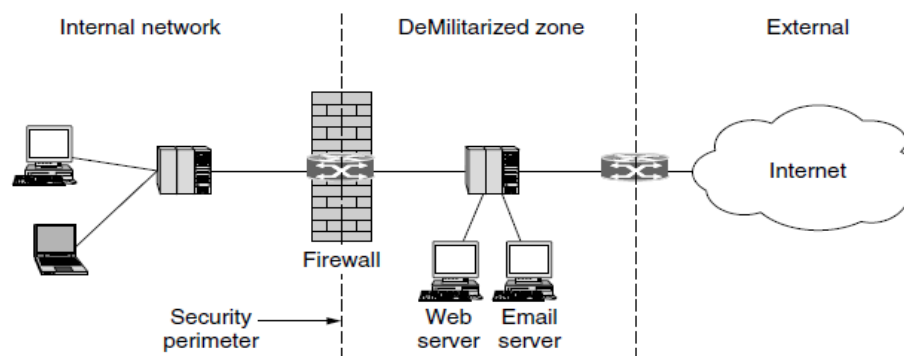


Figura 2-4. Funcionamiento Firewall

Los criterios de filtrado vienen dados por listas donde vienen especificadas las fuentes de destino que son aceptadas y aquellas que están bloqueadas.

El *firewall*, trabaja en las capas de red, transporte y aplicación, y puede evitar que un ataque de denegación de servicio afecte a sistemas que forman parte de una red de mayor nivel de seguridad. También puede evitar que desde una red de menor protección se puedan descargar o modificar datos confidenciales de la red segura. Teóricamente podrían programarse las reglas del *firewall* para evitar cualquier tipo de filtrado de información al exterior. No obstante, esta solución presenta algunas limitaciones. Al tratarse de elementos *software* reprogramables, los *firewalls* son susceptibles de tener errores de configuración, puertas traseras o vulnerabilidades de seguridad explotables por un atacante externo que dejan a las redes expuestas frente a amenazas potenciales. Además, aunque el *firewall* funcione correctamente y solo permita transferir datos desde la red insegura a la segura, los protocolos de comunicación convencionales como TCP, necesitan transmitir asentimientos de la transmisión o ACKs en sentido contrario para confirmar la recepción de los paquetes y adaptar la tasa de transmisión al ancho de banda disponible. Un atacante externo utilizando un troyano o un enemigo infiltrado en la red de alta seguridad podría utilizar estos asentimientos como un canal encubierto (por ejemplo, retrasando o adelantando los ACKs convenientemente) para transmitir información al exterior de forma indetectable para el *firewall*. Estas debilidades de los *firewalls* hacen inviable su uso en redes

con altos requisitos de confidencialidad, como algunas redes militares y, como consecuencia, estas redes permanecen físicamente desconectadas de las redes inseguras, como Internet, por medio de *air gaps* [10], **Figura 2-5**.

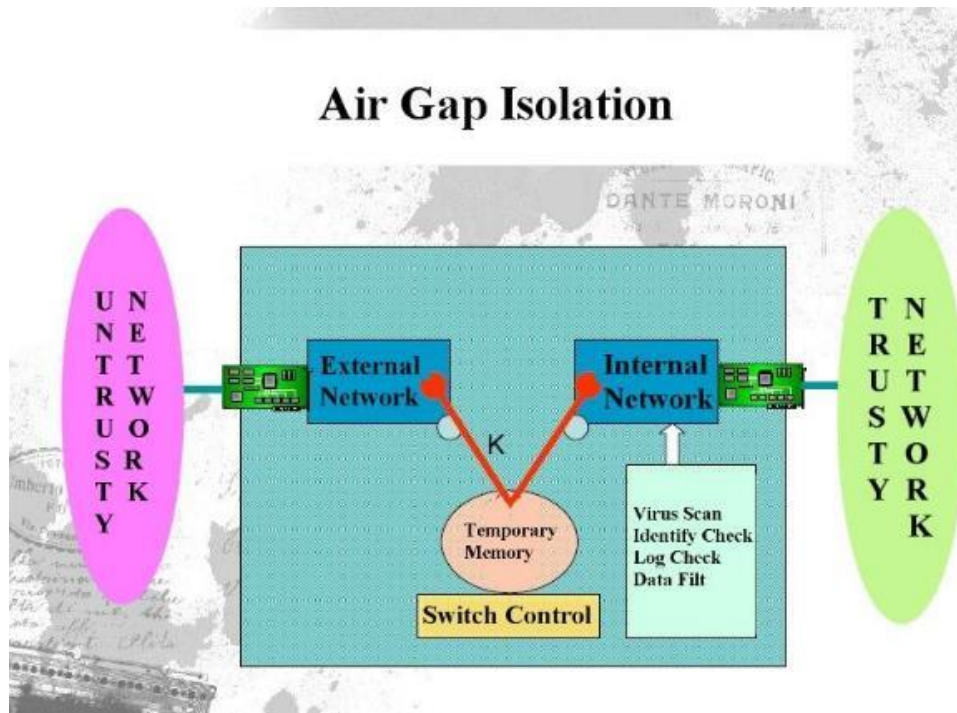


Figura 2-5. Funcionamiento Air gap

Las limitaciones de los *air gaps*, a parte del tiempo y recursos empleados en copiar la información a una memoria intermedia, suponen la exposición a medidas TEMPEST, debido a las fugas y emisiones electromagnéticas, y la vulnerabilidad frente a *software* malicioso o malware.

Para evitar las limitaciones e inconvenientes de los *air gaps*, mientras que se asegura a nivel físico que en ningún caso salga información de la red securizada hacia el exterior, se ha propuesto el empleo de **redes unidireccionales** basadas en el diodo de datos, que se describen en la siguiente sección.

2.2.2 Diodo de datos

Las redes unidireccionales, también denominadas puertas de enlace seguras unidireccionales o diodos de datos surgieron en las décadas de los 80 y los 90 en entornos de defensa y banca con la función de proteger redes y sistemas que tuvieran que manejar información confidencial. El diodo se utiliza para proteger la información asegurando su confidencialidad.

El diodo de datos aparece como una nueva barrera de defensa inmune a errores de configuración, puertas traseras y vulnerabilidades debido a su naturaleza. Consta un dispositivo *hardware* no reprogramable que combinado con algunos componentes *software*, permite realizar transferencias de datos rápidas y fiables, que cumplen estrictamente con una única regla: los datos se transfieren en una única dirección entre las redes, en nuestro caso, de la red insegura a la segura, y nunca a la inversa.

2.2.2.1 Funcionamiento

Las redes unidireccionales identifican las dos partes de la red, emisor y receptor, como una parte segura (*high side*), que es totalmente confidencial y privada, y una insegura (*low side*), **Figura 2-6**, que es la que se encuentra conectada a la red global y por tanto expuesta a sus amenazas.

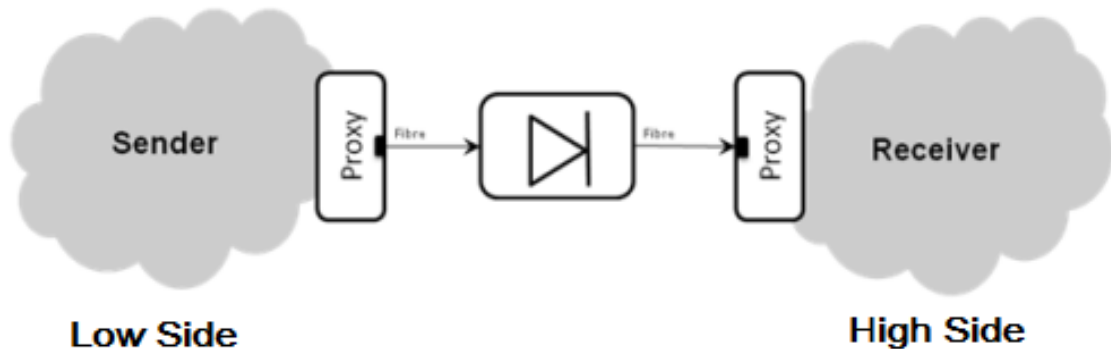


Figura 2-6. Partes High/Low

La unidireccionalidad de la transmisión se asegura típicamente mediante un fotodiodo y un fotoreceptor, como los que se emplean en las redes de fibra óptica. En concreto, se utiliza un diodo emisor de luz, *light emitting diode* (LED), en el emisor para convertir los pulsos eléctricos en pulsos de luz. Estos pulsos de luz viajan a través de un cable de fibra óptica hasta el receptor, que cuenta con un fotoreceptor, que trabaja a la inversa, convirtiendo nuevamente los pulsos de luz en pulsos eléctricos.

La finalidad es aislar las dos partes de la red: en la parte segura, que es la que no tiene salida a Internet, se almacena la información sensible, manteniendo invulnerable la confidencialidad de los datos frente a cualquier tipo de ataque externo. A diferencia de un firewall que trabaja a nivel software, aquí la seguridad se establece a nivel físico, de manera que no se puede atacar, no tiene bugs y, además, una persona física no puede filtrar ni transmitir información desde dentro de forma intencionada o accidental.

De igual manera, el diodo de datos puede funcionar a la inversa, permitir únicamente datos salientes, práctica que se viene usando actualmente, en sistemas de control industrial, plantas de energía nuclear o eléctrica, con la finalidad de proteger entornos e infraestructuras críticas. En este caso el diodo de datos se instala en sentido contrario al mostrado en la **Figura 2-6** con el objetivo de exportar la información generada en el entorno industrial, por maquinaria o autómatas, sin que se introduzcan elementos ajenos al sistema que pueden alterar el funcionamiento de estas máquinas. Caso especialmente sensible son las centrales nucleares, es conocido el caso del gusano Stuxnet [11], que en 2010 infectó las centrifugadoras de uranio enriquecido en la central nuclear de Natanz (Irán). Este software malicioso, de procedencia incierta, se insertó, al parecer, mediante una memoria USB, en los controladores de las centrifugadoras y haciéndose pasar por lícito e indetectable alteró la velocidad de funcionamiento de las mismas, produciendo que más de un millar de éstas centrifugadoras quedaran dañadas, y necesitando ser reemplazadas ante el desconcierto de los técnicos iraníes. Con un dispositivo unidireccional como el aquí descrito, **Figura 2-7**, el malware habría sido incapaz de penetrar en el *software* de control de la central nuclear.



Figura 2-7. Funcionamiento del diodo de datos a la inversa en entornos industriales

Los estándares internacionales marcados por[12] la *International Electrotechnical Commission* (IEC), *International Society of Automation* (ISA) y el *National Institute of Standards and Technology* (NIST), recomiendan que para la automatización industrial, así como los muros, vallas y puertas que protegen los perímetros físicos, se protejan los segmentos de la red industrial. El *firewall* es la tecnología más comúnmente utilizada pero debido al incremento de los ataques cibernéticos, cada vez más estas organizaciones están recomendando la combinación de *firewall* y diodo de datos para proteger la información generada y prevenir la infección mediante cualquier malware. Actualmente el diodo de datos, es la única tecnología digital, que permite y asegura que el tráfico de red se realice en una única dirección, al desconectar físicamente la salida de información de la parte sensible.

2.2.2.2 Protocolos y medidas

La unidireccionalidad de la transmisión en un diodo de datos restringe la utilización de los protocolos tradicionales de transmisión. El protocolo TCP, empleado actualmente para la inmensa mayoría de las aplicaciones, es un protocolo que establece una conexión bidireccional entre transmisor y receptor. Aunque la transmisión se realice en un único sentido, TCP envía los denominados “ACK” paquetes de asentimiento o “acuse de recibo” y control del flujo de información en sentido contrario, del receptor al transmisor. En el propio establecimiento de la conexión, TCP abre una comunicación bidireccional, **Figura 2-8**, en el que cada extremo le indica al otro un número inicial de secuencia que se envía en el segmento de sincronización y el otro responde con un mensaje de asentimiento (ACK) momento en el que empieza el envío de datos.

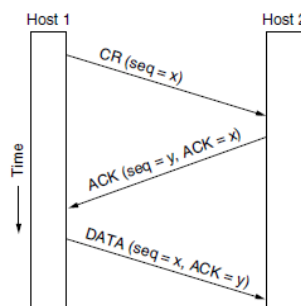


Figura 2-8- Triple “Hand-Shake” [2]

Por lo tanto, el protocolo TCP no se puede utilizar a través del enlace unidireccional del diodo según el diseño de transmisión propuesto[13]. Por ello, son necesarios dos proxies, antes y después del enlace unidireccional, de manera que los datos se transmitan hasta el proxy del emisor mediante TCP (bidireccional), atraviesen el diodo de datos (unidireccional), y se descarguen desde el segundo proxy con TCP (bidireccional), como se puede ver en la **Figura 2-9**.

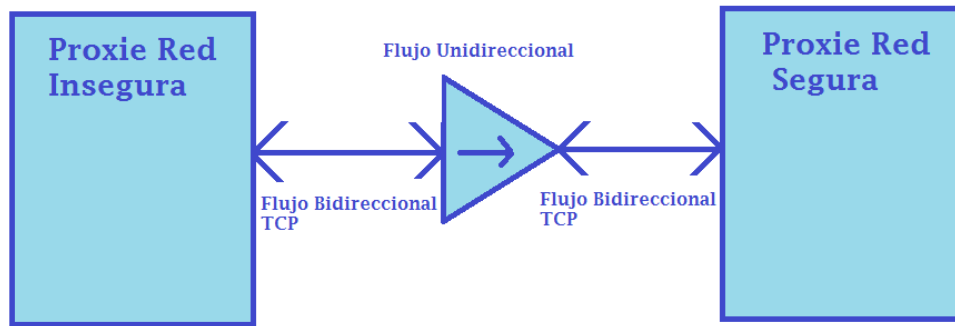


Figura 2-9. Diodo de datos

El protocolo que se adecúa a la transmisión en un único sentido es UDP que, a diferencia de TCP, no está orientado a la conexión y soporta la transmisión unidireccional. Además es preciso considerar que algunos protocolos auxiliares de bajo nivel como el protocolo de resolución de direcciones, *Address Resolution Protocol*, (ARP), utilizado para la resolución de direcciones, también emplean comunicación bidireccional. Para solventar este inconveniente se puede configurar en el emisor estáticamente la dirección *Media Access Control* (MAC) del receptor de modo que no utilice ARP para resolver dicha dirección MAC.

Otro problema al que hay que hacer frente son los errores de transmisión, que propician la pérdida de paquetes durante la misma. En los enlaces bidireccionales se tratan mediante los mensajes de asentimiento ACK y los mecanismos de retransmisión asociados de paquetes perdidos o dañados. No obstante, en un enlace unidireccional, donde no hay realimentación entre emisor y receptor, se hace especialmente importante su detección y corrección. Las técnicas de detección y errores se basan en la adición de información redundante en la emisión. En el primer caso, se incluye suficiente redundancia para que el receptor deduzca que se ha producido un error, pero no qué error, para solicitar una retransmisión de la información. En el segundo, la información redundante es suficiente para permitir al receptor recuperar la información que le ha sido transmitida aunque se produzcan errores.

En redes convencionales se utilizan tanto los códigos de corrección de errores, *forward error correction* (FEC), **Figura 2-10**, como los códigos de detección de errores en diferentes ámbitos. En nuestro caso, la imposibilidad de una comunicación entre emisor y receptor nos obliga a la adopción de códigos de corrección de errores, estableciéndose que un porcentaje de la información será enviada de manera redundante a través del diodo de datos.

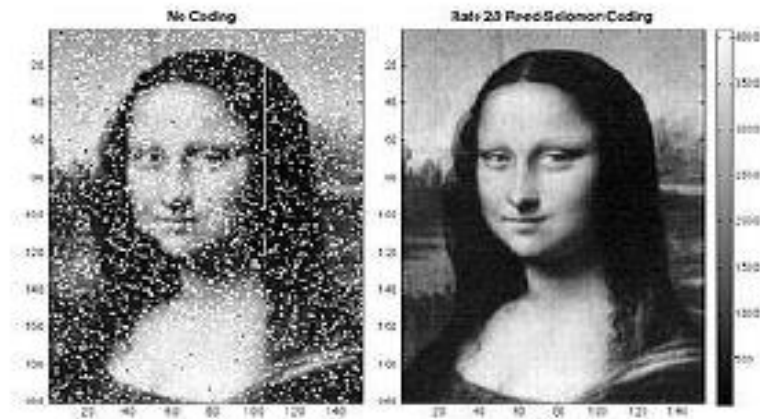


Figura 2-10. Ejemplo uso FEC

Una vez definido lo que es un diodo de datos y sus características principales, su funcionamiento completo constaría de las siguientes fases[14]:

1. Transmisión de datos: Incluyendo opcionalmente cifrado y otras medidas para evitar errores como paridad de *bits*, *checksums*, códigos de detección y corrección de errores. Estos datos transmitidos (señales eléctricas) se convierten en señales ópticas en el diodo.
2. Recepción de datos: El fotodiodo recibe la señal óptica convirtiéndola de nuevo en eléctrica.
3. Retransmisión y recepción de datos repetidos: Ante la imposibilidad de un *handshaking*, “apretón de mano” o ACKs, se reenvían los datos repetidas veces o se emplean códigos con mucha redundancia.
4. Detección de errores en la transmisión o recepción de los datos y emisión de una señal de aviso si se introdujo algún error en las repetidas transmisiones/recepciones.
5. Almacenamiento de los datos recibidos (si no hay ningún error) en un servidor u ordenador seguro.

2.2.2.3 Productos y aplicaciones

La necesidad de permanecer seguro y mantener la inviolabilidad de la información ha propiciado el desarrollo de estos sistemas unidireccionales en diferentes ámbitos que van desde proyectos públicos o de defensa con grandes inversiones económicas, empresas dedicadas a la venta de productos específicos de seguridad o desarrollos artesanales o de investigación de bajo coste en universidades.

La mayoría de los diodos de datos disponibles en el mercado tienen un impresionante conjunto de certificados de seguridad de nivel superior de países de todo el mundo. [12]El *National Institute of Standards and Technology* (NIST), certifica y asegura que esta unidireccionalidad forzada por medios *hardware* a nivel físico es un método muy eficiente como mitigación de amenazas.

En este apartado se exponen algunos productos y proyectos más relevantes para el ámbito de defensa y de las Fuerzas Armadas y la finalidad de la aplicar esta tecnología con fines militares.

- The pump

El proyecto *The pump*[15] , **Figura 2-11**, fue desarrollado por el Laboratorio de Investigación Naval, ubicado en Washington DC a partir de 1996. A pesar de no utilizar un diodo de datos para garantizar la unidireccionalidad, el fin a alcanzar es el mismo, es decir, la conexión de forma segura de dos redes de diferentes clasificaciones de confidencialidad, sin tener un intercambio de datos bidireccional.

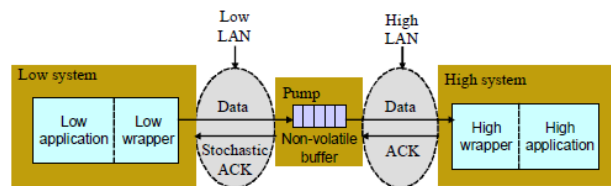


Figura 2-11. Funcionamiento

La principal diferencia es la creación de un *buffer* intermedio, **Figura 2-12**, entre las redes *Low* y *High* en la que se almacenan los datos que van a ser intercambiados, de manera que no hay intercambio de ACKs entre ambas redes. Este método garantiza la recepción de los datos, sin necesidad de métodos de corrección de errores ni el envío repetido de la información y proporciona también protección frente a una caída de la conexión que en el diodo de datos convencional propiciaría la pérdida de la información que en ese momento atraviesa el dispositivo.

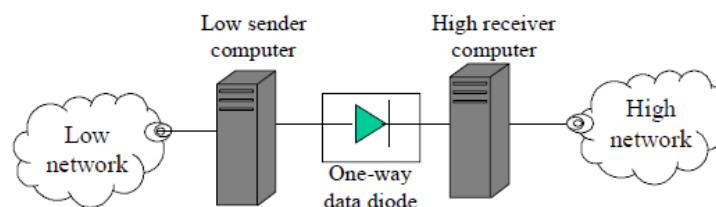


Figura 2-12. Enlace unidireccional con diodo de datos

The pump está acreditado por la Armada estadounidense y está siendo producido y distribuido bajo la patente Network PumpTM (US Patent application, 10/627,102, 25 de Julio de 2003.)

- Owl

La empresa americana Owl Computing Technologies, Inc.[16] Se encarga del diseño y venta de diodos de datos en materia de seguridad para el gobierno e instituciones militares de los Estados Unidos, además de diferentes sistemas al sector industrial. Con más de 2000 productos disponibles es una de las marcas que lidera este sector.

Entre las características de ciberseguridad que prometen los productos de esta compañía están la capacidad de transferir información de manera unidireccional con absoluta confidencialidad de los

datos, la transferencia de información desde varios orígenes y con varios tipos de datos a través de un único sistema *Owl DualDiode* y la integridad de la transferencia de datos con el *hardware Owl* certificado Common Criteria EAL-4. Todo ello aprobado por la empresa de certificación alemana TUV.

El sistema *Owl* básico, *hardware y software*, **Figura 2-13**, está compuesto por un diodo dual, con dos tarjetas de comunicación, solo envío y solo recepción por separado, para asegurar la transmisión unidireccional de datos. Se garantiza que ningún tipo de información, incluyendo protocolos de enlace (TCP/IP, SCSI, puerto serie/paralelo, etc.) pasa de la red de alta seguridad, *high*, a la de baja seguridad, *low*.

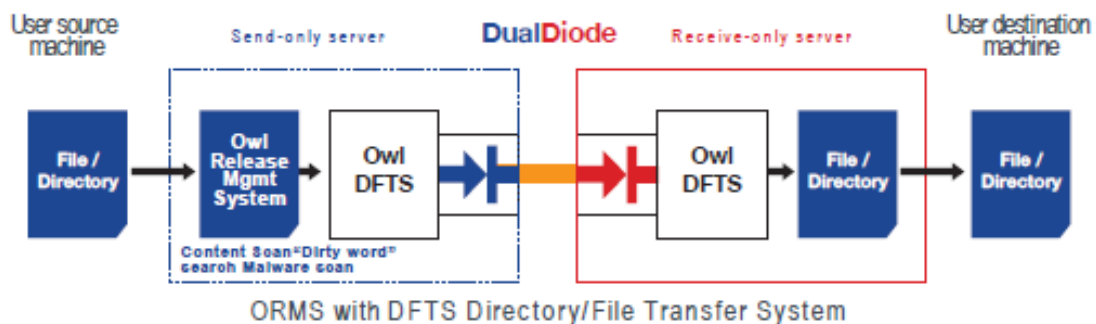


Figura 2-13. Diodo de datos OWL

De la gran variedad de oferta, el más versátil y aplicable a cualquier tipo de unidad militar que esté operando, es un diodo de datos de tamaño reducido, fácilmente transportable y ruggedizado diseñado para resistir las altas exigencias a nivel táctico de un combatiente.



Figura 2-14. Owl Miniaturized Cross Domain Solution (MCDS)

El llamado *Owl Miniaturized Cross Domain Solution* (MCDS), **Figura 2-14**, [17] tiene la función de conectar dos redes de diferente clasificación de seguridad, de manera unidireccional.

Está diseñado con interfaces USB, tanto para el envío como para la recepción de datos, que están aislados físicamente por un acoplador óptico. Esta tecnología USB permite la fácil manipulación y el envío de información desde portátiles o PDAs.

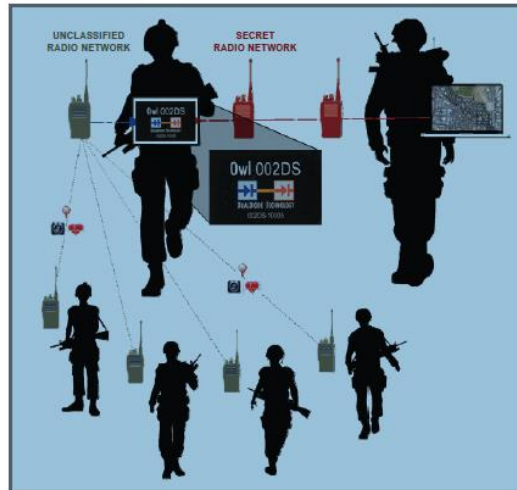


Figura 2-15. Situación táctica

La flexibilidad de este dispositivo permite, como vemos en la **Figura 2-15**, la capacidad de interconexión entre la red radio de combate, sin clasificar, de cualquier tipo de unidad de infantería, con otra que sí tendría clasificación, por ejemplo un puesto de mando.

- Data Diode FOX IT

La empresa de ciberseguridad industrial *FOX IT*[18], **Figura 2-16**, ha desarrollado y comercializa un diodo de datos para la transmisión segura de la información en entornos industriales, conectando los sistemas de control industrial que generan una información que puede ser crítica, como por ejemplo en los datos generados por las centrifugadoras de uranio de una central nuclear, con la red de oficinas que administran estos sistemas de control.

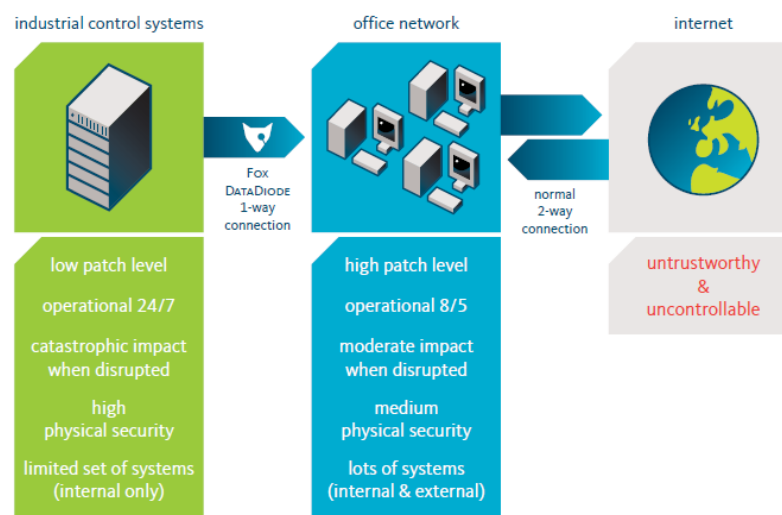


Figura 2-16. Diodo de datos *FOX IT*

El diodo desarrollado por *FOX IT* cuenta con diferentes estándares de seguridad aprobados por las organizaciones que manejan gran parte de la información clasificada como, NATO (*Secret*), NATO *Green Scheme Evaluated* o NL-NCSA (*Secret*).

○ Sina One Way

La empresa alemana Secunet[19] también desarrolla un diodo de datos, **Figura 2-17**, de transmisión unidireccional habilitado para la transmisión de información confidencial en redes, hasta la categoría de seguridad más alta, secreto o *NATO Secret*.

Denomina las dos partes como: una red abierta o sin clasificación de seguridad, negra, y una de alto nivel de clasificación llamada red roja. Así la información sin clasificar obtenida de Internet puede ser transferida a la red roja, habilitada como secreta.



Figura 2-17. Sina One Way

Este diodo de datos incluye un protocolo automático de detección y corrección de errores, pero en ningún caso existe un canal de retorno desde el receptor. La transmisión de datos únicamente ocurre en una dirección, de la red negra a la roja. Así, entre otros usos esta empresa destaca la transmisión desde Internet de resultados de búsqueda, los datos de registro del sistema y de auditoría correspondientes a la supervisión de actualizaciones de seguridad.

Las clasificaciones de seguridad con las que cuenta este producto se pueden ver en la siguiente figura vienen expuestas en la **Figura 2-18**.

		"Red" target system or network		
		NATO RESTRICTED	NATO CONFIDENTIAL	NATO SECRET
		 RESTRICTED	 CONFIDENTIAL	 SECRET
"Black" source system or network	NATO CONFIDENTIAL		—	
	NATO RESTRICTED	—		
	unclassified			
	open			

Figura 2-18. Clasificaciones Sina

Entre las características de este producto podemos destacar una velocidad de transmisión de 945 Mbits/segundo y una capacidad de almacenamiento de hasta 3 TB.

- High Speed Data Diode

El Diodo de datos de Alta Velocidad NC3A[20], *High Speed Data Diode* con sus siglas en inglés HSDD, fue desarrollado para proporcionar un producto rentable y eficiente que facilitara el automatizado de la importación y transmisión de datos específicos sensibles a los requerimientos de las operaciones militares de la OTAN, desde una red de menor nivel de clasificación de seguridad (*Unclassified and Public Networks*) a una de alto nivel (*NATO Restricted*, *NATO Confidential* o *NATO Secret*).

El HSDD es un regulador de flujo en una única dirección que utiliza una solución *hardware* para proporcionar una conexión unidireccional entre estos sistemas clasificados y no clasificados. Se compone de tres elementos:

- Un elemento de menor seguridad, *low side*, que transmite información de la parte baja de la CEI, *Communications Electronic Instructions*, de la OTAN hacia el elemento de mayor confidencialidad, *high side*.
- Un elemento de seguridad de un solo sentido (HSDD), **Figura 2-19**, que asegura que la información fluya solamente hacia la parte segura de la red.
- Y un elemento de alta seguridad que recibe la información a través del elemento anterior de un solo sentido, y empuja la información al dominio NR (*NATO Restricted*), NC (*NATO Confidential*), o NS (*NATO Secret*) de la CEI de la OTAN.



Figura 2-19. HSDD

El *hardware* implementado por la OTAN consiste en un dispositivo de un solo sentido de flujo con un mecanismo de control en la capa 1, o capa física. Al igual que el proyecto aquí desarrollado, cuenta con un convertidor de Ethernet a fibra óptica, dentro de un sistema seguro a prueba de manipulación y que permite la transmisión de los datos en la capa 2, o capa de enlace.

- Tinfoil chat

Markus Ottela de la Universidad de Helsinki, desarrolló en su TFG el *Tinfoil chat*, *Instant messaging with endpoint security* [21], un sistema de mensajería instantánea o chat de alta fidelidad construido sobre *hardware* libre y *software* de código abierto, que protege los datos en tránsito contra ataques activos y pasivos. En resumen, el proyecto debía de realizar las siguientes acciones para garantizar la confidencialidad de la información:

1. La información se encripta de forma simétrica con diferentes cifrados como AES256-GCM.
2. Se realiza la autenticación con contraseña de un solo uso con MAC o un conjunto de algoritmos: HMAC-SHA2-512.
3. Las claves de encriptación y de autenticación se generan mediante un *hardware* de números aleatorios.
4. Los extremos están asegurados por diferentes cifrados y descifrados en dos *trusted computin base* (TCB), un sistema computacional con un ambiente seguro, que interactúa con la red a través de un diodo de datos que crea un enlace unidireccional entre ambos canales. La eliminación del canal bidireccional previene de la exfiltración de claves y de ataques remotos independientemente del *software* utilizado, introducido mediante las denominadas puertas traseras.
5. Por último se ocultan los metadatos sobre cuándo se produce la comunicación y cuál es el flujo de datos, mediante el envío constante de datos cifrados a la recepción de las unidades TCB.

Ottela realizó la construcción de un diodo de datos a partir de componentes electrónicos (resistencias, diodos transistores, etc.) montados sobre una protoboard, como se indica en la **Figura 2-20**.

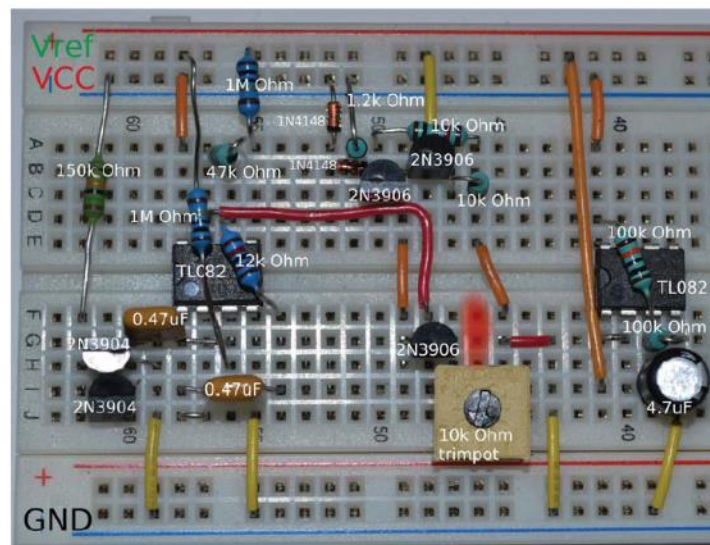


Figura 2-20. Montaje sobre protoboard

Posteriormente desarrolló el *software* para su chat seguro, con fuentes abiertas o libres, como RPi, Raspbian, Python, genkey, etc., llegando a implementar un *software* propio descargable en <https://www.cs.helsinki.fi/u/oottela/TFC-pub.key>. Este proyecto, a pesar de contar con una enorme complejidad es el que está mejor desarrollado y explicado de todos los analizados.

- Cimation, diodo de datos con Open Source

Animado por la gran eficacia que pueden proporcionar dos tarjetas de red ópticas, un cable de fibra óptica, dos ordenadores haciendo de proxies y un *software* libre de protección de errores, Austin Scott realizó un montaje artesanal de un diodo de datos basando en la simplicidad de los componentes citados.

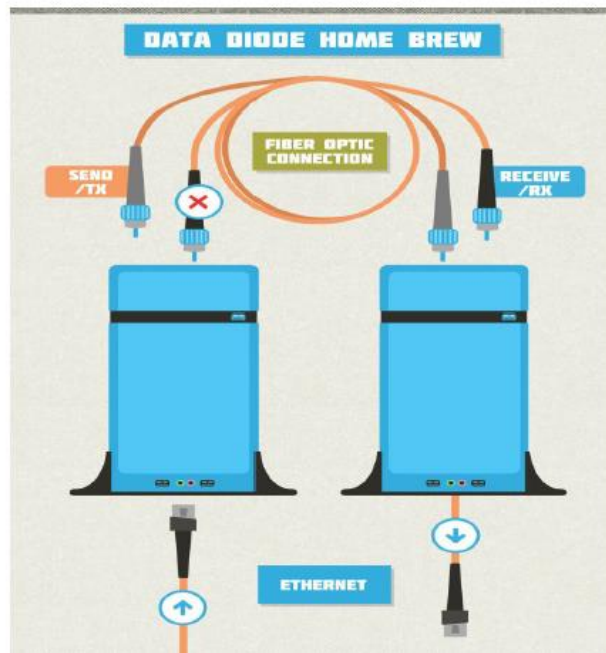


Figura 2-21. Esquema de diodo de datos

En su blog[22], se expone el esquema general de montaje, **Figura 2-21**, pero sin llegar a dar detalles específicos. Únicamente se enumeran los diferentes elementos necesarios para realizar el dispositivo, citando las características y precios de los elementos empleados. El *software* utilizado, libre y gratuito, para la protección frente a pérdidas y errores mediante la creación de redundancia fue *OpenBSD*, un sistema operativo libre tipo UNIX multiplataforma. El coste total del desarrollo ascendió a 1612 US\$.

3 DESARROLLO DEL TFG

Tras la revisión del estado del arte, en esta sección se describen los pasos seguidos para el desarrollo del sistema de transmisión unidireccional, desde el diseño de la arquitectura, la instalación y configuración de los proxies en las Raspberries y su interconexión mediante fibra, la configuración de las interfaces de red de cada proxy, la instalación de los diferentes programas y la creación de scripts que controlen la transmisión de la información.

3.1 Componentes de la arquitectura

Tras analizar, en el estado del arte, las diferentes clasificaciones de seguridad en las Fuerzas Armadas, las necesidades de la Armada, los sistemas de protección de información y sus características y los productos y desarrollos experimentales, se han determinado los componentes *hardware* y elementos *software* que se requieren para desarrollar el diodo de datos de este proyecto.

La solución propuesta está compuesta de dos proxies, cada uno materializado por una RPi, conectándose distintamente a su conversor de fibra óptica para el establecimiento del enlace unidireccional entre ambos proxies. A nivel *software* se ha empleado el protocolo UDP, con el programa de Linux UDPcast, y códigos de protección de errores (FEC) incluidos en este programa.

3.1.1 Proxies

Se requieren dos proxies, uno para la red de bajo nivel y el otro para la red de alto nivel de seguridad, que maneja documentos de alta confidencialidad.

En nuestro caso, ambos proxies estarán formados por *hardware* libre, *RaspberryPi*[23], un ordenador de bajo coste del tamaño de una tarjeta de crédito que se conecta a un monitor de ordenador o la televisión y utiliza teclado y ratón estándares. Tiene capacidad para permitir que personas de todas las edades exploren en el mundo de la computación y aprendan a programar en lenguajes como Scratch y Python. Lo más destacable es su capacidad de realizar lo que se puede esperar de un ordenador de escritorio, como navegar por Internet, reproducir video, jugar a videojuegos, etc. Se hace especialmente atractiva para la docencia, por sus reducidas dimensiones y precio.

Para nuestro proyecto se ha contado con dos Raspberries +B, **Figura 3-1**. Las especificaciones técnicas de cada RaspberryPi son:

- Memoria RAM de 512 MB
- Cabezal GPIO de 40 pines
- 4 puertos USB
- Ranura microSD
- Salida HDMI
- Salida audio jack 3,5mm
- Salida Micro USB para alimentación 5V/600mA (3.5 W)

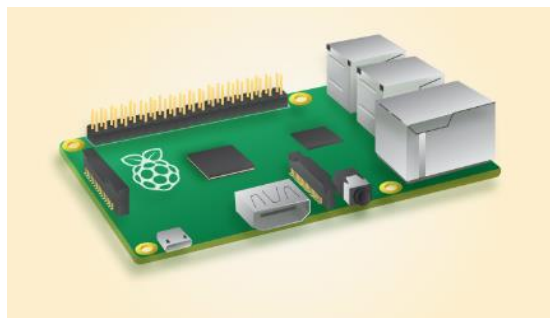


Figura 3-1. RaspberryPi B+

Como muestra la **Figura 3-2**, cada proxy está conectado a su propia red, de bajo y alto nivel de seguridad, por un cable Ethernet, ejecutando ambos un servidor FTP, y conectados también por una segunda interfaz Ethernet al convertor a fibra óptica.

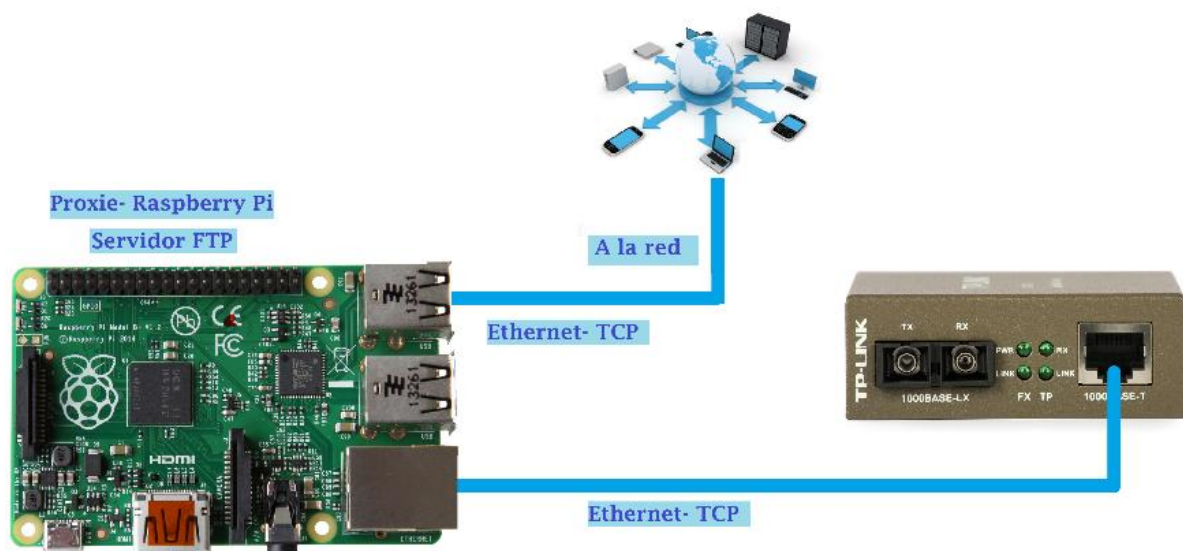


Figura 3-2. Proxy RaspberryPi

3.1.2 Conversores a fibra óptica

El conversor Ethernet-Fibra Óptica utilizado ha sido el TP-LINK MC100CM, **Figura 3-3**, con las siguientes características [24] :

- Velocidad elegible de 10 o 100 Mbits/segundo.
- Modo de transferencia Medio Dúplex o Dúplex Completo para el puerto FX.
- Amplía la distancia de fibra de hasta 2 km.



Figura 3-3. Convertidor Multimedia Multi-modo de 10/100Mbps

Se han empleado dos conversores que se comunican con su proxy por cable Ethernet, mediante protocolo TCP. De los conversores sale una única vía unidireccional de fibra que conecta ambos.

3.1.3 UDP y UDPcast

El protocolo de transmisión de datagramas (UDP), permite la transmisión de información sin que haya una comunicación bidireccional. A diferencia del protocolo de control de la transmisión (TCP), no está orientado a conexión, por lo que no hay una comunicación recíproca entre el transmisor y el receptor. En nuestro caso, únicamente el emisor envía información, utilizando para ello el canal de fibra óptica unidireccional.

Ante la imposibilidad de utilizar el protocolo TCP para el control de la transmisión de la información, se ha decidido utilizar una aplicación basada en (UDP), concretamente UDPcast [25], una herramienta de transmisión que permite el envío de datos simultáneamente a distintos receptores en una red local, *Local Area Network* (LAN) y proporciona métodos de corrección de errores (FEC). En nuestro caso solamente vamos a tener un receptor, pero la aplicación brinda la posibilidad de realizar la transmisión de manera unidireccional a varios destinatarios (*multicast*) en otros escenarios.

3.1.4 FEC

Los métodos *forward error correction* (FEC) introducen redundancia o duplicación de la información con la finalidad de proteger una comunicación unidireccional frente a errores. Esto va a ser de especial utilidad en el proyecto ya que al no haber retransmisiones entre emisor y receptor la tasa de errores puede ser muy acusada.

Para proteger al dispositivo frente a pérdidas por errores, interferencias, atenuación, fibra dañada, etc., hay que configurar la aplicación UDPcast para utilizar un mecanismo de corrección de errores. Este mecanismo tiene por objetivo la introducción de redundancia para compensar la pérdida de paquetes en tránsito de modo que el receptor pueda recomponer la información completa. Como se trabaja en modo unidireccional esta es la única forma de poder tratar la pérdida de paquetes. Este mecanismo corrige las pérdidas causadas por las ráfagas de paquetes, fragmentándolas e intercalando paquetes redundantes.

Con `--fec` “*intercalado*” $\times$ “*redundancia*” / *tamaño de la trama FEC*, [26] se indica el número y frecuencia de intercalado de paquetes redundantes. Este algoritmo está diseñado de tal manera que si se transmiten R paquetes redundantes se puede compensar la pérdida de cualquier paquete que esté en el mismo grupo FEC (*stripe* o trama):

- **Intercalado:** determina la cantidad de tramas en las que se fragmentan los datos. Un intercalado alto mejorará la robustez frente a pérdidas en la misma ráfaga. No aumenta la robustez frente a la pérdida aleatoria de paquetes en distintas tramas. Así mismo, un tamaño mayor de 8 va a reducir el ancho de banda.
- **Redundancia:** Indica cuantos paquetes extra se incluyen en cada trama, de manera que cuanto más alto sea, más redundancia se introduce, lo que se traduce en una mayor transmisión de datos y un aumento del tiempo de envío.
- **Tamaño de la trama (*stripesize*):** Establece la cantidad de bloques de datos que hay en cada trama. No pudiendo ser mayor de 128, la reducción del tamaño de trama es una forma indirecta de introducir redundancia (relativa), sin que se produzca una penalización de CPU por redundancia absoluta.

3.1.5 FTP

Para la conexión entre el proxy, la RPi, y los clientes, se ha utilizado el servidor FTP de Linux, *very secure file transfer protocol daemon* (*vstftpd*), para la compartición de archivos entre los diferentes elementos de la red, servidor-clientes.

3.2 Diseño de la solución

La arquitectura general del diodo de datos queda de la forma indicada en la **Figura 3-4**.

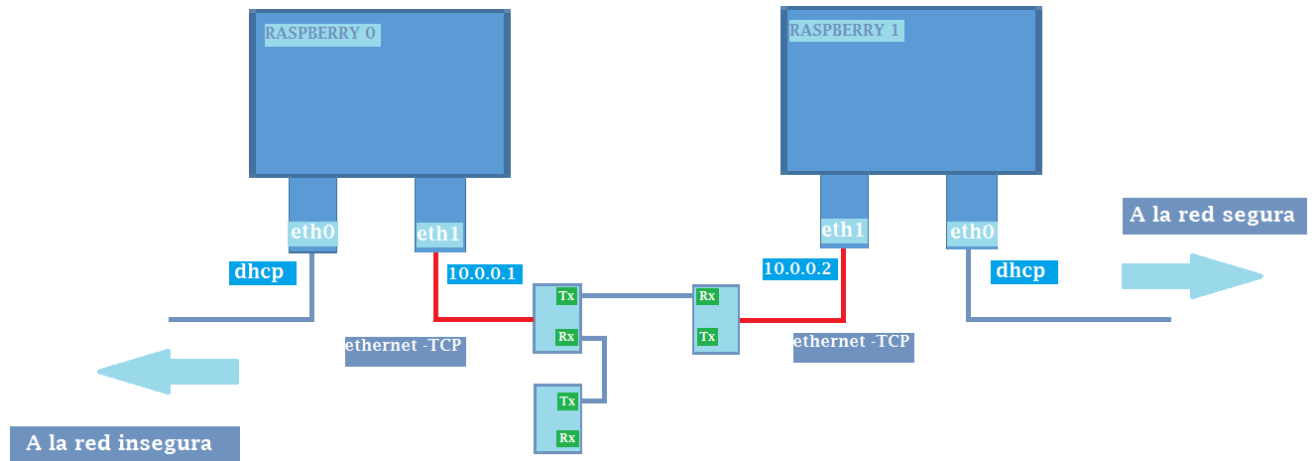


Figura 3-4. Montaje del diodo de datos

- Cada RPi hace las veces de proxy de su red correspondiente, la RPi 0 de la red insegura y la RPi 1 de la red con información sensible.
- La configuración de red de cada proxy se ha establecido con los interfaces de red, uno basado en, *Dynamic Host Configuration Protocol* (dhcp), para el enlace proxy-red y una IP estática, 10.0.0.1 y 10.0.0.2 respectivamente de cada proxy, con el otro.
- Los conversores a fibra reciben el cable Ethernet de cada proxy y convierten la señal eléctrica de este en señales de luz que pasan en una única dirección, desde la red insegura a la segura.
- Además, como se explicará más adelante se hace necesaria la presencia de un convertidor *dummy*, que va a la entrada (RX) del convertidor transmisor.

3.3 Configuración de los conversores y conexión mediante fibra óptica

Las redes de fibra óptica modernas son bidireccionales y típicamente utilizan dos circuitos ópticos, uno para cada dirección. Estas redes pueden ser convertidas en unidireccionales desconectando uno de los circuitos, es decir, dejando una única vía de un único sentido.

El diodo de datos asegura esta unidireccionalidad de la información mediante transceptores de fibra óptica y dos servidores o proxies, configurados con aplicaciones específicas sobre distintos protocolos. Los proxies, **Figura 3-5**, se encuentran antes y después del enlace unidireccional, de manera que los datos se transmiten hasta el proxy mediante TCP (bidireccional), atraviesan el diodo de datos (unidireccional), con la protección del código y se descargan en el segundo proxy con TCP (bidireccional).

Un diodo de datos consiste en un circuito en el que solo transmite el equipo fuente, TX, y la información únicamente llega al equipo receptor determinado. Normalmente, se utiliza un diodo emisor de luz o *Light Emitting Diode* (LED) en el emisor para convertir los pulsos eléctricos en pulsos de luz. Estos pulsos de luz viajan a través de un cable de fibra óptica hasta el receptor, que cuenta con un fotodiodo, que trabaja a la inversa, convirtiendo los pulsos de luz en pulsos eléctricos.

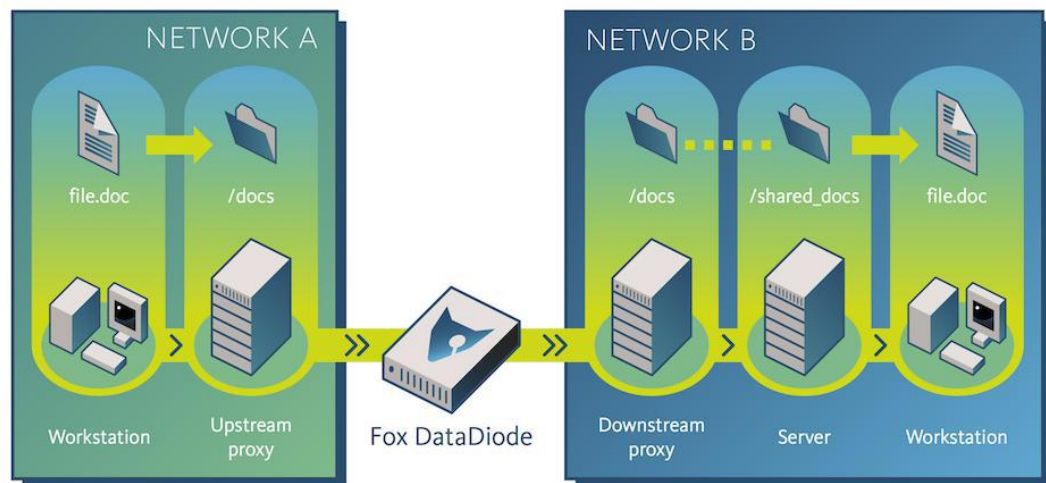


Figura 3-5. Diodo de datos FOX IT[18]

En nuestro caso se va a desarrollar el montaje mostrado en la **Figura 3-4**, cada RPi ejerce de proxy transmisor y receptor respectivamente. La información parte de la RPi 0 a través de la interfaz de red eth1 por un cable Ethernet hasta llegar al conversor emisor. Aquí hay que tener en cuenta que a la hora de realizar el montaje, es necesario montar las conexiones de fibra como se muestra en la **Figura 3-7**, en primer lugar y posteriormente alimentar los conversores. De la salida TX del conversor emisor sale un cable de fibra óptica al RX del conversor receptor, de manera que no hay un canal de retorno, la transmisión se realiza de manera unidireccional. Aquí, el conversor receptor reenvía la información por un cable Ethernet hacia la RPi 1, completándose el envío unidireccional de los datos.

No obstante, el conversor receptor empleado no es capaz de transmitir si no recibe una señal portadora en su terminal RX. Para solventar este inconveniente sin comprometer la unidireccional de la transmisión se conecta la recepción, RX, de este terminal al terminal TX de un tercer conversor denominado convertidor *dummy*. El convertidor *dummy* no recibe ni transmite ningún dato, su única función consiste en “engañar” al conversor de transmisión para que envíe los datos al conversor de recepción. Para que el *dummy* consiga “engañar” al transmisor se deben alimentar los convertidores una vez que está dispuesta la configuración mencionada, como se puede ver en la **Figura 3-6**.

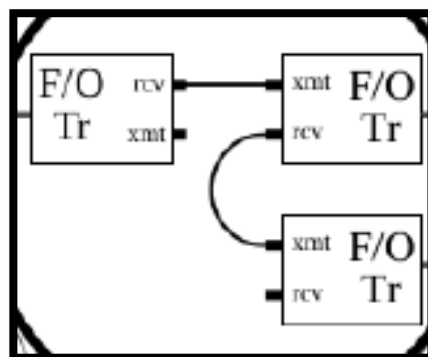


Figura 3-6. Configuración convertidores ópticos



Figura 3-7. Montaje de los convertidores

En cuanto a la configuración de los convertidores a fibra se ha dispuesto, como muestra la **Figura 3-8**, de la siguiente manera:

- TP_DIS: Desactiva el Modo Auto-Negociación, [27] que automáticamente establece el modo de transmisión, Full-Dúplex o Half-Duplex y la velocidad de transmisión.
- TP_100M: La velocidad de transmisión de 100 hasta Mbits/segundo.
- TP_HDX: Para que opere en modo Half-Duplex.
- LFP_OFF: Desactiva la función LFP (*Link Fault Pass-through*) [28] que desactivará el lado Ethernet cuando el enlace de fibra esté caído, y viceversa. Al estar en OFF nunca desactivará ninguno de los enlaces, aunque el lado de fibra no tenga conexión.

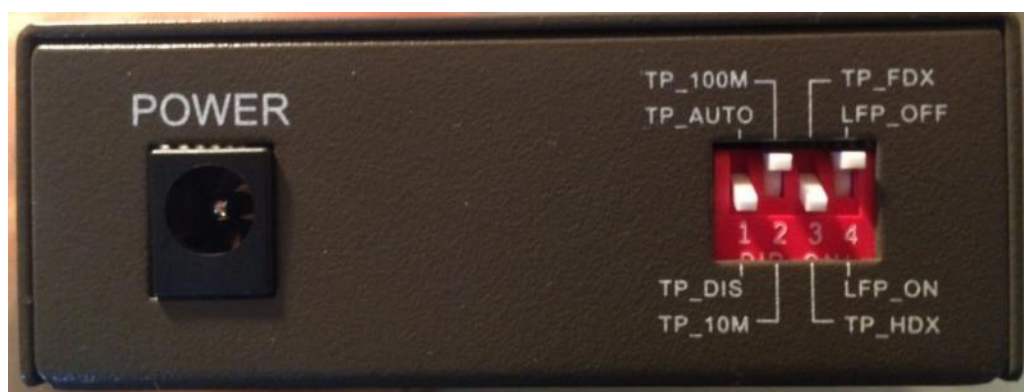


Figura 3-8. Configuración convertidor de fibra

3.4 Configuración de las RaspberryPi

3.4.1 Instalación Raspbian

Después de descargar la imagen del sistema operativo Raspbian Jessie, de la página web <https://www.raspberrypi.org/downloads/raspbian/>, con el *software* WIN32, **Figura 3-9**, se escribe el sistema operativo en la tarjeta de memoria SD, disco duro de la RPi.

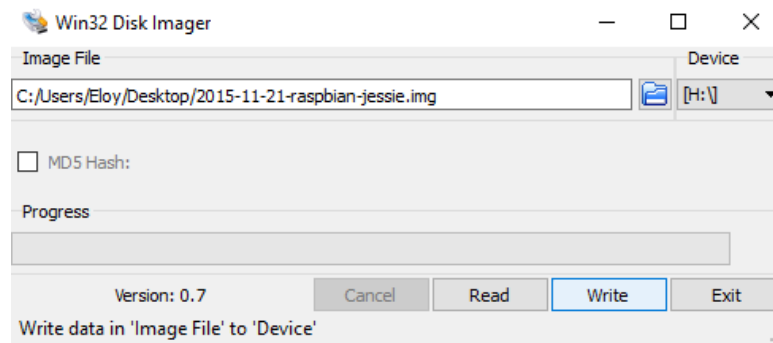


Figura 3-9. Escritura de Raspbian en tarjeta SD.

Una vez instalado el sistema operativo se introduce en cada tarjeta micro SD en la respectiva RPi y se lleva a cabo su primera ejecución. Con el terminal en modo texto, **Figura 3-10**, se ejecuta el comando *sudo apt-get -y update* para actualizar los repositorios.

```
pi@raspberrypi0:~ $ sudo apt-get -y update
```

Figura 3-10. Actualización de repositorios

Y a continuación, para actualizar a la última versión, **Figura 3-11**, se introduce *sudo apt-get upgrade*.

```
pi@raspberrypi0:~ $ sudo apt-get -y upgrade
```

Figura 3-11. Actualizar versión

El comando *uname -r*, **Figura 3-12**, indica que versión del sistema operativo está instalada. En este caso es la versión 4.1.13+.

```
pi@raspberrypi0:~ $ uname -r
4.1.13+
```

Figura 3-12. Versión Raspbian Jessie

Introduciendo *nano raspi-config*, **Figura 3-13**, se abre el terminal para la configuración de las funcionalidades de RPi, como el cambio de contraseña, huso horario, tipo de teclado, etc.

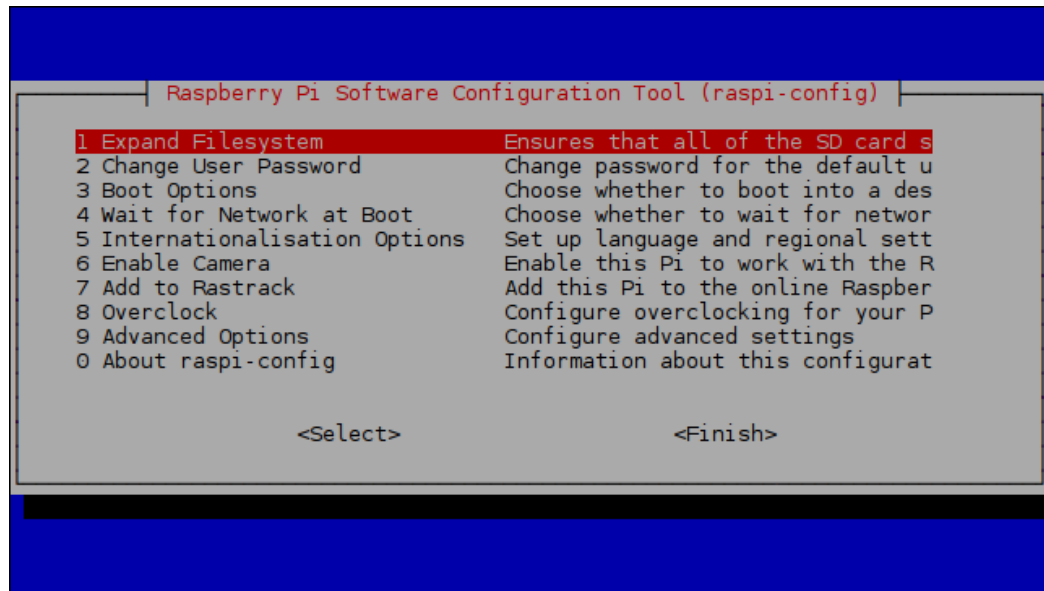


Figura 3-13. Interfaz de configuración raspi-config

3.4.2 Configuración de los interfaces de red

A continuación, se procede, en cada una de las dos Raspberries, a configurar los interfaces de red. Con el comando[29] *nano /etc/network/interfaces*, **Figura 3-14**, se accede al archivo de configuración de interfaces donde, o bien, se puede asignar una dirección IP fija o bien usar una dinámica con *dhcp*, también permite establecer la información de enrutamiento, configurar el enmascaramiento IP, poner las rutas por defecto, etc.

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto eth0

iface lo inet loopback

iface eth0 inet dhcp
address 169.254.0.1
netmask 255.255.255.0
gateway 192.168.1.1

auto eth1

iface eth1 inet static
address 10.0.0.1
netmask 255.0.0.0
```

Figura 3-14. Configuración de interfaces

Como se indica en la **Figura 3-15** la interfaz 0, eth0, va a ser la que esté conectada a la red, por lo que se deja en dhcp. Para la interfaz 1, eth1, será la utilizada para el enlace unidireccional por lo que sus respectivas IPs se establecerán manualmente, dándole 10.0.0.1 y 10.0.0.2 distintamente. También la máscara de red será 255.0.0.0 para ambas.

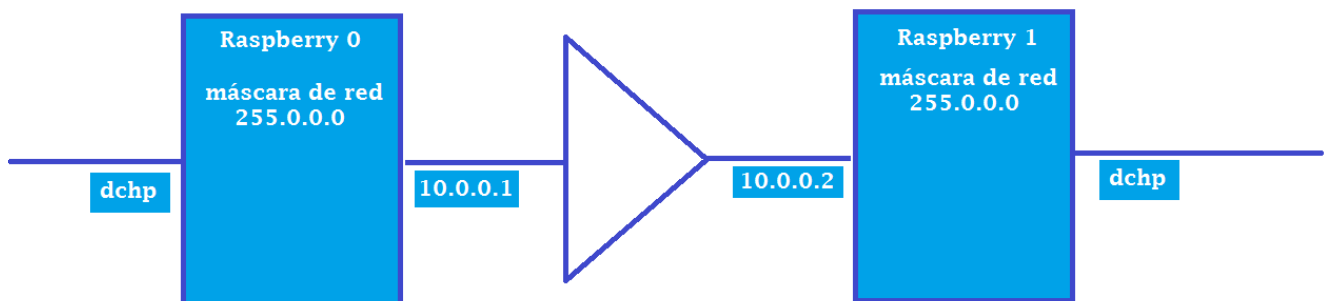


Figura 3-15. Configuración IP del dispositivo

3.4.3 Instalación de programas

En esta sección se desarrollan todos los pasos necesarios para la instalación de los diferentes programas a utilizar, así como su configuración inicial.

3.4.3.1 UDPcast

Para instalar la aplicación [25], con permisos de administrador, se introduce en cada RPi, *apt-get install udpcast*, **Figura 3-16**.

```
root@raspberrypi0:/home/pi# apt-get install udpcast
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Figura 3-16. Instalación UDPcast

3.4.3.2 Instalación Iperf

La aplicación Iperf, [30] es una muy útil herramienta cliente-servidor para, entre otras cosas, medir el ancho de banda máximo entre redes IP sobre varios protocolos como TCP, UDP, SCTP, lo que permite la optimización de parámetros, en este caso la optimización de la velocidad de transmisión que evite la pérdida de paquetes debidos al desbordamiento de colas. Para instalarla, **Figura 3-17**, `sudo apt-get install iperf`.

```
pi@raspberrypi0:~ $ sudo apt-get install iperf
Reading package lists... Done
Building dependency tree... 79%
```

Figura 3-17. Instalación Iperf

3.4.3.3 Servidor FTP

Se ha utilizado el servidor de Unix *very secure file transfer protocol daemon* (`vsftpd`), para crear un servidor FTP en cada proxy, RaspberryPi, para que permita el envío de información desde un cliente remoto. Con privilegios de administrador[31] la sintaxis para instalar la última versión del servidor FTP, **Figura 3-18**, es `sudo apt-get install vsftpd`.

```
root@raspberrypi0:/home/pi# apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Figura 3-18. Instalación vsftpd

Una vez que esté descargado se abre el archivo de configuración, con privilegios de administrador, con el editor de textos nano, **Figura 3-19**, introduciendo `nano /etc/vsftpd.conf`.

```
root@raspberrypi0:/home/pi# nano /etc/vsftpd.conf
```

Figura 3-19. Acceso a configuración vsftpd

Y se descomentan las líneas mostradas en la **Figura 3-20** para permitir la escritura de archivos a los usuarios de la RPi.

A terminal window with a black background and white text. The text shows configuration lines for vsftpd, specifically enabling local users to log in and enabling write commands. The lines are commented out with a hash symbol at the beginning.

Figura 3-20. Autorización de escritura en servidor FTP

Para hacer efectivos los cambios se reinicia el servicio, **Figura 3-21**, teniendo permisos root, con *service vsftpd restart*.

A terminal window with a black background and white text. The text shows a root prompt followed by the command to restart the vsftpd service.

Figura 3-21. Reinicio vsftpd

3.4.3.4 Conexión de las Raspberries entre sí.

Una vez que las dos Raspberries tienen instalado su sistema operativo, y sus aplicaciones configuradas, se realiza la primera conexión entre las mismas. Para ello la primera vez se ha utilizado un cable de red cruzado, únicamente con la finalidad de comprobar que las interfaces de red se han configurado correctamente.



Figura 3-22. Unión de las RPi con cable cruzado

Cuando se encuentran las dos Raspberries conectadas entre sí con el cable cruzado, **Figura 3-22**, se realiza un *ping*, desde la emisora a la receptora y se verifica que efectivamente que se “ven” y la RPi receptora responde el *ping*, mandándolo a la IP de la RPi con *sudo ping 10.0.0.2*

```
pi@raspberrypi0:~ $ sudo ping 10.0.0.2
```

Figura 3-23. Ping

También se prueba la configuración de UDPcast, **Figura 3-24**, introduciendo en la estación transmisora el comando *udp-sender -f fichero.txt --interface eth1*.

```
pi@raspberrypi0:~ $ udp-sender -f fichero.txt --interface eth1
```

Figura 3-24. Envío de información por UDPcast

Donde se indica el nombre del fichero a transmitir y la interfaz que se va a utilizar, en este caso la interfaz *eth1* que es la utilizada para el enlace unidireccional entre proxies.

De la misma forma, en la estación receptora, **Figura 3-25**, se introduce *udp-receiver -f fichero.txt --interface eth1*.

```
pi@raspberrypi0:~ $ udp-receiver -f fichero.txt --interface eth1
```

Figura 3-25. Recepción de información por UDPcast

3.4.3.5 Tabla de encaminamiento ARP

Debido a que no existe retroalimentación o *feedback* entre los dos terminales y por tanto no funciona el protocolo de resolución de direcciones, ARP, hay que indicar, de forma manual, **Figura 3-25**; al dispositivo transmisor la dirección MAC del dispositivo receptor asociada a su dirección IP, para que la incluya en la tabla ARP con *sudo arp-s 10.0.0.2 xx:xx:xx:xx:xx:xx*.

```
@raspberrypi0:/home/pi/Desktop# sudo arp-s 10.0.0.2 xx:xx:xx:xx:xx:xx
```

Figura 3-26. Añadir MAC/IP a tabla ARP

- -s indica que se introduce una dirección *Hardware* (MAC)
- 10.0.0.2 es la dirección IP de la RPi de recepción
- xx:xx:xx:xx:xx:xx es la dirección MAC de la RPi de recepción

Para verificar que la conexión entre ambos proxies de forma unidireccional con el cable de fibra óptica ha sido exitosa se comprueba con el programa nc, *netcat*, el cual permite establecer una conexión TCP o UDP entre las máquinas. En nuestro caso utilizaremos UDP.

Se ejecuta en el dispositivo receptor el comando de la aplicación *netcat* para establecer un enlace unidireccional, en este caso mandar un datagrama (UDP) escribiendo `nc -u -l -p 2000`.

```
root@raspberrypi0:/home/pi/Desktop# nc -u -l -p 20000
```

Figura 3-27. Netcat en Rx

- -u utiliza UDP, ya que por defecto el programa usa TCP.
- -l indica que este dispositivo queda a la escucha de una conexión entrante.
- -p indica el puerto a utilizar para esta conexión, en este caso el 20000 es el de TCP/UDP.

Y en el transmisor, **Figura 3-28**, `echo Hello | nc -u -n 10.0.0.2 20000`.

```
root@raspberrypi0:/home/pi/Desktop# echo Hello | nc -u -n 10.0.0.2 20000
```

Figura 3-28. Netcat en Tx

- -n no utiliza DNS, suprime la resolución nombre/puertos

Una vez ejecutado el comando anterior, si la conexión está correctamente establecida en el dispositivo receptor aparece, **Figura 3-29**, lo que se había establecido que se escribiera en el terminal.

```
root@raspberrypi0:/home/pi/Desktop#  
Hello
```

Figura 3-29. Enlace correcto Tx-Rx

Lo que quiere decir que el transmisor ha asociado la dirección IP del receptor con su dirección MAC, para realizar el envío de información a través del enlace unidireccional.

3.5 Transmisión de información de forma unidireccional

Una vez se ha instalado correctamente el circuito de fibra, se realizan las primeras pruebas de transmisión de información con la aplicación instalada con anterioridad, UDPcast. Para ello hay que introducir los comandos de transmisión y recepción añadiendo el comando `--async` en el transmisor y `--nosync` para el receptor de manera que el receptor no envíe un mensaje de asentimiento al recibir la

información. También **--nokbd**, para evitar tener que aceptar por teclado la operación de envío de información. Por ello el receptor debe estar a la escucha antes de que el transmisor comience la emisión de datos. Las **Figuras 3-30** y **3-31** muestran la sintaxis del programa UDPcast para receptor y transmisor distintamente.

```
pi@raspberrypi0:~ $ udp-receiver -f fichero --interface eth1 --async
```

Figura 3-30. Transmisión unidireccional receptor

```
pi@raspberrypi0:~ $ udp-sender -f fichero --interface eth1 --nosync --nokbd
```

Figura 3-31. Transmisión unidireccional transmisor

3.6 Limitación de tasa y FEC

Debido a que el programa UDPcast[32], por el hecho de ser unidireccional no permite el control de la congestión que adapte la tasa a la máxima disponible, cuando se envía información de gran tamaño se debe limitar la velocidad de envío para evitar que del desbordamiento de las colas de transmisión propicie el descarte y pérdida de paquetes.

Se añade **--max-bitrate tasa**, con la tasa en bits/segundo, **Figura 3-32**, para evitar la pérdida de paquetes por desbordamiento de las colas.

```
udp-sender -f fichero --interface eth1 --async -- nokbd --max-bitrate 35m
```

Figura 3-32. Udp-sender limitación de la tasa

Como ya se ha mencionado antes, UDPcast ofrece un mecanismo de protección de errores, que introducirá redundancia que protege el envío de información frente a interferencias, atenuación y otros errores.

La configuración del código, **Figura 3-33**, se realiza de la siguiente manera:

```
udp-sender -f fichero --interface eth1 --async --nokbd --max-bitrate 35m --fec 12x12/128.
```

```
udp-sender -f fichero --interface eth1 --async -- nokbd --max-bitrate 35m --fec 12x12/128
```

Figura 3-33. Udp-sender protección de errores

- **interface eth1**: indica que la salida de la información se hace por la interfaz eth1, que es la que va al conversor a fibra.
- **async**: el enlace se realiza sin establecimiento de conexión entre los proxies emisor-receptor.
- **nokbd**: no necesita la confirmación por teclado para iniciar el envío de información.
- **max-bitrate**: limitación de la tasa para evitar el desbordamiento y la pérdida de paquetes por colas.
- **fec 12x12/128**: método de protección frente a errores, introduce redundancia para robustecer el dispositivo frente a pérdidas aleatorias de paquetes.

3.7 Script

Una vez que se ha realizado con éxito la transmisión de información de forma unidireccional, hay que automatizar la transmisión entre las carpetas del emisor y del receptor que son accesibles desde ambas redes a través del protocolo FTP. Para ello se desarrolla un script que automatice la transmisión y recepción de información. En transmisión se ocupará de:

1. En primer lugar realiza una captura de los archivos que hay en la carpeta del servidor FTP y crea un archivo de texto con la lista de los mismos.
2. A continuación compara la lista de archivos de la carpeta con la lista de archivos transmitidos y aquellos que no estén en transmitidos los pasa a la cola para transmitir.
3. Manda por UDPcast primero un fichero de texto con el nombre del archivo y a continuación envía el archivo y lo guarda en la lista de elementos enviados.

En el receptor el script realiza lo siguiente:

1. Recibe el nombre del archivo en un fichero de texto.
2. Extrae el nombre con su extensión y lo utiliza para dar nombre al archivo que recibe.

Los pasos realizados para la configuración para los scripts transmisor y receptor son los detallados en los siguientes apartados.

3.7.1 Transmisor

1. Creación del script.

La creación y pruebas del script se han realizado sobre el Sistema operativo Ubuntu de Linux, en una máquina virtual. Ejecutando, **Figura 3-34**, desde el terminal de comandos `sudo nano rp0.sh`, se abre el editor de textos nano y crea un archivo nuevo, al que se introducen al que se introducen las líneas de código *bash* detalladas en el ANEXO II: SCRIPT TRANSMISOR.



```
pi@raspberrypi0:~ $ sudo nano rp0.sh
```

Figura 3-34. Creación script transmisor

Una vez que se ha está el código introducido, se guarda el código con `ctrl+O`, con su nombre (*rp0.sh*) y se sale con `ctrl+X`.

2. Copiar script en la carpeta de arranque:

Para que el script diseñado se ejecute de forma automática al iniciar la RPi, tiene que estar en la carpeta de inicio, **Figura 3-35**, para ello se copia *rp0.sh* con `sudo cp rp0.sh /etc/init.d/`:



```
pi@raspberrypi0:/etc/init.d $ sudo cp rp0.sh /etc/init.d/
```

Figura 3-35. Script init.d

3. Dar permisos de ejecución al script.

Una vez que el script *rp0.sh* ha sido creado, hay que configurarlo con permisos para la ejecución por el sistema, **Figura 3-36**. Se ejecuta el comando `sudo chmod+x rp0.sh` y para comprobarlo, ejecutando el comando `ls` se verifica que el script *rp0.sh* aparece en verde, **Figura 3-37**.

```
pi@raspberrypi0:/etc/init.d $ sudo chmod +x rp0.sh
```

Figura 3-36. Permisos de ejecución al script

```
pi@raspberrypi0:/etc/init.d $ ls
alsa-utils          hwclock.sh          plymouth            skeleton
avahi-daemon        kbd                  plymouth-log        ssh
bootlogs            keyboard-setup       procps              sudo
bootmisc.sh         killprocs            raspi-config        triggerhappy
checkfs.sh           kmod                 rc                   udev
checkroot-bootclean.sh lightdm              rc.local            udev-finish
checkroot.sh         motd                 rcS                  umountfs
console-setup        mountall-bootclean.sh README               umountnfs.sh
cron                 mountall.sh          reboot              umountroot
dbus                 mountdevsubfs.sh     rmnologin           urandom
dhcpcd               mountkernfs.sh       rp0.sh              vsftpd
dphys-swapfile       mountnfs-bootclean.sh rpcbind              x11-common
fake-hwclock         mountnfs.sh          rsync                xrdp
halt                 networking           rsyslog
hdparm               nfs-common            sendsigs
hostname.sh          ntp                   single
```

Figura 3-37. Script con permisos

4. Convertir el script en daemon:

Se convierte en daemon, **Figura 3-38**, para que se ejecute al inicio de la sesión, introduciendo el comando `sudo update-rc.d rp0.sh defaults`:

```
pi@raspberrypi0:/etc/init.d $ sudo update-rc.d rp0.sh defaults
```

Figura 3-38 Convertir el script en Daemon

3.7.2 Receptor

1. Creación del script.

La creación y pruebas del script se han realizado sobre el Sistema operativo Ubuntu de Linux, en una máquina virtual. Ejecutando, **Figura 3-39**, desde el terminal de comandos `sudo nano rp1.sh`, se abre el editor de textos nano y crea un archivo nuevo, al que se introducen las líneas de código bash detalladas en el ANEXO III: SCRIPT RECEPTOR.

```
pi@raspberrypi0:~ $ sudo nano rp1.sh
```

Figura 3-39. Creación script receptor

Una vez que se ha está el código introducido, se guarda el código con `ctrl+O`, con su nombre (`rp1.sh`) y se sale con `ctrl+X`.

2. Copiar script en la carpeta de arranque:

Para que el script diseñado se ejecute de forma automática al iniciar la RPi, tiene que estar en la carpeta de inicio, **Figura 3-40**, para ello se copia `rp1.sh` con `sudo cp rp1.sh /etc/init.d/`:

```
pi@raspberrypi0:~ $ sudo cp tfg.sh /etc/init.d/
```

Figura 3-40. Script init.d

3. Dar permisos de ejecución al script.

Una vez que el script *rp1.sh* ha sido creado, hay que configurarlo con permisos para la ejecución por el sistema, **Figura 3-41**. Se ejecuta el comando `sudo chmod +x rp1.sh` y para comprobarlo, ejecutando el comando `ls` se verifica que el script *rp1.sh* aparece en verde, **Figura 3-42**.

```
pi@raspberrypi:/etc/init.d $ sudo chmod +x rp1.sh
```

Figura 3-41. Permisos de ejecución al script

```
pi@raspberrypi:/etc/init.d $ sudo chmod +x rp1.sh
pi@raspberrypi:/etc/init.d $ ls
alsa-utils          hwclock.sh         ntp                 single
avahi-daemon        kbd                 plymouth            skeleton
bootlogs            keyboard-setup      Plymouth-log        ssh
bootmisc.sh         killprocs           procs               sudo
checkfs.sh          kmod                raspi-config        triggerhappy
checkroot-bootclean.sh lightdm             rc                  udev
checkroot.sh         motd                rc.local            udev-finish
console-setup        mountall-bootclean.sh rcS                  umountfs
cron                 mountall.sh          README              umountnfs.sh
dbus                 mountdevsubfs.sh     reboot              umountroot
dhcpcd               mountkernfs.sh       rmnologin           urandom
dphys-swapfile       mountnfs-bootclean.sh rp1.sh              vsftpd
fake-hwclock         mountnfs.sh          rpcbind             x11-common
halt                 networking           rsync               xrdp
hdparm               nfs-common           rsyslog
hostname.sh          nodered              sendsigs
```

Figura 3-42. Script con permisos

4. Convertir el script en daemon:

Se convierte en daemon, **Figura 3-43**, para que se ejecute al inicio de la sesión, introduciendo el comando `sudo update-rc.d rp1.sh defaults`:

```
pi@raspberrypi:/etc/init.d $ sudo update-rc.d rp1.sh defaults
```

Figura 3-43. Convertir el script en Daemon

Ya se tienen configurado ambos script para que se ejecuten al iniciar el sistema.

4 PRUEBAS Y VALIDACIÓN

En esta sección se describen las diferentes pruebas realizadas para obtener una velocidad adecuada y una protección frente a las a pérdidas a las que puede verse sometido el dispositivo unidireccional, con el fin de realizar la transmisión de información de forma segura y de la manera más eficiente.

4.1 Establecimiento de la tasa de transmisión

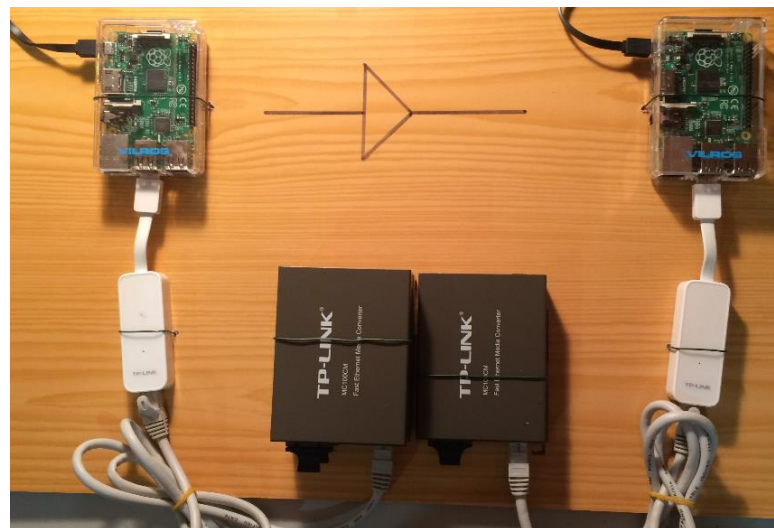


Figura 4-1. Montaje del diodo

Una vez realizada la conexión unidireccional, **Figura 4-1**, debido al uso de datagramas sin realimentación para registrar y corregir las pérdidas, al transmitir un archivo de 10 MB la tasa las pérdidas es enorme. Como se puede apreciar en la **Figura 4-2** apenas llegan 3 paquetes de más de mil.

```
root@raspberrypi:/home/pi# udp-receiver -f 139.mp3 --nosync --interface eth1
Udp-receiver 20120424
UDP receiver for 139.mp3 at 10.0.0.2 on eth1
Connected as #0 to 10.0.0.1
Listening to multicast on 234.0.0.1
bytes= 1 490 944 ( 58.21 Mbps)
Dropped by server now=3 last=0
Missed blocks: 127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,1
43,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,1
63,164,165,166,167,168,174,232,233,234,235,236,237,238,239,240,241,242,243,244,2
45,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,2
65,266,268,272,283,285,297,308,320,321,322,323,324,325,326,327,328,329,330,331,3
32,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,3
52,353,354,355,356,357,358,359,360,361,362,363,365,367,375,384,401,402,408,419,4
20,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,4
40,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,4
60,461,463,465,466,470,481,486,489,516,517,518,519,520,521,522,523,524,525,526,5
27,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,5
47,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,566,575,579,5
86,590,599,600,601,604,606,615,616,617,618,619,620,621,622,623,624,625,626,627,6
28,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,6
48,649,650,651,652,653,654,655,656,657,658,659,660,661,663,665,667,674,675,676,6
77,678,679,685,690,696,699,714,715,716,717,718,719,720,721,722,723,724,725,726,7
27,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,7
47,748,749,750,751,752,753,754,755,756,757,758,759,760,761,763,766,769,772,777,7
82,789,798,805,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,8
31,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,8
51,852,853,854,855,856,857,858,859,860,861,866,867,878,889,893,896,898,906,912,9
13,914,915,916,917,918,922,924,931,936,938,942,955,971,972,973,974,975,976,977,9
78,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,9
98,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1014,1016
Blocks received: 552/552/1024
Transfer complete.
```

Figura 4-2. UDPcast sin límite de velocidad

Se produce una saturación de las colas que propicia la pérdida de casi todos los paquetes, por lo que hay que establecer la velocidad óptima de transmisión e introducir redundancia para corregir los errores (FEC).

Para determinar una velocidad adecuada de transmisión que no desborde la conexión entre las dos Raspberries unidas con la fibra óptica de manera convencional (con ambos TX conectados a las terminales RX como muestra la **Figura 4-3**) se utiliza el programa Iperf.



Figura 4-3. Conexión bidireccional

Una vez realizada la interconexión hay que establecer la conexión entre ambos hosts, **Figura 4-4**, de manera que el dispositivo receptor va a funcionar como servidor y el dispositivo transmisor de cliente. Se introduce en el transmisor *iperf -u -s*.

```
pi@raspberrypi0:~ $ iperf -u -s
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 36609
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 3]  0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.049 ms   0/ 893 (0%)
```

Figura 4-4. Iperf transmisor

Y en el receptor, **Figura 4-5**, *iperf -u -c10.0.0.1*.

```
pi@raspberrypi:~ $ iperf -u -c10.0.0.1
```

Figura 4-5. Iperf receptor

De manera que se obtienen los datos de la conexión entre los hosts, mostrado en la **Figura 4-6**.

```
pi@raspberrypi0:~ $ iperf -u -s
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 10.0.0.1 port 5001 connected with 10.0.0.2 port 36609
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 3]  0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.049 ms   0/ 893 (0%)
```

Figura 4-6. Resultado Iperf

A la hora de determinar la velocidad o ancho de banda máximo sin que se produzcan pérdidas, se han ido probando diferentes tasas. Empezando con un ancho de banda de 70 Mbits/segundo, las pérdidas superan el 40%.

```
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 10.0.0.2 port 56930 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-30.0 sec  245 MBytes 68.4 Mbits/sec
[ 3] Sent 174537 datagrams
[ 3] Server Report:
[ 3]  0.0-30.0 sec  137 MBytes 38.3 Mbits/sec  0.228 ms 76823/174536 (44%)
[ 3]  0.0-30.0 sec  1 datagrams received out-of-order
```

Figura 4-7. Iperf 70 Mbits/segundo

Se puede ver, **Figura 4-7**, como el dispositivo no es capaz de hacer frente a esa velocidad y como consecuencia el ancho de banda efectivo se reduce a 38,3 Mbits/segundo y las pérdidas de paquetes suponen casi la mitad del total, un 44%. En dispositivos con capacidad de procesamiento como las Raspberries emplear una tasa muy superior a la proporcionada por el canal es contraproducente ya que se pierden más paquetes debido a la sobrecarga del procesador.

Si se reduce la velocidad, a 60 Mbits/segundo, **Figura 4-8**, el dispositivo casi es capaz de mantener esa velocidad, pero se siguen produciendo pérdidas de paquetes en un 2,6%.

```
root@raspberrypi:/home/pi# iperf -u -c 10.0.0.1 -b 60000000 -t 30
-----
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 10.0.0.2 port 55470 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-30.0 sec   213 MBytes   59.5 Mbits/sec
[ 3] Sent 151729 datagrams
[ 3] Server Report:
[ 3] 0.0-30.0 sec   207 MBytes   57.9 Mbits/sec   0.210 ms 3967/151728 (2.6%)
[ 3] 0.0-30.0 sec   1 datagrams received out-of-order
```

Figura 4-8. Iperf 60 Mbits/segundo

Tras hacer la prueba con diferentes anchos de banda, se ha concluido que la tasa límite queda alrededor de 50 Mbit/segundo, **Figura 4-9**, de forma que con 53 Mbits/segundo apenas si hay un 0,13% de pérdidas.

```
root@raspberrypi:/home/pi# iperf -u -c 10.0.0.1 -b 53000000 -t 30
-----
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 10.0.0.2 port 49570 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-30.0 sec   190 MBytes   53.0 Mbits/sec
[ 3] Sent 135270 datagrams
[ 3] Server Report:
[ 3] 0.0-30.0 sec   189 MBytes   52.9 Mbits/sec   0.208 ms 223/135269 (0.16%)
[ 3] 0.0-30.0 sec   1 datagrams received out-of-order
```

Figura 4-9. Porcentaje bajo de pérdidas con 53 Mbits

Así con 50 Mbits/segundo quedaría como se muestra en la **Figura 4-10**.

```
root@raspberrypi:/home/pi# iperf -u -c 10.0.0.1 -b 50000000 -t 30
-----
Client connecting to 10.0.0.1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[  3] local 10.0.0.2 port 58657 connected with 10.0.0.1 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-30.0 sec   178 MBytes  49.9 Mbits/sec
[  3] Sent 127326 datagrams
[  3] Server Report:
[  3] 0.0-30.0 sec   178 MBytes  49.9 Mbits/sec   0.158 ms    0/127325 (0%)
[ 31] 0.0-30.0 sec   1 datagrams received out-of-order
```

Figura 4-10. 50 Mbits/s

De esta manera se evita la pérdida de paquetes por la formación de colas. Aun así y como medida preventiva se establece un ancho de banda menor para no estar en el límite, ya que cuando pasamos a UDPcast las pérdidas se ven incrementadas debido a que UDPcast limita la tasa al nivel de aplicación (sin la sobrecarga de los protocolos) mientras que Iperf mide el ancho de banda a nivel de red (incluyendo las cabeceras). También es debido a la diferencia de datos que tratan cada programa, mientras que Iperf transmite unos y ceros de forma aleatoria, UDPcast tiene que mandar la información ordenada y además extraer los datos del disco duro lo que requiere más procesamiento. De este modo, al utilizar la velocidad límite obtenida a partir de Iperf, **Figura 4-11**, se pierden la mayoría de los paquetes.

```
root@raspberrypi:/home/pi# udp-receiver -f 139.mp3 --nosync --interface eth1
Udp-receiver 20120424
UDP receiver for 139.mp3 at 10.0.0.2 on eth1
Connected as #0 to 10.0.0.1
Listening to multicast on 234.0.0.1
bytes= 1 490 944 ( 52.05 Mbps)
Dropped by server now=3 last=0
Missed blocks: 300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,327,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,515,516,564,565,566,567,568
Blocks received: 911/911/1024
Transfer complete.
```

Figura 4-11. UDPcast a 50 Mbits/s

Con velocidades mayores a 35 Mbits/segundo es muy probable que algún paquete sea eliminado. Así que se establece un ancho de banda de 30 Mbits/segundo, **Figura 4-12**, en el transmisor para evitar que se descarten paquetes debidos al desbordamiento de las colas.

```
root@raspberrypi:/home/pi# udp-receiver -f 139.mp3 --nosync --interface eth1
Udp-receiver 20120424
UDP receiver for 139.mp3 at 10.0.0.2 on eth1
Connected as #0 to 10.0.0.1
Listening to multicast on 234.0.0.1
bytes= 6 711 141 ( 17.16 Mbps)
Transfer complete.
```

Figura 4-12. UDPcast a 30 Mbits/segundo

El siguiente paso es probar con la introducción de pérdidas aleatorias con *traffic control*.

4.2 Simulación y corrección de errores

4.2.1 Traffic Control

El diodo de datos es un dispositivo muy robusto frente a interferencias externas. Además como el enlace es punto a punto no se producen colisiones. Aun así, se tendrá en cuenta que en las instalaciones que requieran conexiones con distancias entre puntos mayores de metros, como puede ser un barco donde hay kilómetros de fibra, se pueden producir pérdidas por atenuación (dB/metros).

Para simular esta atenuación se ha utilizado *Traffic Control* [33] (TC) y el módulo de emulación *networking emulation (netem)* de redes de Linux que permite simular diversas condiciones de red. En concreto se pueden configurar los ajustes de transmisión-recepción y la conformación del tráfico de datos: borrado, pérdida, duplicidad o corrupción de los paquetes.

Para emular las pérdidas por atenuación se configura la interfaz de red para introducir una pérdida aleatoria de paquetes el formato de comando de TC responde de la siguiente manera[34]:

1. Para introducir una pérdida aleatoria del 2% de paquetes, **Figura 4-13**, se establece que la interfaz de red *eth1* va a descartar aleatoriamente 2 de cada cien de los paquetes que se envíen a través de la misma. El programa responde a la sintaxis *tc qdisc add/change/show interfaz + acción*.

```
root@raspberrypi0:/home/pi/Desktop# tc qdisc add dev eth1 root netem loss random 2%
```

Figura 4-13. Introducción pérdidas aleatorias

2. Para cambiar el porcentaje, **Figura 4-14**, se cambia *add* por *change*.

```
root@raspberrypi0:/home/pi/Desktop# tc qdisc change dev eth1 root netem loss random 1%
```

Figura 4-14. Modificación

3. Para consultar que porcentaje de pérdidas hay en ese momento en la interfaz indicada, se introduce el comando de consulta, **Figura 4-15**, *tc qdisc show dev eth1*.

```
root@raspberrypi0:/home/pi/Desktop# tc qdisc show dev eth1
qdisc netem 8001: root refcnt 2 limit 1000 loss 2%
```

Figura 4-15. Consulta

4.2.2 Protección frente a errores, FEC

Para blindar el diodo de datos frente a pérdidas de paquetes por desbordamiento de colas, colisiones, interferencias, etc se inserta en *udp-sender* un comando que introduzca la redundancia más adecuada para la protección por perdidas, sin comprometer la velocidad.

```
--fec 16x16/128
```

Figura 4-16. Introducción FEC

Si tenemos $-fec\ Ix\ F/D$, analizando cada parte[32]:

- **I:** Interlineado, indica el número de divisiones de cada bloque. Protege ante pérdidas de paquetes en ráfagas, ya que cuanto mayor sea el bloque está dividido en más partes, por lo que la información perdida es menor.
- **F:** Por cada D bloques incluye F de redundancia. Es directamente proporcional a la redundancia. En este ejemplo, se incluyen 16 bloques redundantes por cada 128.
- **D:** Indica el tamaño de cada trama, es inversamente proporcional a la redundancia.

De esta forma, con la adicción del método de corrección de errores, la aplicación udp-sender quedaría:

udp-sender -f fichero --async --max-bitrate 35m --interface eth1 --fec 12x12/128

Las pruebas realizadas han emulado pérdidas de paquetes del 0.1% hasta el hipotético y casi irreal caso de 20%, y de manera experimental se han determinado las diferentes redundancias necesarias para asegurar la recepción del archivo al completo, por lo que queda de la forma que indica la **Tabla 4-1**.

Pérdidas	FEC	Redundancia
0,10%	8x7/128	5%
0,20%	8x7/128	5%
0,50%	8x7/128	5%
1,00%	8x8/128	6.25%
2,00%	12x12/128	9.40%
5,00%	12x12/128	9.40%
10,00%	16x16/128	12.50%
20,00%	20x30/128	23.40%

Tabla 4-1. Relación Pérdidas-Fec-Redundancia

La tabla relaciona cada porcentaje de pérdidas con la redundancia que hay que introducir para que los datos lleguen al completo y el porcentaje de datos redundantes respecto a la información original.

Así, al final, suponiendo unas pérdidas por defecto del 2%, la redundancia sería ligeramente inferior al 10%. La configuración completada del comando quedaría:

udp-sender -f fichero --interface eth1 --async --nokbd --max-bitrate 35m --fec 12x12/128.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

En esta sección se describen las conclusiones del estudio realizado sobre las ventajas e inconvenientes de los diodos de datos y del propio desarrollo del proyecto.

5.1.1 Conclusiones del estudio

Actualmente, el *firewall* es la barrera tecnológica más utilizada para separar niveles de confidencialidad en una red. Sin embargo los *firewalls* no son impenetrables, son vulnerables a diferentes tipos de ataques y malware y pueden permitir el paso no autorizado, involuntario o intencionado, de información a la red. Incluso con los *firewalls* de mayor seguridad, los diferentes conjuntos de reglas son tan complejos que introducirán retrasos, sobre todo de latencia, en el tráfico de red. Así en redes con requisitos estrictos de latencia, como las redes de control y supervisión de procesos industriales en tiempo real *Industrial Automation and Control System* (IACS) y *Supervisory Control And Data Acquisition* (SCADA), se debe tener en cuenta que el aumento de la latencia debido al *firewall* mermará su capacidad de control.

En estos escenarios, se puede emplear el diodo de datos para desconectar físicamente la transmisión con menos recursos, basándose en una simple combinación de elementos *hardware* que añaden la ventaja de que no utilizan un complejo entramado *software*, únicamente una simple conexión a nivel físico y además están exentos de un costoso mantenimiento, actualizaciones, mal funcionamiento o puertas traseras y la importante característica de una latencia baja, lo que permite un flujo de información a mayor velocidad.

Aun así, también hay que tener en cuenta que la implementación de este tipo de dispositivos supone un reto tecnológico importante y en algunos casos los inconvenientes son superiores a las ventajas, sobre todo porque la amplia mayoría de los protocolos modernos requieren comunicación bidireccional con el fin de funcionar lo más ágilmente posible, ofreciendo integridad, informes de diagnóstico o control del flujo.

A pesar de este inconveniente, una vez configurado correctamente con *software*, se puede obtener ventaja del tráfico unidireccional, con la baja latencia que ofrece este tráfico. Además son

tremendamente efectivos contra exfiltración de información sensible, resistentes a los denominados *Advanced Persistent Threats* (APTs) o Amenaza Persistente Avanzada.

Las **Tablas 5-1 y 5-2**, muestran las fortalezas y debilidades de los *firewalls* y los diodos de datos respectivamente.

Fortalezas Firewall	Debilidades Firewall
Configurable y adaptable	Requiere recursos muy técnicos
Muy utilizado y comprendido	Puede ser sobrepasado/desbordado
Conjunto de reglas poderosos	Introduce latencia, disminuye la velocidad de transmisión
Capaz de enrutar tráfico de todo tipo de redes en dos direcciones	La mayoría de los ataques están diseñados para tráfico bidireccional
Muy barato	Coste a largo plazo, mantenimiento y actualizaciones
Relativamente sencillo encontrar recursos que funcionen con Firewalls	Hackers desarrollan ataques con facilidad

Tabla 5-1. Fortalezas y debilidades de los sistemas *Firewall*

Fortalezas Diodo de Datos	Debilidades Diodo de Datos
El tráfico unidireccional se consigue mediante desconexión física	Pocos protocolos de transmisión unidireccional
Seguro, la mayoría de ataques están diseñados para tráfico bidireccional	Incapaz de enrutar la mayoría del tráfico de red, debido a que no utiliza TCP
No existen casos de haber sido sobrepasados	Uso poco extendido y poco entendimiento
No introduce latencia, transmisiones muy rápidas	No hay forma de saber si el paquete se ha recibido
Bajo coste a largo plazo, no necesita mantenimiento, ni <i>hardware</i> ni <i>software</i>	Los productos industriales son extremadamente caros, debido a su novedad.
Dificultad para los hackers de sobrepasar un diodo de datos industrial	Requiere de conocimiento y herramientas especializadas para instalarlo, además los productores industriales suelen incluir tasas de mantenimiento o configuración que incrementan coste
El tráfico a través del canal óptico no es enrutable	Tráfico vulnerable frente a pérdidas y errores, necesidad de realizar un estudio del enlace e implementar medidas FEC
No hay manera de configurar el <i>software</i> para que sea bidireccional debido a la desconexión física de la interfaz TX	La mayoría de dispositivos comerciales utilizan servidores Linux, que pueden ser vulnerables si no se mantienen al día

Tabla 5-2. Fortalezas y debilidades de los diodos de datos

Al final, como conclusión se puede extraer que no consiste en realizar una comparación sobre qué tipo de tecnología es mejor, la clave es determinar cuál es la que más reduce la amenaza cibernética en cada situación.

5.1.2 Conclusiones del desarrollo

Una vez completado el desarrollo se puede concluir que los objetivos fijados al comienzo del mismo se han alcanzado satisfactoriamente. El diodo de datos desarrollado, sin tener un diseño especialmente complejo garantiza una protección de la información y de los datos de una forma tremendamente efectiva y segura. La simplicidad de sus componentes y funcionamiento no supone una pérdida de confidencialidad ni una merma a la hora de asegurar la integridad de la información. Este dispositivo facilita un funcionamiento completamente unidireccional, la información sensible de la red considerada como segura queda aislada y excluida cualquier posibilidad de filtrado, involuntaria o intencionada. Además descarta cualquier tipo de puerta trasera que pudiera tener un dispositivo comercial o un funcionamiento alejado de las características indicadas.

El dispositivo constituido meramente por sendos proxies, sus conversores y su cableado para constituir el circuito, no necesita de ningún elemento *hardware* añadido, teniendo en cuenta el conversor extra que funciona como *dummy*, para permitir la emisión de información al conversor transmisor.

Respecto a la parte de *software*, con unos conocimientos básicos, se han utilizado herramientas y programas muy sencillos, que operando en conjunto hacen que el diodo de datos funcione correctamente:

- **Sistema operativo Raspbian:** El modo terminal, prácticamente idéntico a Ubuntu y con todas sus funcionalidades, es dinámico e intuitivo.
- **UDPCast:** El programa de transmisión unidireccional cuenta con un gran número de modos de funcionamiento y de posibilidades, como la posibilidad de funcionar sin feedback entre receptor y transmisor (*--async* y *--nosync*), la opción de limitar la tasa (*--max bitrate tasa*) y la inclusión de métodos de corrección de errores (*--fec IxR/T*). Una capacidad que no tiene y que se ha echado en falta es la posibilidad de indicar el nombre del fichero que se transmite al receptor y que ha obligado a realizar la transmisión de información en dos pasos, primero mandar un fichero de texto con el nombre del archivo y una vez que el receptor tiene el nombre y extensión del fichero que va a recibir, proceder a enviar el archivo.
- **Scripts de Bash:** Son herramientas simples, pero tremendamente potentes que brindan la posibilidad de crear programas de cualquier índole. En este proyecto se han creado dos scripts, para el proxy de transmisión y recepción distintamente, con la finalidad de automatizar el envío y recepción de información.

Además este tipo de dispositivos ofrecen una alta relación prestaciones/precio, debido a la relativa economía de los medios, *hardware* libre y *software* de código abierto, que se han utilizado para su desarrollo, si los comparamos con el precio de cualquiera de las soluciones que ofrecen las empresas que se dedican a su comercialización. El valor monetario de todos los elementos adquiridos asciende a 273,35 €. En cualquier caso, el ahorro puede superar ampliamente el 90%-95%, partiendo de que un dispositivo, de los más simples, llega con creces a los varios miles de euros, aunque también hay que tener en cuenta que los diodos de datos comerciales también tienen un sistema más complejo y ofrecen más servicios, aparte de FTP, como puede ser correo, web, etc. En el ANEXO I: PRESUPUESTO, se detalla el precio de cada elemento.

Por último, debido a su reducido tamaño y simplicidad es perfectamente aplicable al entorno de las fuerzas armadas y ofrece una alternativa al concepto *airgap*, que queda obsoleto, con numerosos inconvenientes y puntos débiles. De esta manera su uso se extiende desde situaciones estáticas y

permanentes, como Cuarteles Generales, a cualquier teatro de operaciones con situaciones operativas de gran movilidad, como pueden ser a bordo de buques y en fuerzas desplegadas en tierra, en puestos de Mando de campaña o con el oportuno desarrollo de un dispositivo de tamaño reducido, tener capacidad de montarse sobre vehículos e incluso para infantería ligera.

5.2 Líneas futuras

El diodo de datos desarrollado, a pesar de ser tremendamente funcional, está lejos de ser un producto aplicable a un entorno militar, donde la transmisión es crítica y las condiciones, en el campo de batalla, son extremadamente exigentes, así como la información tratada tiene un valor extraordinario. El dispositivo no está lo suficientemente “rugerizado”, tanto en su parte *hardware* como en la parte *software*.

En futuras investigaciones u otros trabajos posibles, se puede ahondar en diferentes aspectos de este:

- Desarrollar un script más complejo, robusto frente a problemas de desincronización y que incluya el nombre del archivo en la primera transmisión y solucione la problemática de tener que realizar dos envíos distintos primero el nombre el fichero para luego mandar el archivo completo.
- Desarrollar un dispositivo que no requiera del convertidor extra o “*dummy*” para que el emisor funcione adecuadamente.
- Implementar en el diodo de datos distintos puertos y servicios en adición a FTP, como correo, web, chat, etc.
- Mejorar el método de corrección de errores (FEC) o probar con otro medio *software* disponible distinto al ofrecido por UDPcast.
- Desarrollar un dispositivo más rugerizado que permita la realización de pruebas con el diodo de datos en Buques o Unidades de la Armada.
- Estudiar la implementación de un diodo de datos en la separación de dos redes de diferente clasificación de seguridad de la Armada.

6 BIBLIOGRAFÍA

En esta última sección se incluyen todas las referencias de recursos web, libros, artículos, etc. consultados:

- R. U.-T. X.1205, «ITU - Recomendación X.1205 (04/08),» [En línea]. Available:
1] <https://www.itu.int/rec/T-REC-X.1205-200804-I/es>. [Último acceso: 01 02 2016].
- A. Tanenbaum y D. J. Wetherrall, *Computers Networks*, Amsterdam; Washington: Prentice Hall,
2] 2010, pp. 203-204.
- Centro Nacional de Inteligencia, «www.cni.es,» 03 12 2012. [En línea]. Available:
3] http://www.cni.es/comun/recursos/descargas/NS-04_Seguridad_de_la_Informacion.pdf. [Último acceso: 07 02 2016].
- INTELPAGE, «www.intelpage.info,» [En línea]. Available: <http://www.intelpage.info/centro-de-operaciones-de-seguridad-de-la-informacion-del-ministerio-de-defensa.html>. [Último acceso: 2016 02 15].
4]
- M. Abalo Cores, «Las comunicaciones y sistemas de información en la Armada. Hacia dónde vamos.,» *Revista General de Marina*, nº 265, p. 8, 2013.
5]
- J. Garat Caramé, «www.dintel.org,» 11 10 2011. [En línea]. Available:
6] <http://www.dintel.org/Documentos/2011/TESIC/ponencias/jgarat.pdf>. [Último acceso: 15 02 2016].
- Ministerio de Defensa de España, «[emad.mde.es](http://www.emad.mde.es),» 18 09 2015. [En línea]. Available:
7] <http://www.emad.mde.es/EMAD/novemad/noticias/2015/09/Listado/150918-reunion-fmn-emacon.html>. [Último acceso: 30 03 2016].
- NATO-OTAN, «ANEXO N VOL II,» de *NATO FMN IMPLEMENTATION PLAN v3.1*, 2014, p. 317.
8]
- NATO, MARITIME AND MOBILE TACTICAL WIDE AREA NETWORKING - (ACP 200(D) Volume 2), 2015.
9]
- wikipedia, «wikipedia.org,» [En línea]. Available:

- 10] https://en.wikipedia.org/wiki/Air_gap_%28networking%29. [Último acceso: 24 03 2016].
- i.BBC, «bbc.com,» 11 10 2015. [En línea]. Available:
- 11] http://www.bbc.com/mundo/noticias/2015/10/151007_iwonder_finde_tecnologia_virus_stuxnet. [Último acceso: 21 03 2016].
- A. Scott, «www.sans.org,» [En línea]. Available: [https://www.sans.org/reading-](https://www.sans.org/reading-room/whitepapers/firewalls/tactical-data-diodes-industrial-automation-control-systems-36057)
- 12] [room/whitepapers/firewalls/tactical-data-diodes-industrial-automation-control-systems-36057](https://www.sans.org/reading-room/whitepapers/firewalls/tactical-data-diodes-industrial-automation-control-systems-36057). [Último acceso: 19 03 2016].
- deep-secure, «deep-secure.com,» [En línea]. Available: [http://www.deep-secure.com/how-data-](http://www.deep-secure.com/how-data-diodes-work/)
- 13] [diodes-work/](http://www.deep-secure.com/how-data-diodes-work/). [Último acceso: 2016 02 05].
- C. A. Nilsen y C. Valley', «Method for transferring data from an unsecured computer to a secured
- 14] computer». Estados Unidos Patente US5703562, 20 11 1996.
- 15] M. H. Kang, I. S. Moskowitz y S. Chincheck, «The Pump: A Decade of Covert Fun».
- DualDiode®, Owl, «<http://owlcti.com/>,» [En línea]. Available: <http://owlcti.com/>. [Último acceso:
- 16] 15 01 2016].
- owlinc, «Miniaturized Cross Domain Solution (MCDS),» [En línea]. Available:
- 17] www.owlcti.com/pdfs/datasheets/government/mcdfs.pdf. [Último acceso: 15 01 2016].
- L. FOX IT, «foxit.com,» LOGITEK, [En línea]. Available:
- 18] http://www.ciberseguridadlogitek.com/microsite/documentos/ciberseguridadlogitek_DataDiode_FoxIT_wp.pdf. [Último acceso: 16 01 2016].
- Secunet, «www.secunet.com,» 03 2014. [En línea]. Available: www.secunet.com/en/sina. [Último
- 19] acceso: 2016 01 15].
- NATO IACD, «www.ia.nato.int,» [En línea]. Available:
- 20] https://www.ia.nato.int/niapc/Product/Speed-Data-Diode--HSDD-_249. [Último acceso: 2016 02 15].
- M. Ottela, *Tinfoil Chat- Instant messaging with endpoint security*, University of Helsinki, Finland:
- 21] TFG, 2014.
- A. Scott, «blog.cimention.com,» 03 02 2013. [En línea]. Available:
- 22] <http://blog.cimention.com/blog/building-your-own-data-diode-with-open-source-solutions>. [Último acceso: 2016 02 15].
- RaspberryPi, «raspberrypi.org,» RaspberryPi, [En línea]. Available:
- 23] <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>. [Último acceso: 2016 02 04].
- TP-LINK, «www.tp-link.com,» [En línea]. Available: [http://www.tp-](http://www.tp-link.com/ar/products/details/cat-4792_MC100CM.html)
- 24] [link.com/ar/products/details/cat-4792_MC100CM.html](http://www.tp-link.com/ar/products/details/cat-4792_MC100CM.html). [Último acceso: 15 01 2016].
- Udpcast, Linux, «udpcast.linux.lu,» [En línea]. Available: <https://www.udpcast.linux.lu/>. [Último
- 25] acceso: 2016 02 04].
- A. U. Knaff, «linus.die.net,» udp-cast, [En línea]. Available: <http://linus.die.net/man/1/udp-sender>.
- 26] [Último acceso: 18 02 2016].

- wordpress.com, «enredandoconredes.com,» [En línea]. Available:
27] <http://enredandoconredes.com/2012/05/07/autonegociacion/>. [Último acceso: 2016 02 08].
- Startech, «startech.com,» [En línea]. Available:
28] http://sgcdn.startech.com/005329/media/sets/ET91000xx2_Manual/ET91000xx2-MX.pdf. [Último acceso: 08 02 2016].
- D. Alonso, «Debian Network Configuration,» NetworkConfiguration, [En línea]. Available:
29] <https://wiki.debian.org/es/NetworkConfiguration>. [Último acceso: 2016 01 04].
- iperf, «iperf.fr,» 06 06 2010. [En línea]. Available: <https://iperf.fr/>. [Último acceso: 17 02 2016].
30]
- A. E. Rodríguez, «<https://geekytheory.com>,» [En línea]. Available:
31] <https://geekytheory.com/tutorial-raspberry-pi-9-servidor-ftp/>. [Último acceso: 2016 03 06].
- A. Knaff, «linux.die.net,» Linux, [En línea]. Available: <http://linux.die.net/man/1/udp-sender>.
32] [Último acceso: 07 02 2016].
- A. N. Kuznetsov, «manpages.ubuntu.com,» Ubuntu, [En línea]. Available:
33] <http://manpages.ubuntu.com/manpages/intrepid/man8/tc.8.html>. [Último acceso: 18 02 2016].
- M. A. Brown, «Traffic Control HOWTO,» Linux, [En línea]. Available:
34] http://www.tldp.org/HOWTO/html_single/Traffic-Control-HOWTO/. [Último acceso: 22 02 2016].
- 35] A. S. Tanenbaum, Computer Networks, Amsterdam.

ANEXO I: PRESUPUESTO.

En la **Tabla AI-1** de este anexo se detalla el coste particular de cada elemento *hardware* adquirido para la realización del diodo de datos.

	Producto	Coste Unitario	Coste Total
1.	2xRaspberryPi +B	31,90 €	63,80 €
2.	3xConvertidores Ethernet-fibra óptica	33,65 €	100,95 €
4.	2xCargadores Mini-USB	2,50 €	5,00 €
5.	2x Cable Ethernet	2,50 €	5,00 €
6.	2xCable Fibra Óptica	24,10 €	48,20 €
7.	1xCable Ethernet cruzado	2,50 €	2,50 €
8.	2xUSB-Ethernet	18,95 €	37,90 €
9.	2xCarcasa <i>RaspberryPi</i>	5,00 €	10,00 €
		TOTAL	273,35 €

Tabla AI-1. Presupuesto

ANEXO II: SCRIPT TRANSMISOR.

El script realizado para el control de la información en el proxy transmisor responde a la siguiente sintaxis mostrada en la **Figura AII-1**.

```
GNU nano 2.2.6                               File: rp0.sh                               Modified

#!/bin/bash
### BEGIN INIT INFO- Permite la inserción en scripts de arranque
# Provides:          tfg
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: blabla
# Description:
#
### END INIT INFO
while ;;
do
ls -l diododatos/* | awk '{print $5" "$6" "$7" "$8" "$9}' > ficheros.txt
cat ficheros.txt transmitidos.txt | sort | uniq -u | cut -d"/" -f2 > portransmitir.txt

for archivo in `cat portransmitir.txt`; do

echo $archivo > nombre_archivo.txt

udp-sender -f nombre_archivo.txt --async --interface lo --max-bitrate 35m

sleep 1

udp-sender diododatos/$archivo --async --interface lo --max-bitrate 20m

ls -l diododatos/$archivo | awk '{print $5" "$6" "$7" "$8" "$9}' >> transmitidos.txt

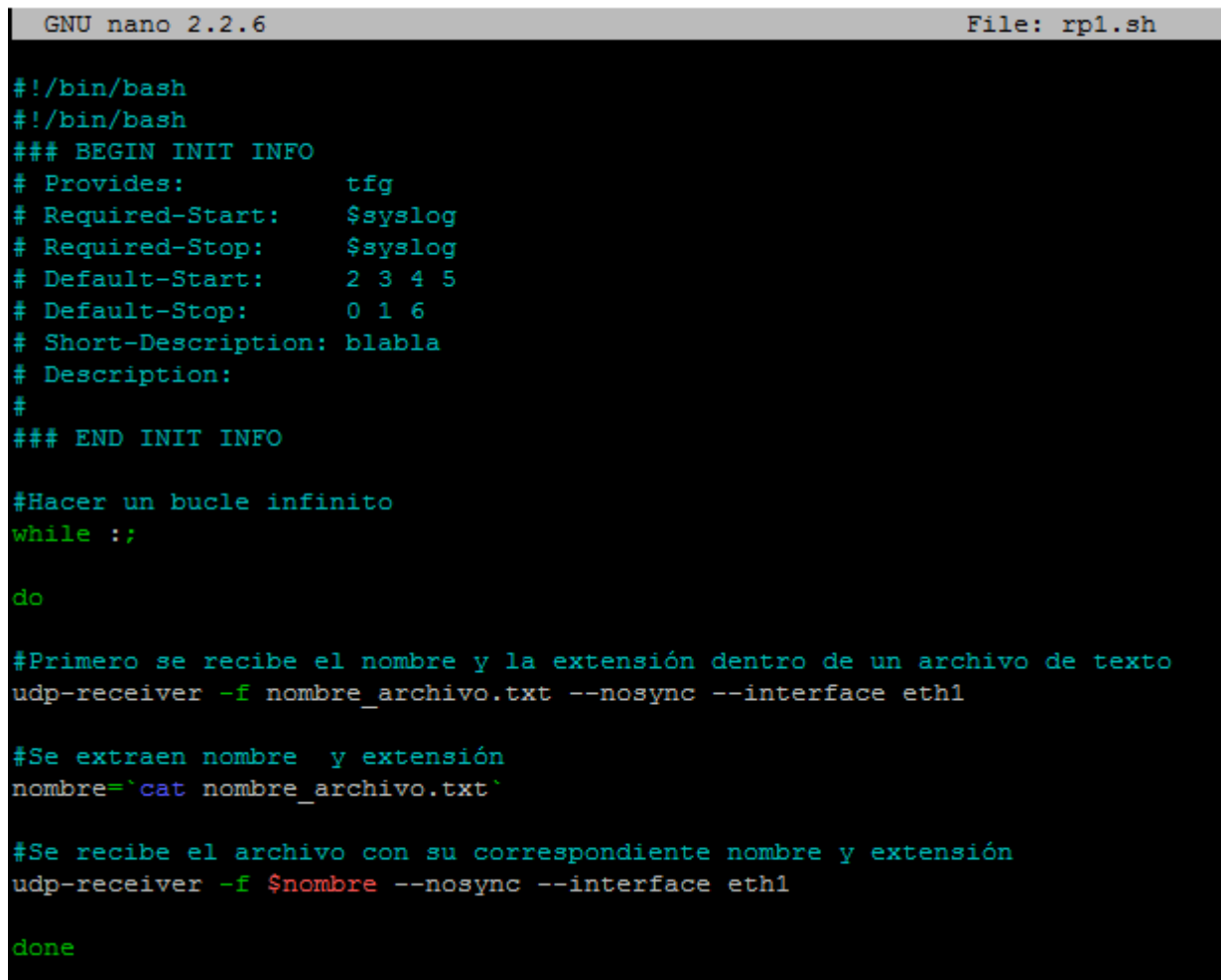
done
sleep 10

done
```

Figura AII-1. Script Transmisor

ANEXO III: SCRIPT RECEPTOR.

El script realizado para el control de la información en el proxy receptor responde a la sintaxis mostrada en la **Figura AIII-1**.



```
GNU nano 2.2.6                                     File: rp1.sh

#!/bin/bash
#!/bin/bash
### BEGIN INIT INFO
# Provides:          tfg
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: blabla
# Description:
#
### END INIT INFO

#Hacer un bucle infinito
while ;;

do

#Primero se recibe el nombre y la extensión dentro de un archivo de texto
udp-receiver -f nombre_archivo.txt --nosync --interface eth1

#Se extraen nombre y extensión
nombre=`cat nombre_archivo.txt`

#Se recibe el archivo con su correspondiente nombre y extensión
udp-receiver -f $nombre --nosync --interface eth1

done
```

Figura AIII-1. Script Receptor

ANEXO IV: ACRÓNIMOS Y GLOSARIO.

En este anexo se muestran los acrónimos nombrados en toda la memoria, con su significado y, cuando corresponde, su traducción.

ACK	Acknowledgement (Acuse de recibo/asentimiento)
APT	Advanced Persistent Threats (Amenazas Avanzadas Persistentes)
ARP	Address Resolution Protocol (Protocolo de resolución de direcciones)
CEI	Communications Electronic Instructions (NATO) (Instrucciones de Comunicaciones)
CDS	Cross Domain Solution (Solución para la conexión de dominios)
CIS	Communications and Information Systems Services (Sistemas de comunicación e información)
DHCP	Dynamic Host Configuration Protocol (Protocolo de configuración dinámica de host)
DLP	Data Loss Prevention (Prevención de pérdida de datos)
FEC	Forward Error Correction (Métodos de corrección de datos)
FTP	File transfer protocol (Protocolo de transferencia de archivos)
LED	Light Emitting Diode (Diodo Emisor de luz)
IACS	Industrial Automation and Control System (Automatización Industrial y Control de Sistemas)
IEC	International Electrotechnical Commission (Comisión Electrotécnica Internacional)
IP	Internet Protocol (Protocolo de Internet)
ISA	International Society of Automation (Sociedad Internacional de Automatización)
MAC	Media Access Control (Control de acceso de medios)
NC	NATO CONFIDENTIAL
NIST	National Institute of Standards and Technology (Instituto Nacional de Normas y Tecnología) (EEUU)
NR	NATO RESTRICTED
NS	NATO SECRET
RX	Receptor
SCADA	Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos)
SMN	Sistema de Mando Naval
TC	Traffic Control (Control del Tráfico)
TCP	Transmission Control Protocol (Protocolo de Control de la Transmisión)
TX	Transmisor

UDP	User Datagram Protocol (Protocolo de Datagrama de Usuario)
UIT	Unión Internacional de Comunicaciones
Vsftpd	Very Secure FTP Daemon (Servidor FTP muy seguro, ejecutado como demonio)