Escuela Politécnica Superior de Linares

Grado en Ingeniería Telemática

UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

IMPLEMENTACIÓN DEL PROTOCOLO MULTICAMINO TCP EN SISTEMAS LINUX

Alumno: José Antonio Parrilla Martínez

Tutor: Juan Carlos Cuevas Martínez

Depto.: Ingeniería de Telecomunicación

Tutor: Antonio Jesús Yuste Delgado

Depto.: Ingeniería de Telecomunicación

Junio, 2019

ÍNDICE

RESUMEN

El estado actual de la tecnología, que demanda unas conexiones cada vez más exigentes en cuanto a rendimiento y retraso, propone un replanteamiento del esquema de conexión que se ha ido siguiendo a lo largo de estos años. Frente al uso de una sola interfaz para la conexión, la tecnología multicamino TCP (MPTCP) propone utilizar varias interfaces para establecer diferentes caminos o rutas, aportando un mayor rendimiento y una mayor robustez.

En el presente trabajo, se ha realizado una instalación del protocolo MPTCP empleando una implementación disponible para Ubuntu. Gracias a ella, se ha realizado un estudio de las posibilidades que tiene el protocolo en diversos escenarios.

Mediante este estudio se han podido observar las ventajas del protocolo frente a caídas en la red, el aumento del rendimiento respecto a TCP y la disminución del tiempo de retraso de los paquetes enviados también en comparación con TCP.

ABSTRACT

The current state of technology, which demands ever more demanding connections in terms of performance and delay, proposes a rethinking of the connection scheme that has been followed throughout these years. Faced with the use of a single interface for the connection, the multipath TCP (MPTCP) technology proposes using several interfaces to establish different paths or routes, providing greater performance and greater robustness.

In the present work, the MPTCP protocol has been installed employing an available implementation for Ubuntu. Thanks to it, the protocol has been tested in different scenarios to analyse its behaviour and main features.

Through this study we have been able to observe the advantages of the protocol against network drops, the increase in performance with respect to TCP and the decrease in the delay time of the packets sent also compared to TCP.

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN

El primer esfuerzo por desarrollar y abrir una arquitectura de red, dio lugar a la arquitectura TCP/IP, usada desde entonces hasta hoy en día debido a sus capacidades y versatilidad. Dentro de esta arquitectura se encuentra el protocolo de control de transmisión, más conocido como TCP (Transmission Control Protocol) [35] en la capa de transporte, el cual es usado de forma masiva en la actualidad. Este protocolo está orientado a conexión, es decir, establece un camino virtual entre los extremos de la comunicación. Se diseñó para establecer un flujo de datos confiable, extremo a extremo, que intenta superar los problemas que ocurran en la red. Además, implementa mecanismos de control de flujo, control de error y control de congestión.

Sin embargo, hoy en día las necesidades de servicios de intercambio de información en Internet son cada vez mayores y de gran diversidad, por lo que surgen nuevas herramientas. En la actualidad el auge de los contenidos 4k, e incluso 8k en algunos países, con varias plataformas de servicios de vídeo bajo demanda y *streaming* existentes*,* hacen que se requiera un aumento del rendimiento de las conexiones. Además, existen otros escenarios en los que se necesita un gran ancho de banda como por ejemplo en la descarga de programas de gran peso, o las actualizaciones de software de los sistemas operativos, de forma que sean más rápidas y transparentes para el usuario. Por otro lado, los equipos finales han evolucionado, pero no aprovechan al máximo sus capacidades de conexión. Por ejemplo, los dispositivos móviles disponen de varias interfaces inalámbricas y solamente usan una al mismo tiempo (la mayoría de las veces), de hecho, cuando el usuario cambia de una red a otra se le asigna una nueva dirección IP que provoca el restablecimiento de las conexiones TCP. Por ello surgió la opción *Multipath Operation Transport Control Protocol* (MPTCP). En enero del año 2013, el *Internet Engineering Task Force* (IETF) publicó la especificación como experimental en la *Request for Comments* (RFC) 6824 [17], cuya última actualización corresponde al 20 de diciembre de 2018.

MPTCP es un conjunto de extensiones de TCP para mejorar el uso de recursos dentro de la red y, por lo tanto, mejorar la experiencia de usuario a través de un mayor rendimiento y una mayor resistencia a fallos en la red. Esta opción proporciona la capacidad de usar simultáneamente múltiples rutas en una comunicación. De esta forma emplea varias interfaces físicas a la hora de realizar una conexión creando sub-flujos de datos. Sin embargo, el tipo de servicio que ofrece MPTCP a las aplicaciones es el mismo que el ofrecido por TCP, es decir, envío de flujo de datos confiable.

Así, en el presente Trabajo Fin de Grado (TFG) se ha realizado un estudio de este protocolo para poder evaluar su rendimiento mediante una serie de escenarios que permitan conocer

sus posibilidades y ventajas frente al protocolo de transporte más usado en la actualidad: TCP. Por otro lado, también se han descrito una serie de problemas, tanto a la hora de la preparación de las pruebas (instalación en los equipos) como los problemas intrínsecos del protocolo MPTCP. Con la lectura de este trabajo, se pretende que se ponga de manifiesto el comportamiento de este protocolo ante diferentes escenarios reales y de esta forma pueda comprenderse su relevancia en las comunicaciones actuales si se implementa de forma general, como lo está su predecesor TCP.

1.1 Objetivos

Los objetivos del presente proyecto teórico/experimental son:

- Estudio en profundidad del protocolo multi-camino TCP.
- Implementación del protocolo en dispositivos con diferentes interfaces de red para sistemas operativos Linux.
- Realización de pruebas con distintos escenarios de simulación.

Estos objetivos se enmarcan dentro de las competencias que se desarrollan en el proyecto y que son expuestas a continuación:

- TEL2 – Capacidad para aplicar las técnicas en que se basan las redes, servicios y aplicaciones telemáticas, tales como sistemas de gestión, señalización y conmutación, encaminamiento y enrutamiento, seguridad (protocolos criptográficos, tunelado, cortafuegos, mecanismos de cobro, de autenticación y de protección de contenidos), ingeniería de tráfico (teoría de grafos, teoría de colas y teletráfico) tarificación y fiabilidad y calidad de servicio, tanto en entornos fijos, móviles, personales, locales o a gran distancia, con diferentes anchos de banda, incluyendo telefonía y datos.
- TEL5 – Capacidad de seguir el progreso tecnológico de transmisión, conmutación y proceso para mejorar las redes y servicios telemáticos.
- TEL7 – Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

1.2 Estado del arte

El número de opciones y usos de MPTCP es bastante amplio, a pesar de que aún se encuentra en una fase de RFC – Experimental [17]. Actualmente hay diversos documentos en los que se estudia el uso de esta extensión de TCP, analizando el comportamiento del protocolo en redes de sensores [34], o en diferentes plataformas Linux con el uso de dos interfaces de red [4]. En ellos se suelen utilizar, principalmente, las métricas de rendimiento [7] (en inglés *throughput),* que es la tasa de transferencia efectiva o información neta que

transcurre a través de un sistema, y el retardo [39] (en inglés *delay),* que es la demora que se produce en una comunicación a distancia. Adicionalmente se hace uso de otras métricas como la fluctuación de fase [13], (en inglés *Jitter),* que es la desviación de la periodicidad real de una señal, a menudo en relación con una señal de reloj de referencia o la tasa de bits, (en inglés *bit rate),* que serían el número de bits que se transmiten o procesan por unidad de tiempo. Por último, el tipo de estudios que se suelen realizar comprenden diversos escenarios, como el uso de MPTCP para vídeo, audio, video-llamadas, visitas a páginas webs, o descargas de archivos.

Sin embargo, en general, lo que suele unir todos los estudios es su comparativa con TCP, ya sea con unos escenarios u otros, para poder comprobar si realmente aporta valor la inclusión de MPTCP al protocolo de transporte por excelencia a lo largo de todos estos últimos años. Otro aspecto a tratar en este documento, tras realizar las pruebas pertinentes y comprobar si aporta las mejoras requeridas, será el hecho de su posible implementación de forma general en Internet y cómo podría ser su adopción.

1.3 Evolución y versiones

Esta ampliación del protocolo TCP, está aún en desarrollo y del mismo se encarga el grupo de trabajo Multipath TCP. El primer documento disponible acerca de MPTCP reconocido en el IETF data del 8 de mayo de 2009. En este borrador (Internet-Draft) se hablaba del problema de que los usuarios finales de una comunicación están conectados por varias rutas, pero la naturaleza de TCP/IP restringe las comunicaciones a una única ruta por socket [18]. Desde entonces se publicaron diferentes borradores hasta llegar a la publicación de la RFC el 18 de enero de 2013. Posteriormente, tras la publicación, la *Internet Assigned Numbers Authority* (IANA) actualizó sus registros para incluir la RFC 6824 publicada y la última actualización de esta RFC se realizó el 20 de diciembre de 2018 en la que se cambió el resumen (*abstract)* del documento. A continuación, se puede ver la evolución con las distintas publicaciones hasta llegar a la RFC final:

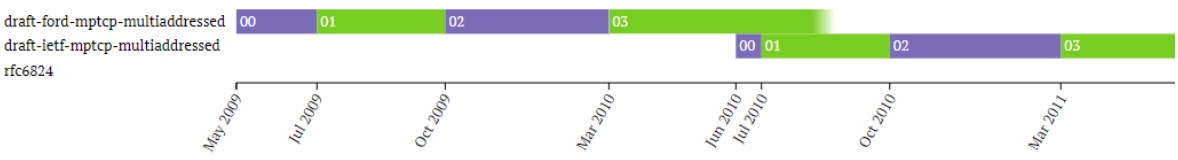

Figura 1: Estado de las publicaciones de la Extensión de TCP (MPTCP) para comunicación por varias rutas con múltiples interfaces (1). Fuente: IETF

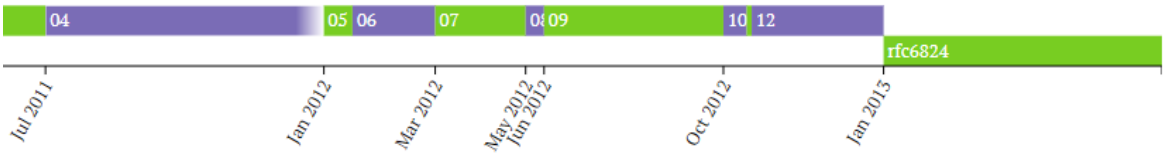

Figura 2 : Estado de las publicaciones de la Extensión de TCP (MPTCP) para comunicación por varias rutas con múltiples interfaces (2). Fuente: IETF

La última actualización de MPTCP, a fecha de finalización de este documento, corresponde al draft-ietf-mptcp-rfc6824bis [2] publicada el 30 de mayo de 2019, por lo que se puede comprobar que se está trabajando en el protocolo de forma actual.

Por otro lado, hay que señalar que para la realización del proyecto se ha utilizado una modificación del kernel de Linux de la distribución Ubuntu 18.04.2 con soporte de larga duración, para poder instalar una implementación de MPTCP. Esta implementación la facilita la Universidad Católica de Louvain en una página web en la que se pueden ver las diferentes versiones y formas de implementación [14]. En nuestro caso se ha optado por la instalación mediante un repositorio con la herramienta para la instalación de paquetes avanzada (en inglés *Advanced Packaging Tool) apt*, en el cual se encuentra la última versión disponible a la entrega de este documento, v0.94, publicada en marzo del año 2018. Esta versión está basada en el kernel de Linux con soporte a largo plazo v4.14.x. Existen un total de 9 versiones desde la v0.86 hasta la actual con las siguientes evoluciones destacables:

- Versión 0.87.x o menor: permite habilitar o deshabilitar MPTCP y la suma de comprobación (en inglés *checksum)* de los paquetes. Por otro lado, es posible modificar la frecuencia de envío de la opción *MP_CAPABLE* (que permite anunciar la posibilidad de utilizar MPTCP) descrita en la sección 2.2.1 y la creación varios sub-flujos en la misma interfaz.
- Versión 0.88.x: se añade la creación de administradores de rutas, de forma que el equipo pueda crear nuevos sub-flujos y no solamente aceptar de forma pasiva.
- Versión 0.89.x: incorpora la configuración de mecanismos de gestión y dos planificadores, para enviar los datos por los sub-flujos con menor retardo o con el método *roundrobin.* Además, se incorpora una opción para habilitar MPTCP solamente si la aplicación que se usa, ha establecido la opción de activar MPTCP en el *socket* (en *SOL_TCP)*.
- Versión 0.91.x: Se añade un planificador más para enviar los datos de forma redundante, útil para una baja latencia sacrificando ancho de banda.

- Versión 0.92.x o superior: se permite volver a crear sub-flujos después de un tiempo determinado y se añade una opción para tener información detallada de cada sub-flujo.

2 DESARROLLO

En esta sección se va a describir el funcionamiento del protocolo y sus características más importantes. Por otro lado, se explicará las áreas de aplicación más interesantes de MPTCP y los problemas que puede conllevar su uso.

2.1 Introducción MPTCP y motivación

Como se ha visto en la primera sección de este documento, este protocolo surgió por las necesidades de comunicaciones que hay en la actualidad, y es por ello que está comenzando a implantarse de forma experimental y gradual. No obstante, la empresa Apple publicó en septiembre de 2013 la inclusión en iOS 7 del protocolo y desde Android 4.4.2, lanzada el 31 de octubre de 2013, fue posible instalar MPTCP, aunque solamente teniendo acceso de super-usuario[1] (*root)*. En la actualidad el protocolo está disponible en iOS, Android (dependiendo del kernel solamente para algunas funciones y apps determinadas), para servicios de *Amazon Web Services* y Google, la mayoría de distribuciones de Linux, *OpenWRT* o Raspberry PI.

MPTCP es un conjunto de extensiones de TCP, aunque también considerado un protocolo, que proporciona una serie de herramientas para ampliar el protocolo TCP. Su objetivo es conseguir que una conexión TCP estándar pueda establecer diferentes caminos[2], o sub-flujos de datos, simultáneamente. En la siguiente figura podemos ver cuál sería su posición dentro de la pila de protocolos:

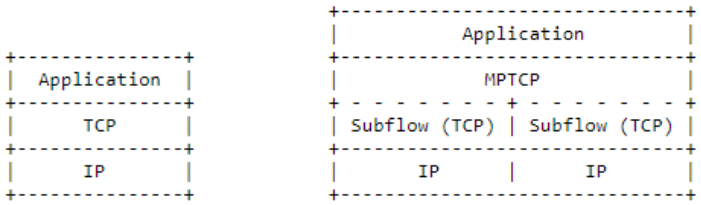

Figura 3: Comparativa entre la pila de protocolo estándar y la pila de protocolos con MPTCP. Fuente: RFC 6824

MPTCP opera en la capa de transporte, como se puede ver en la figura 3, y es transparente para las capas superiores e inferiores. Su diseño permite ser utilizado por las aplicaciones

[1] El acceso de super-usuario permite tener acceso al directorio raíz del sistema, por lo que se controla al completo. Gracias a ello, se puede instalar modificaciones del sistema operativo y optimizar diversas áreas del dispositivo. Sin embargo, es desaconsejable si no se tienen los conocimientos necesarios, ya que se puede bloquear el dispositivo, no se pueden instalar actualizaciones oficiales automáticamente y se puede perder la garantía. Los dispositivos vendidos al público no tienen este acceso.

[2] Un camino es una ruta en la red entre una dirección IP de origen y otra dirección IP de destino, con sus correspondientes puertos respectivos. Mediante MPTCP, los paquetes pueden llevar rutas diferentes en la comunicación entre ambas partes.

sin necesidad de cambios. De esta forma, una aplicación comienza abriendo un *socket* de manera normal y la señalización y operación de MPTCP las maneja la implementación en el sistema de MPTCP.

La motivación que tiene este protocolo es intentar aprovechar la capacidad que tienen los recursos de una red, ya que TCP limita la conexión entre dos extremos a una única a nivel de transporte. Estas conexiones se realizan extremo a extremo, por lo que cada uno de estos extremos de la comunicación tiene un par dirección IP y puerto, de forma que solamente se tiene una única interfaz de red y un único flujo de datos en la red. Por ello, si tenemos en un equipo de comunicación varias interfaces de red, conectadas a diferentes redes de acceso a Internet, estaríamos desaprovechando los recursos, por lo que surge la idea de la utilización de esas interfaces al mismo tiempo mediante el protocolo MPTCP. Si estas interfaces se encuentran en la misma red, el acceso a Internet sería el mismo, pero se añadiría una interfaz como respaldo, por lo que si fallase una de las dos se podría seguir comunicando con la otra sin esperas. Sin embargo, si se tiene un acceso diferente en cada interfaz, por ejemplo, en una empresa con una red de un operador y otra red de otro, sí se podría mejorar el rendimiento sumando estas dos redes con el uso de MPTCP.

Si una aplicación no es compatible con MPTCP, lo cual ocurre cuando uno de los dos extremos no lo soporta, el funcionamiento a nivel de transporte será de la misma forma que con TCP normal. Si hay rutas adicionales disponibles entre los dos equipos que se comunican y ambos soportan el protocolo, se crean sesiones TCP adicionales, que se denominan sub-flujos, en esas rutas y se combinan con la sesión existente que continúa apareciendo como una conexión única a las aplicaciones de ambos extremos.

Tras la realización de las pruebas necesarias para el estudio del protocolo, se comprobará la mejora que supone MPTCP frente a TCP en cuanto al retraso sufrido por los paquetes y el incremento del ancho de banda con el uso de varias redes.

2.2 Funcionamiento

Las características de este protocolo permiten un uso óptimo de los recursos que hay en la red y en los equipos de comunicaciones. Su funcionamiento consiste en el uso de varias interfaces de red para establecer la comunicación entre los dos extremos de la comunicación por varios caminos simultáneamente, llamados sub-flujos como se ha visto anteriormente. A continuación, se va a proceder a describir cómo se realiza el proceso de conexión desde el inicio de la misma hasta su finalización.

2.2.1 Inicio de una conexión MPTCP

El inicio de la conexión se produce con la misma señalización que para iniciar una conexión TCP normal, pero los paquetes *SYN*, *SYN/ACK* y *ACK* disponen de la opción *MP_CAPABLE,* la cual podemos ver en la figura 4.

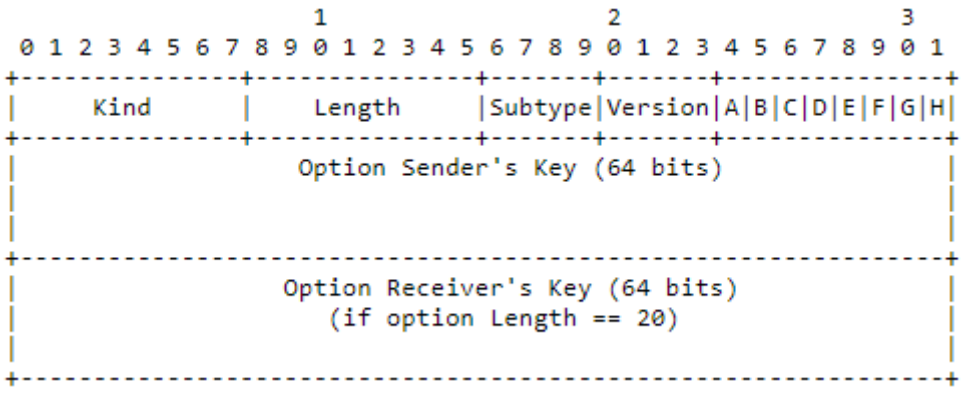

Figura 4: Opción Multipath_Capable (MP_CAPABLE). Fuente: RFC 6824

Esta opción es de longitud variable y sirve para verificar si el equipo remoto soporta MTPCP, además de permitir a los equipos intercambiar información para autenticar el establecimiento de sub-flujos adicionales futuros, mediante una clave[3] de 64 bit que genera el remitente. El intercambio que se realiza junto a las claves usadas puede observarse en la figura 5.

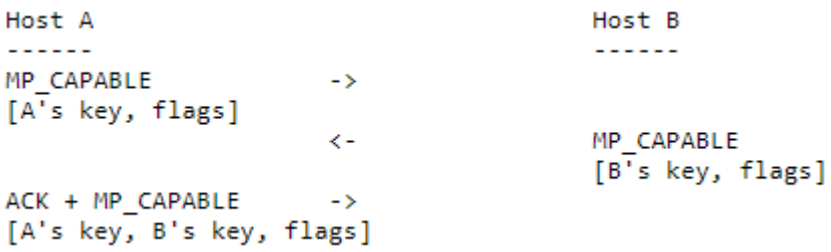

Figura 5: Inicio de la conexión. Fuente: RFC 6824

2.2.2 Asociación de un nuevo sub-flujo a una conexión MPTCP existente

El establecimiento de claves mediante la opción *MP_CAPABLE,* proporciona los datos necesarios que se pueden usar para autenticar los equipos finales cuando se configuren nuevos sub-flujos. Estos nuevos sub-flujos comienzan de la misma manera de la que se

[3]Esta clave la genera el remitente de forma aleatoria y se envía como texto plano sólo en este primer inicio de conexión. Todos los futuros sub-flujos identificarán la conexión mediante un hash criptográfico de 32 bits de esa clave.

inicia una conexión TCP normal, pero los paquetes *SYN, SYN/ACK* y *ACK* llevan también la opción *MP_JOIN,* la cual se muestra a continuación.

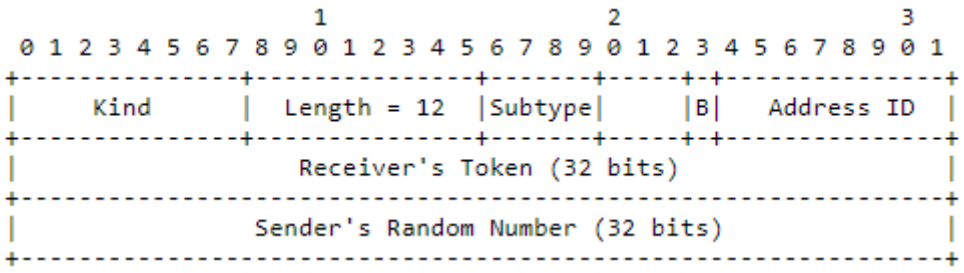

Figura 6: Opción de conexión MP_JOIN para SYN inicial. Fuente: RFC 6824

El equipo A inicia un nuevo sub-flujo entre una de sus direcciones y otra de las del equipo B. Mediante la clave de la sección anterior, se genera una cadena de caracteres (*Token*) usada para identificar que conexión MPTCP se está uniendo y el código de autentificación de mensajes en clave-hash llamado *Hash-based Message Authentication* (*HMAC)* [17] (Sección 3.2 del documento referenciado) es usado para la autenticación. Este código de autenticación utiliza las claves intercambiadas mediante la opción *MP_CAPABLE*, anteriormente descrita, y unos números aleatorios (*nonce)* intercambiados en la opción MP_JOIN. Esta opción, también contiene banderas y un identificador de dirección que puede ser usado para referirse a la dirección de origen sin que el remitente necesite saber si ha sido cambiado por un dispositivo de traducción de direcciones de red (NAT). Este intercambio se muestra en la figura 7.

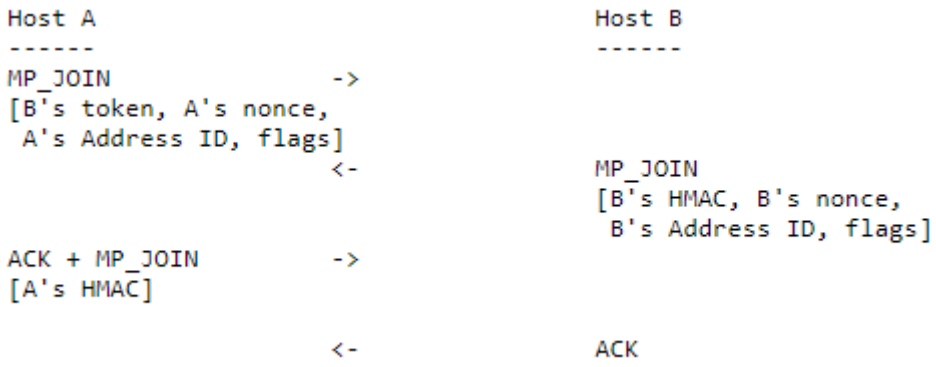

Figura 7: Asociación de un nuevo sub-flujo a una conexión existente. Fuente: RFC 6824

2.2.3 Información sobre otra dirección potencial disponible al otro extremo

El conjunto de direcciones o interfaces que tiene disponible un equipo puede cambiar durante una conexión MPTCP. Por ello, el protocolo soporta añadir y eliminar direcciones de forma implícita y explícita. Si el equipo A ha establecido un sub-flujo con una dirección IP#-A1 y desea abrir un segundo sub-flujo con la dirección IP#-A2, simplemente inicia el

establecimiento del sub-flujo como se ha explicado en la sección anterior. El equipo remoto será informado implícitamente acerca de la nueva dirección IP.

Por otro lado, en algunas circunstancias, un equipo desea informar al otro de que dispone de una nueva dirección IP sin establecer un nuevo sub-flujo, por ejemplo, cuando un NAT impide la configuración en una dirección. En el ejemplo expuesto en la figura 8, el equipo A informa al B de su dirección IP alternativa (IP#-A2). El equipo B puede enviar más tarde un *MP_JOIN* a esta nueva dirección. Debido a la presencia de equipos intermedios, como NAT que pueden traducir direcciones IP, esta opción usa un identificador de dirección para identificar una dirección con un equipo.

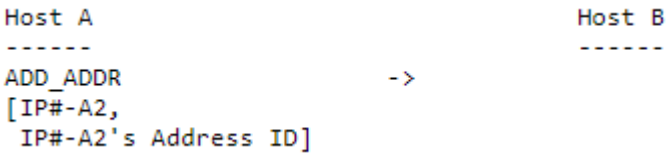

Figura 8: Informar al otro equipo de una nueva dirección IP disponible. Fuente: RFC 6824

La figura 9 muestra el mensaje para eliminar una dirección, haciendo uso del identificador de dirección que se ha confirmado al añadir una dirección en el establecimiento de la comunicación.

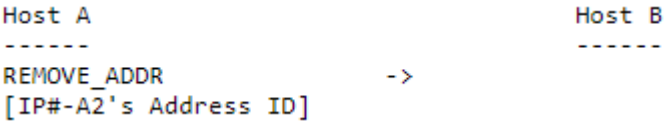

Figura 9: Eliminación de una dirección IP. Fuente: RFC 6824

2.2.4 Transferencia de datos usando MPTCP

Para garantizar una entrega confiable y en orden de los datos que van sobre los sub-flujos que pueden aparecer y desaparecer en cualquier momento, el protocolo utiliza un número de secuencia de datos de 64 bits llamado *Data Sequence Number (DSN),* para numerar todos los datos enviados sobre la conexión MPTCP. Cada sub-flujo tiene su propio número de secuencia de 32-bit y una opción de MPTCP asigna la secuencia del sub-flujo con los datos del sub-flujo. De esta manera los datos pueden retransmitirse en diferentes sub-flujos (asignados al mismo *DSN)* en caso de fallo.

El mensaje *Data Sequence Signal*, *DSS,* lleva el mapeo de la secuencia de datos. Este mapeo incluye el número de secuencia de sub-flujo, el número de secuencia de datos y la longitud para la cual el mapeo es válido. Esta opción puede también llevar un *ACK* para el *DSN* recibido.

Con MPTCP, todos los sub-flujos comparten el mismo espacio de memoria de recepción y anuncian la misma ventana de recepción. Existen dos niveles de asentimiento en MPTCP. Los *ACK´s* normales de TCP que son usados en cada sub-flujo para informar de la recepción de segmentos enviados sobre el sub-flujo, independientemente de su *DSN*. Por otro lado, a nivel de conexión, ACK´s para el número de secuencia de datos. Estos asentimientos rastrean el flujo de *bytes* y deslizan la ventana del receptor. El mapeo del número de secuencia de datos y el *ACK* de los datos van en la opción *Data Sequence Signal (DSS)* que aparece en la figura 10.

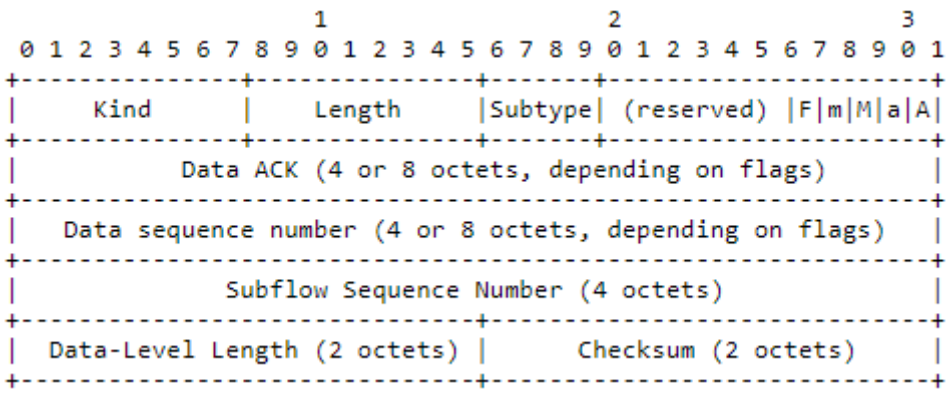

Figura 10: Opción Señal de Número de Secuencia de Datos (DSS). Fuente: RFC 6824

2.2.5 Solicitud de cambio en la prioridad de un camino

Los equipos pueden indicar, en la configuración inicial de un sub-flujo, si desean que se use como una ruta regular o una ruta de respaldo[4], la cual se usa si no hay rutas regulares disponibles. Durante una conexión, un equipo puede solicitar el cambio en la prioridad de un sub-flujo con la opción *MP_PRIO* que viene descrita en la figura 11.

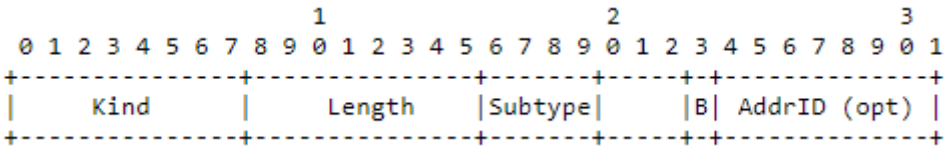

Figura 11: Opción para el cambio de prioridad de un sub-flujo (MP_PRIO). Fuente: RFC 6824

2.2.6 Cierre de una conexión MPTCP

Cuando un equipo quiere informar al otro de que no tiene más datos que enviar, le envía un *FIN* de los datos como parte de un *DATA_SEQUENCE_SIGNAL,* como se observa en la figura 12. Tiene el mismo significado y comportamiento que un *FIN* de TCP, pero a nivel de conexión. Una vez que todos los datos en MPTCP han sido enviados correctamente, entonces este mensaje de *FIN* es reconocido a nivel de conexión con un *DATA_ACK.*

[4] Una ruta regular es aquella en la que se envían los datos en la mayoría de ocasiones, de forma ordinaria, y una ruta de respaldo es aquella en la que sólo se envían los datos si no hay una ruta regular activa.

Ambos extremos deben enviar estos mensajes de *FIN* a todos los sub-flujos como cortesía, para que los equipos de encaminamiento intermedios limpien el estado incluso si un sub-flujo individual ha fallado. Cualquier sub-flujo donde todavía haya datos en cola, que hayan sido transmitidos en otros sub-flujos para obtener el asentimiento de *DATA_FIN,* puede cerrarse con un *RST.*

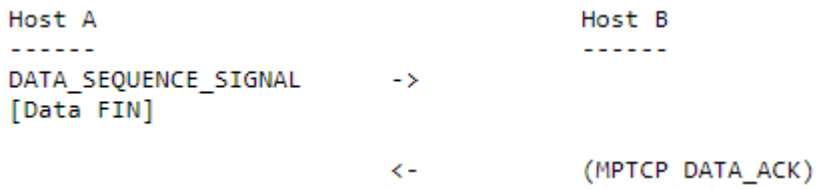

Figura 12: Cierre de la conexión. Fuente: RFC 6824

2.3 Aplicaciones de MPTCP

El protocolo MPTCP puede ser aplicado a un amplio abanico de aplicaciones, y por ello, a continuación, se presentan las situaciones en las que es interesante su uso y se explica su utilidad.

- **Dispositivos móviles:** Es una de las aplicaciones más utilizadas y presente en la mayoría de dispositivos hoy en día de forma más o menos extendida. Consiste en que el dispositivo esté conectado mediante dos interfaces diferentes, una a la red móvil de datos que ofrece un operador y la otra mediante una conexión Wi-Fi. De esta forma si se finaliza una de las dos conexiones, el dispositivo puede seguir comunicándose mediante la otra interfaz de forma transparente para el usuario y sin una pérdida o retraso de los datos. Además, mediante el uso de las dos interfaces nos permite aumentar el ancho de banda disponible en el dispositivo.

  En la actualidad este servicio está disponible tanto para iOS como para Android, aunque en este último depende de la capa de personalización de cada fabricante y, por ejemplo, en el caso de Samsung en la versión 9 de Android sobre *One UI* 1.1 (capa de personalización de Samsung), la función está limitada solamente para descargas superiores a 30 MB de Galaxy Store y Google Play Store y se denomina *Download Booster* [38].
- **Itinerancia en redes Wi-Fi:** En este caso, el usuario estaría conectado a dos puntos de acceso al mismo tiempo que estuviesen en cobertura con su dispositivo. De esta manera cuando el usuario se disponga a desplazarse y pierda cobertura con un punto de acceso, continuará conectado con el otro y no tendrá pérdidas ni retrasos en la comunicación. Este caso de uso en *roaming* también es interesante, ya que ha habido varios estándares para lograr una movilidad entre puntos de

acceso con transferencia rápida y segura de unos a otros, como se publicó en la norma 802.11r [11].

- **Centros de datos:** Fue el primer caso en el que se documentó el uso de MPTCP, como se puede ver en la bibliografía [36] y [28]. Se consiguió antes de llegar a la RFC 6824 final y permitió un mayor rendimiento agregado, ya que explorando más rutas y balanceando su carga adecuadamente se reduciría el número de enlaces inactivos e inutilizados. También permitió una mayor igualdad, ya que el rendimiento de las diferentes conexiones MPTCP debería ser mejor que el de TCP al distribuirse la congestión en los enlaces de las centrales de la red más equitativamente. Por último, permitió una mejor robustez al fallar un conmutador o un enlace ya que, a pesar de que con TCP se encaminará por otra ruta también, ello conllevará un tiempo. Sin embargo, con MPTCP la conexión continuará por otro camino sin pausa. Incluso en el caso de que el fallo de la red no provoque tener que volver a encaminar, pero cause que la auto-negociación del enlace regrese a una velocidad de enlace baja o tasas de pérdidas de paquetes muy altas. En este último caso, el protocolo TCP al tener solamente un camino, no tiene más remedio que enviar los datos por el camino lento, mientras que MPTCP puede evitar el envío de tráfico por un camino congestionado.

2.4 Problemática del uso de MPTCP

MPTCP fue diseñado para ser desplegado en el mundo y las redes actuales. Por ello tuvo en cuenta la existencia de dispositivos de red intermedios. En esta sección se va a proceder a describir algunos escenarios de fallos debidos a estos dispositivos y cómo los maneja MPTCP.

Uno de los principales fallos que cometen los dispositivos intermedios es que algunos no reenvían paquetes con opciones desconocidas, y MPTCP hace uso de una nueva opción en TCP llamada *Kind.* Sin embargo, todos los mensajes están definidos como subtipos, como se puede ver en la figura 13 y se han explicado los más importantes en la sección 2.2 del presente trabajo, por lo que debería reducir las posibilidades de que solamente se envíen algunos tipos de opciones de MPTCP.

Value	Symbol	Name
0x0	MP_CAPABLE	Multipath Capable
0x1	MP_JOIN	Join Connection
0x2	DSS	Data Sequence Signal (Data ACK and data sequence mapping)
0x3	ADD_ADDR	Add Address
0x4	REMOVE_ADDR	Remove Address
0x5	MP_PRIO	Change Subflow Priority
0x6	MP_FAIL	Fallback
0x7	MP_FASTCLOSE	Fast Close

Figura 13: Subtipos de opciones de MPTCP. Fuente RFC 6824

Los paquetes *MPTCP SYN,* en el primer sub-flujo de una conexión, contienen la opción *MP_CAPABLE,* que es la declaración de que un equipo soporta *multipath* TCP. Si este paquete es cambiado por un dispositivo intermedio, MPTCP volverá a una conexión TCP regular como podemos ver en la figura 15. Si los paquetes con la opción *MP_JOIN* son eliminados, ese camino no se utilizará.

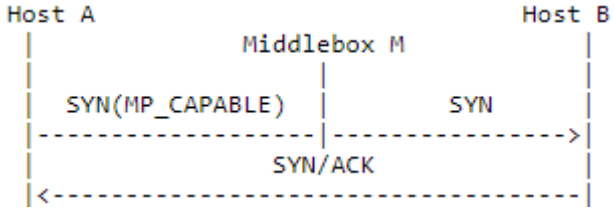

Figura 14: Opción MP_CAPABLE atraviese el dispositivo intermedio. Fuente: RFC 6824

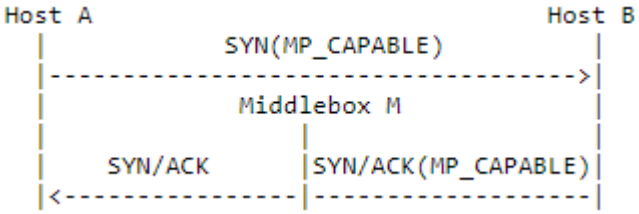

Figura 15: Opción MP_CAPABLE no atraviesa el dispositivo intermedio y es eliminada. Fuente: RFC 6824

Para finalizar esta sección se enumeran los dispositivos que pueden alterar los paquetes con la opción MPTCP activa o realizar modificaciones que afecten al protocolo: *NATs, Proxies,* Normalizadores de tráfico, Firewalls, Sistemas de detección de intrusos o dispositivos intermedios de nivel de aplicación. Cuando el protocolo detecta alguna anomalía en estos dispositivos, vuelve a TCP.

3 IMPLEMENTACIONES DE MPTCP Y PRUEBAS

Hay numerosas implementaciones de MPTCP para varios sistemas operativos diferentes, pero el presente trabajo fin de grado se centra en el sistema operativo Linux, concretamente en la distribución Ubuntu 18.04.2 LTS (Soporte de Larga Duración).

3.1 Implementación de MPTCP en Linux

El desarrollo en Linux de MPTCP comenzó dentro de un grupo de la Universidad de Louvain compuesto principalmente por el Dr. Sébastien Barre y el Dr. Christoph Paasch [14] en el año 2009. Para el presente trabajo se va a hacer uso de diferentes materiales disponibles en [14], gracias a la licencia *Creative Commons Attribution ShareAlike 4.0 International License*. Este grupo de trabajo ha continuado activo, teniendo la última publicación en su blog fecha de 21 de diciembre de 2018 y su última versión estable para implementar ha sido la v0.94 publicada el 9 de marzo de 2018, que ha sido utilizada para este documento y cuyos contribuidores en orden de mayor número de aportaciones son:

- Christoph Paasch (Apple).
- Hoang Tran (UCLouvain).
- Benjamin Hesmans (UCLouvain).
- Alexander Frömmgen (TU Darmstadt).
- Henrique Cabral (Veniam).
- Francois Finfe (Tessares).

El proyecto de esta universidad, se trata de un parche para el kernel de Linux de forma que se le añaden nuevos módulos dentro del módulo de red del sistema operativo para integrar MPTCP.

Las distribuciones de Linux para las cuales está disponible este proyecto son sistemas basados en Debian, sistemas basados en la herramienta de administración de paquetes GNU/Linux (RPM), Fedora 19 hasta 21, CentOS, Gentoo, ArchLinux, OpenSUSE y NixOS. Algunas de ellas están disponibles mediante enlaces a diferentes proyectos en *GitHub*.

Para la realización de las pruebas necesarias del presente proyecto, se ha optado por la instalación en el sistema operativo Ubuntu 18.04.2 LTS (*Long Term Support*). En cuanto al método de implementación de la modificación del kernel necesaria, se ha elegido la instalación mediante el repositorio *Advanced Packaging Tool* (APT) que mantiene el grupo de trabajo de la universidad de Louvain. En la sección 8.1 se puede ver el proceso para la instalación.

3.2 Otras implementaciones

A pesar de que la implementación más extendida es la discutida anteriormente, hay otras disponibles para diferentes sistemas operativos y utilidades. A continuación se muestran algunas de ellas además del responsable o responsables de su desarrollo:

- Android : Gregory Detal escrbió el paquete inicial y Matthieu Baerts y Quentin De Coninck lo extendieron [21] (Universidad de Louvain).
- OpenWRT: Hay tres versiones de Yannick Chabanois [10], Mario Krüger [26], Andrea Ghittino [23].
- PlanetLab: Felición Németh [30] (Universidad de Budapest).
- Amazon EC2 (*Elastic Compute Cloud):* Victor S. Andrei [1].
- Raspberry PI: Sunit Nandi [29].
- FreeBSD: Con los miembros Grenville Armitage, Nigel Williams, Lawrence Stewart, Radika Veera Valli y Jason But [22].
- Apple en iOS: Apple [6].

Como podemos comprobar, hay implementaciones de sistemas operativos abiertos en su mayoría, pero también propietarios como iOS. En este sistema operativo se implementó desde la versión 7, y se utiliza el protocolo MPTCP para establecer las peticiones que hace el usuario tanto en el interfaz Wi-fi como en el de datos móviles para posteriormente seleccionar la interfaz más adecuada, teniendo preferencia la Wi-fi. El sistema operativo iOS utiliza MPTCP por defecto en la aplicación de su asistente virtual Siri, de forma que la conexión siempre comienza por Wi-fi, cuando está disponible, y no se envían datos por la red móvil, pero si la señal Wi-fi empeora o se pierde, se crea automáticamente un nuevo sub-flujo TCP en la interfaz de la red móvil. Este sub-flujo se elimina una vez que la conexión Wi-fi se reestablece o vuelve a ser estable [5].

3.3 Descripción de los equipos de prueba

Para llevar a cabo las pruebas necesarias en este proyecto se ha contado con el equipo portátil del alumno, un equipo de sobremesa de la Universidad de Jaén perteneciente al laboratorio docente del Departamento de Ingeniería de Telecomunicación L-136 en la E. P. S. de Linares, un router TP-Link y un router Energy System, además del acceso a todas las redes disponibles en el laboratorio que se detallarán debidamente en cada escenario de pruebas. A continuación, se va a describir las características más importantes de cada equipo, centrándose en las capacidades de red, enumerado anteriormente:

- Ordenador portátil del alumno:
    - Procesador Intel Core i7 4700HQ @2.40 GHz.

- Memoria Ram 12 GBytes DDR3 Dual Channel 798.1 MHz.
- Tarjeta Gráfica Nvidia GTX Geforce 806M 2 GBytes GDDR5.
- Tarjeta Wi-fi Intel ® Dual Band Wireless-AC 7260 con dirección física E8:2A:EA:7C:4B:C4 y velocidad máxima de 1300 Mbps.
- Tarjeta Realtek PCIe Gigabit Ethernet Family Controller con dirección física F8:A9:63:35:B9:60 y velocidad máxima de 1000 Mbits/s.

- Ordenador de sobremesa:
  - Procesador Inter Core I5 760. 2.8 GHz
  - Memoria RAM: 8 GBytes.
  - Realtek RTL8111/8168/8411 PCIE Gigabit Ethernet Controller (“enp2s0”). Velocidad máxima 1000 Mbps.
  - D-Link System DGE-528T Gigabit Ethernet Adapter (“enp5s2”). Velocidad máxima 1000 Mbps.
  - Tarjeta Wi-fi Ralink Conceptronic C54RU v3 802.11bg wireless adapter [Ralink RT2571W (“wlx0022f7003d52”).
  - Tarjeta Wi-fi Ralink Conceptronic C54RU v3 802.11bg wireless adapter [Ralink RT2571W] (“wlx0022f70043aa”).
- Router TP-Link (Red: “telema2”):
  - Modelo: TL-WR841ND.
  - Estándar utilizado: 802.11b/g/n.
  - Velocidad máxima teórica: 300 Mb/s.
- Router Energy Sistem (Red “Multimedia”):
  - Modelo: WNR2000.
  - Estándar utilizado: 802.11b/g/n.
  - Velocidad máxima teórica: 300 Mb/s.

3.4 Problemática de la ejecución

A la hora del desarrollo de este proyecto se han tenido varias dificultades a lo largo del mismo y que se van a desarrollar a continuación, ya que se pretende que este proyecto sea replicable y ampliable en un futuro.

El primer problema que surgió fue en la implementación del kernel, el cual era necesario modificar para poder habilitar el protocolo MPTCP. La instalación del mismo se puede ver en la sección 8.1 y no requiere mayor dificultad. Sin embargo, una vez instalado, es necesario reiniciar el equipo e iniciarlo seleccionando el kernel instalado, concretamente la versión 4.14.110.mptcp, y se encontró un error al cargalo debido a que el sistema nos advertía de que el kernel no tenía una firma correcta. Tras una búsqueda, tanto en foros

como en páginas web [24], se determinó que la solución era deshabilitar el inicio seguro en la BIOS del equipo, LENOVO versión 9ECN36WW(V2.00), de forma que no hiciese comprobación de la firma. Hay que señalar que en el equipo del laboratorio no ha sido necesario realizar esta modificiación y que depende únicamente de la BIOS que tenga el sistema.

Otra problemática existente fue la instalación de la herramienta *IProute2,* cuya utilización se explica en la sección 9.1.2. Esta utilidad, viene comprendida dentro de la instalación del kenel que hemos utilizado y para su instalación solamente sería necesario proceder como en la sección 9.1.1 actualizando el sistema operativo y sus utilidades. Sin embargo no se ha podido realizar de esta forma debido a que el sistema operativo empleado disponía de una versión de la herramienta posterior. Esta dificultad se solucionó mediante la búsqueda en foros [8], en los que se explicaba que era posible que una versión posterior no dejase el libre funcionamiento de la versión que se requería. Efectivamente así fue, el sistema tenía instalada una versión de *Iproute* posterior a la versión de *Iproute2* que se intentaba instalar, por lo cual no era posible. Finalmente se consiguió y se puede comprobar la solución en la sección 9.1.2.

3.5 Herramientas utilizadas

A continuación se va a enumerar las herramientas utilizadas para realizar las mediciones de las pruebas y comprobar el funcionamiento del protocolo:

- Wireshark: Se conoció a través de la bibliografía existente que había una extensión del programa para soportar el protocolo MPTCP. Sin embargo, posteriormente se descubrió que a partir de la versión 1.7.1 del programa se soporta el protocolo por defecto. Gracias a esta herramienta se ha podido comprobar cómo se generaban y se recibían los paquetes MPTCP, de forma que el protocolo estaba activo, y las direcciones de las máquinas con las que se establecían conexiones.
- Iproute2: Se instaló una extensión de esta herramienta provista por el equipo de desarrollo de la misma página en la cual se ha obtenido la modificación del kernel [12]. Esta extensión permite activar o desactivar el modo MPTCP de una interfaz o configurar una concreta como interfaz de *backup.*
- Netstat: Con la instalación de la extensión de la herramienta *net-tools*, se ha podido comprobar el estado de los sub-flujos existentes así como datos de utilidad de los mismos.
- Iperf3: Con esta herramienta se han medido los datos de la conexión FTP entre el ordenador del alumno y el sobremesa en los intercambios de archivos.

- Wi-fi Scanner: Se trata de un programa para el sistema operativo Windows 10 con la que se pueden analizar numerosas características de las redes Wi-fi. Se ha utilizado una versión de evaluación del mismo con la que se ha analizado el uso de las bandas de 2.4 GHz y 5 Ghz en el lugar de pruebas, así como otras características de los puntos de acceso que emitían en el lugar.

4 EJECUCIÓN Y RESULTADOS DE LAS PRUEBAS

En esta sección vamos a proceder a mostrar los resultados de las pruebas para poder analizar el protocolo MPTCP y verificar sus posibles ventajas frente a TCP.

4.1 Introducción y descripción de las pruebas

Para comenzar con estas pruebas, se decidió hacer un estudio del espectro en frecuencia de la zona en la que se iban a llevar acabo, la cual era el laboratorio L-136 del Campus Científico-Tecnológico de la Universidad de Jaén en Linares. Este estudio nos alojó los siguientes resultados acerca de las bandas ocupadas en 2.4 GHz y 5GHz y la intensidad de las mismas. A continuación vemos el listado de puntos de acceso que emitían a nuestro alcance y la ocupación de las bandas.

Graph	Network Name (SSID)	Signal Strength	Signal Quality	MAC Address (BSSID)	Comment	Vendor	Achievable Rate	Max Rate	Network Mode	Channel
	eduroam	**-70 dBm**	**99%**	**B0-AA-77-8E-FA-7F**		**Cisco Systems, Inc**	**90 Mbps**	**600 Mbps**	**a, n, ac**	**0**
	INV	-44 dBm	93%	F0-79-59-D4-FA-08		ASUSTek COMPUTE...	216,7 Mbps	216,7 Mbps	b, g, n	11
	telema2	-44 dBm	93%	18-A6-F7-59-F1-44		TP-LINK TECHNOLO...	300 Mbps	300 Mbps	b, g, n	9,-1
	INV_5G	-51 dBm	81%	F0-79-59-D4-FA-0C		ASUSTek COMPUTE...	1170 Mbps	1733,3 Mbps	a, n, ac	42
	WiFi-Config	-67 dBm	55%	B0-AA-77-8E-FA-71		Cisco Systems, Inc	86,7 Mbps	216,7 Mbps	g, n	1
	Teleco2	-67 dBm	55%	B0-AA-77-8E-FA-72		Cisco Systems, Inc	86,7 Mbps	216,7 Mbps	g, n	1
	eduroam	-68 dBm	53%	B0-AA-77-8E-FA-70		Cisco Systems, Inc	86,7 Mbps	216,7 Mbps	g, n	1
	Teleco2	-69 dBm	51%	B0-AA-77-8E-FA-7D		Cisco Systems, Inc	90 Mbps	600 Mbps	a, n, ac	0
	WiFi-Config	-69 dBm	51%	B0-AA-77-8E-FA-7E		Cisco Systems, Inc	90 Mbps	600 Mbps	a, n, ac	0
	Despacho.b	-70 dBm	50%	FA-8F-CA-7D-C4-1A			21,7 Mbps	72,2 Mbps	b, g, n	7
	multimedia	-71 dBm	48%	74-44-01-88-80-F2		NETGEAR	43,3 Mbps	144,4 Mbps	b, g, n	12
	WiFi-Config	-75 dBm	41%	B0-AA-77-B0-16-D1		Cisco Systems, Inc	43,3 Mbps	216,7 Mbps	g, n	6
	telema2	-76 dBm	40%	B0-4E-26-E5-76-51		TP-LINK TECHNOLO...		300 Mbps	b, g, n	9,-1
	Teleco2	-76 dBm	40%	B0-AA-77-B0-16-D2		Cisco Systems, Inc	21,7 Mbps	216,7 Mbps	g, n	6
	eduroam	-76 dBm	40%	B0-AA-77-B0-16-D0		Cisco Systems, Inc	21,7 Mbps	216,7 Mbps	g, n	6
	Teleco2	-79 dBm	35%	B0-AA-77-B0-16-DD		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	eduroam	-80 dBm	33%	B0-AA-77-B0-16-DF		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	WiFi-Config	-81 dBm	31%	B0-AA-77-B0-16-DE		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	eduroam			B0-AA-77-A4-95-71		Cisco Systems, Inc		216,7 Mbps	g, n	1
	WiFi-Config			B0-AA-77-A4-95-70		Cisco Systems, Inc		216,7 Mbps	g, n	1
	WiFi-Config			B0-AA-77-A4-95-7F		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	eduroam			B0-AA-77-A4-95-7E		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	WiFi-Config			B0-AA-77-67-DF-AF		Cisco Systems, Inc		600 Mbps	a, n, ac	0
	eduroam			B0-AA-77-67-DF-AE		Cisco Systems, Inc		600 Mbps	a, n, ac	0

Figura 16: Listado de Puntos de Acceso en el lugar de las pruebas. Fuente: Wi-fi Scanner

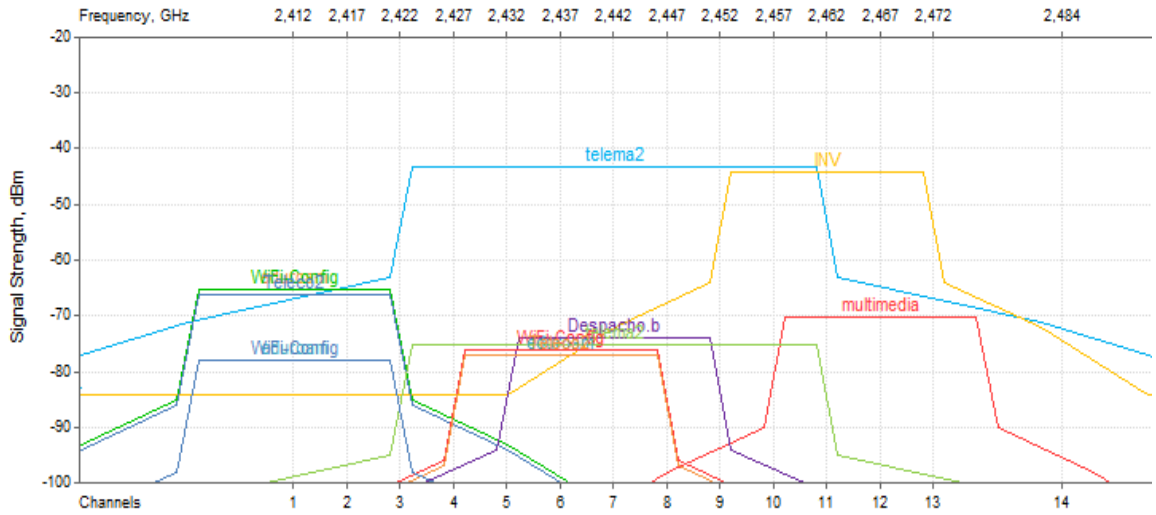

Figura 17: Canales ocupados e intensidad de señal de cada punto de acceso en la banda de 2.4 GHz. Fuente: Wi-fi Scanner

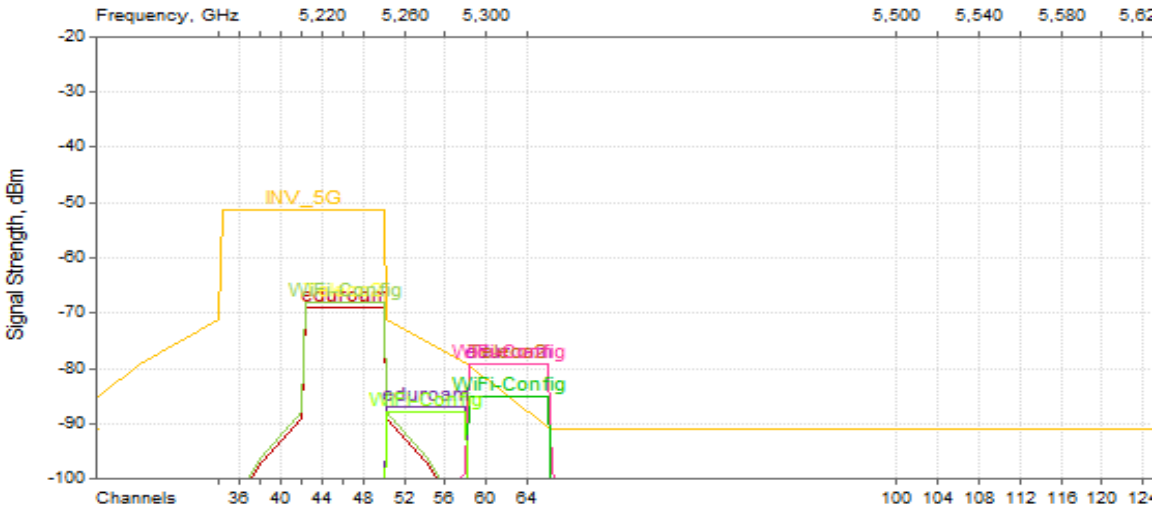

Figura 18: Canales ocupados e intensidad de señal de cada punto de acceso en la banda de 5 GHz. Fuente: Wi-fi Scanner

Como se puede observar en la gráfica, hay varias redes en la zona y algunas de ellas se encuentran en canales solapados, por lo que afectan a las interferencias y al rendimiento de las pruebas. En concreto, las dos redes inalámbricas utilizadas han sido “telema2” y “multimedia”, las cuales puede verse como tienen varias redes en los mismos canales emitiendo. Este inconveniente se puso de manifiesto en la realización de las pruebas, ya que en algunas ocasiones el rendimiento de la red era muy inferior a otras, también debido al uso y los usuarios conectados a la red. Hay que señalar que esta red no es la red principal del campus a la que se conectan la mayoría de usuarios, la cual es la red “*eduroam”* que aparece en la figura 15 con la mayor intensidad. La red utilizada para la realización de las pruebas es independiente de la red principal “eduroam” ya que se trata de una red diferente.

A esta red solamente se tiene acceso en los laboratorios. Para que estos aspectos no tuvieran gran influencia en el estudio, se realizaron las mediciones en varios días consecutivos y se eliminaron aquellos que presentaban métricas anómalas. Con esta metodología se ha podido observar que a primera hora de la mañana, entre las 8 y las 10, se obtenían los resultados más consecuentes y estables.

Por otra parte, para la realización de estas pruebas se ha decidido utilizar las métricas más usuales que se han podido observar en la bibliografía para estudios de este tipo, es decir, el rendimiento o ancho de banda de la conexión y el retraso de los paquetes desde que se envían hasta que vuelven al equipo origen (en inglés *Delay*).

Para las mediciones del rendimiento se ha utilizado la herramienta [3] ya que permite obtener el ancho de banda de varios sub-flujos como se necesitaba. Esta página web, en cada resultado que muestra, realiza una media de la suma de las mediciones que toma a lo largo de diez segundos de cada uno de los subflujos. Este resultado se muestra en Megabytes por segundo pero se ha optado por presentar los resultados en Megabits por segundo al resultar más sencillos de interpretar.

La otra métrica utilizada, el retraso de los paquetes, se ha obtenido mediante la herramienta *"ping"* desde el ordenador del laboratorio hasta la página web *"multipath-tcp.org"* , ya que se conoce que tiene habilitado el protocolo MPTCP en su servidor. También se ha estudiado la posibilidad de realizar esta métrica mediante las opciones de desarrollador de cualquier navegador web, pero sin embargo no eran precisas debido a que, en algunas ocaciones, el navegador obtenía los elementos de la página web de la caché al realizar la prueba de forma iterativa. Para esta métrica se hans utilizado las unidades en milisegundos. Además, con la utilidad usada, *"ping",* se ha establecido en diez[5] el número de repeticiones de eco. Posteriormente se ha calculado la media de esas diez repeticiones y se ha repetido el proceso treinta veces.

También hay indicar el procedimiento de medición de las pruebas, que ha sido la toma de datos de cada métrica en treinta ocasiones y se ha eliminado los valores del extremo superior e inferior para que el resultado de la prueba se normalizase y no contuviese datos que ocasionasen distorsiones en el estudio.

Señalar que, además, se ha decidido realizar las pruebas de ancho de banda solamente de bajada de la conexión ya que la herramienta que se ha seleccionado para poder medir el ancho de banda que soporta MPTCP no dispone de mediciones para la subida [3].

[5] Para poder realizar esta medición se ha utilizado el comando de la siguiente forma :
`ping multipath-tcp.org -n 10.`

Otro aspecto importante de los escenarios estudiados es que no se ha podido realizar las pruebas con la red de la Universidad de Jaén, debido a que dispone de un cortafuegos que no permite paquetes con opciones TCP, uno de los problemas del protocolo comentado en la sección 4.1. Por lo tanto se ha decidido utilizar para las pruebas una red propia de los laboratorios docentes del Departamento de Ingeniería de Telecomunicación en la E. P. S. de Linares sin esta limitación, que ya estaba habilitada.

Hay que destacar también que todas las pruebas realizadas son independientes, y por ello, a pesar de utilizar las mismas interfaces en cuanto a hardware y las mismas redes, los resultados no son exactamente iguales, sino que fluctúan entre unas pruebas y otras. Esto es debido al uso de la red, que se ha intentado minimizar eliminando los datos anómalos como se ha comentado anteriormente.

Se han tenido en cuenta los siguientes escenarios y pruebas para poder obtener una visión global acerca del protocolo MPTCP y sus ventajas. La tecnología de las interfaces cableadas de ambos ordenadores ha sido Gigabit Ethernet con el estándar 802.3ab [27], y la tecnología inalámbrica empleada ha sido 802.11n [33]:

- Conexión del equipo sobremesa con cuatro interfaces, dos cableadas y dos inalámbricas, a la red de los laboratorios de docencia.
- Conexión del equipo de sobremesa con una interfaz cableada y otra inalámbrica.
- Conexión del equipo de sobremesa con dos interfaces cableadas.
- Conexión del equipo de sobremesa con dos interfaces inalámbricas.
- Conexión del equipo de sobremesa con el portátil del alumno para realizar un intercambio con un servidor FTP propio, teniendo en un equipo un servidor MPTCP y en el otro un cliente.
- Simulaciones de caídas de una red durante la descarga de un archivo reales.

4.2 Escenario A – Cuatro interfaces en el PC del laboratorio

En el primer caso se ha conectado el ordenador del laboratorio con sus cuatro interfaces disponibles, dos cableadas y dos inalámbricas. Las direcciones IP de las mismas han sido asignadas dinámicamente mediante un servidor DHCP (*Dynamic Host Configuration Protocol*) y son las siguientes para cada interfaz, junto con su máscara y puerta de enlace:

- Enp2s0: 192.168.0.101 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.
- Enp5s2: 192.168.0.103 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.

- wlx0022f7003d52: 192.168.0.109 ; Máscara: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.
- wlx0022f70043aa: IP: 192.168.1.25 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.1.1.

Las dos interfaces cableadas y la interfaz inalámbrica wlx0022f7003d52 (conectada a la red inalámbrica "telema2"), se han conectado a la misma red: 192.168.0.0. La interfaz wlx0022f70043aa ha sido conectada a la red 192.168.1.0 ("multimedia"). Para medir ambas métricas se ha procedido según lo explicado en la sección 4.1, es decir, el rendimiento se ha medido con el test de velocidad experimental [3] y el retraso se ha obtenido mediante la herramienta *ping* enviando peticiones al servidor [14].

Tabla 1: Comparativa de MPTCP y TCP con cuatro interfaces (Rendimiento en Mb/s y Retraso en ms). Fuente: Elaboración propia

Interfaces	Protocolo	Métrica	Mínimo	Media	Máximo
Enp2s0	TCP	Rendimiento	8,72	9,687	12,37
		Retraso	51,922	52,284	52,938
	MPTCP	Rendimiento	2,997	3,292	6,154
		Retraso	51,864	52,663	55,150
Enp5s2	TCP	Rendimiento	8,184	9,885	11,269
		Retraso	51,9	52,802	53,9
	MPTCP	Rendimiento	4,003	4,552	5,887
		Retraso	51,837	52,577	54,546
wlx0022f7003d52	TCP	Rendimiento	5,109	6,248	7,07
		Retraso	61,136	70,865	100,734
	MPTCP	Rendimiento	2,516	3,137	3,581
		Retraso	53,121	65,633	78,267
wlx0022f70043aa	TCP	Rendimiento	6,287	8,371	9,21
		Retraso	61,927	93,587	204,351
	MPTCP	Rendimiento	2,764	2,932	3,25
		Retraso	52,873	67,167	81,406
Enp2s0 Enp5s2 wlx0022f7003d52 wlx0022f70043aa	MPTCP	Rendimiento	9,897	14,324	16,645
		Retraso	51,84	52,133	52,72

A la vista de los resultados, para comenzar, podemos indicar que hay algo que puede llamar la atención, y no es más que la diferencia de ancho de banda en todas las interfaces

cuando están en modo MPTCP o TCP, siendo en este último notablemente mayor. Este hecho es debido a dos cuestiones, la primera es que la herramienta utilizada para realizar el análisis del ancho de banda en modo MPTCP está aún en modo experimental, de forma que puede que no sea capaz de aprovechar todo el ancho de banda disponible por las cuatro interfaces. Sin embargo, se ha tenido que utilizar esta herramienta porque no hemos logrado encontrar ninguna otra que nos permitiese medir el ancho de banda de cuatro interfaces de forma simultánea y, tras realizar numerosas pruebas, los resultados que hemos visto han sido consistentes. Por otro lado, hay que señalar que el protocolo MPTCP es en realidad una opción dentro del paquete TCP, por lo cual se incrementan las cabeceras que se están enviando y baja el rendimiento de la red, ya que pasamos de enviar únicamente información por un par formado por IP-dirección física a mandarla mediante cuatro interfaces con varios caminos dentro de cada una de ellas, por lo cual hay una sobrecarga en ello.

También hay que señalar que en las redes inalámbricas el rendimiento es menor, como cabe esperar viendo la sección anterior debido a la congestión de los canales inalámbricos. Sin embargo, si observamos el rendimiento medio obtenido en modo MPTCP con las cuatro interfaces conectadas vemos que es mayor que en cualquier caso de cada interfaz por separado, tanto en MPTCP como en TCP. Podemos comprobar que el resultado final de las cuatro interfaces en modo MPTCP (14.324 Mb/s), es aproximadamente la suma de cada una de ellas en modo MPTCP, concretamente un 97%.

Si se analizan los resultados del retraso o *delay,* se puede observar que los de las dos interfaces cableadas son muy similares y los de las dos inalámbricas por su parte también. En las pruebas, al tomar los datos de las interfaces inalámbricas se obtuvo en algunos casos una pérdida de paquetes, debido a la congestión de canales mencionada anteriormente.

Si nos centramos en los resultados de MPTCP con las cuatro interfaces, podemos ver que el resultado medio en cuanto al retraso, es el menor de todos los medidos y similar al obtenido con la primera interfaz cableada. Esto nos indica que con este protocolo, el retraso que tenemos será el menor que haya en cada uno de los caminos existentes.

4.3 Escenario B – Dos interfaces en el PC del laboratorio

En el segundo escenario se ha conectado el ordenador del laboratorio mediante una interfaz cableada y otra inalámbrica, ambas en la misma red. Para la obtención de las métricas se ha procedido según la sección 4.1. La configuración de red de los equipos ha sido la siguiente:

- Enp2s0: IP: 192.168.0.104 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.
- wlx0022f7003d52: IP :192.168.0.109 ; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1.

Tabla 2: Comparativa de MPTCP y TCP con una interfaz cableada y otra inalámbrica (Rendimiento en Mb/s y Retraso en ms). Fuente: Elaboración propia

Interfaces	Protocolo	Métrica	Mínimo	Media	Máximo
Enp2s0	TCP	Rendimiento	8,72	9,687	12,37
		Retraso	51,822	52,584	52,938
	MPTCP	Rendimiento	4,04	4,8824	6,536
		Retraso	51,885	52,440	53,351
wlx0022f7003d52	TCP	Rendimiento	4,33	5,342	7,011
		Retraso	66,582	71,521	76,659
	MPTCP	Rendimiento	2,256	2,814	4,248
		Retraso	60,855	65,672	73,899
Enp2s0 wlx0022f7003d52	MPTCP	Rendimiento	6,328	6,616	7,744
		Retraso	51,858	52,297	53,351

En este escenario podemos comprobar como el rendimiento baja significativamente al utilizar una red cableada y otra inalámbrica. Como podemos observar, el rendimiento de la red inalámbrica es bajo en este escenario y por ello el resultado obtenido con las dos interfaces en modo MPTCP es finalmente de 6.616 Mb/s. Sin embargo, volvemos a comprobar al igual que en el apartado anterior, que la suma de las dos interfaces en modo MPTCP da aproximadamente el rendimiento total obtenido, siendo superior a cada una de ellas por separado.

También hay que destacar que, en este escenario, el rendimiento obtenido en MPTCP con las dos interfaces ha sido inferior al rendimiento de la interfaz “enp2s0” en modo TCP, concretamente un 32% menor. Esto es debido a la pérdida de paquetes en la red inalámbrica, ya que la zona de pruebas se ha visto que tenía los canales congestionados, y a la herramienta de medición, que como se ha comentado en la sección anterior no llega a aprovechar todo el ancho de banda disponible por las dos interfaces. No obstante, hemos

podido comprobar con la herramienta *ifstat*[6] que el funcionamiento de las dos interfaces era correcto y ambas estaban transmitiendo datos.

En cuanto al retardo sufrido por los paquetes en la red, puede verse en este caso que en la red inalámbrica es muy alto comparado con la cableada y el resultado obtenido finalmente con MPTCP es el mejor caso de todos ellos, observando el resultado medio.

4.4 Escenario C – Dos interfaces cableadas en el PC del laboratorio

En este caso se ha procedido a conectar las dos interfaces cableadas del ordenador del laboratorio y se han obtenido las métricas accediendo a los servicios disponibles en la página [14] y según lo explicado en la sección 4.1. Para ello se han configurado los equipos de la siguiente manera:

- Enp2s0: 192.168.0.101 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.
- Enp5s2: 192.168.0.103 ; Máscara de subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.

Tabla 3: Comparativa de MPTCP y TCP con dos interfaces cableadas (Rendimiento en Mb/s y Retraso en ms). Fuente: Elaboración propia

Interfaces	Protocolo	Métrica	Mínimo	Media	Máximo
Enp2s0	TCP	Rendimiento	8,72	9,687	12,37
		Retraso	51,922	52,284	52,938
	MPTCP	Rendimiento	3,672	4,2152	4,8
		Retraso	51,808	52,1	52,758
Enp5s2	TCP	Rendimiento	8,184	9,885	11,269
		Retraso	51,9	52,302	52,9
	MPTCP	Rendimiento	3,096	4,076	5,576
		Retraso	51,971	52,277	52,951
Enp2s0 Enp5s2	MPTCP	Rendimiento	6,712	8,3368	9,552
		Retraso	51,881	52,134	53,36

En este caso se puede ver como el rendimiento obtenido sigue con la tendencia explicada en los dos escenarios anteriores y es la suma de las dos interfaces en modo MPTCP prácticamente, teniendo en cuenta el valor medio. Además el ancho de banda conseguido

[6] Herramienta encargada de medir el tráfico instantáneo de entrada y salida de cada una de las interfaces del equipo.

de 8.3368 Mb/s es un 14% menor que el obtenido con la primera interfaz en TCP, lo cual es mejor resultado que en el apartado anterior, debido a que las dos redes son cableadas.

En cuanto al retardo sufrido por los paquetes, puede observarse nuevamente que el mejor caso lo obtenemos con las dos interfaces en modo MPTCP para la métrica media. Además se consigue,en cuanto al tiempo mínimo obtenido en cada uno de los modos, el mejor (51.881 ms) con las interfaces en MPTCP conjuntamente.

4.5 Escenario D – Dos interfaces inalámbricas en el PC del laboratorio

En este escenario se ha conectado el ordenador del laboratorio solamente de forma inalámbrica mediante sus dos interfaces disponibles. En este caso, cada una de las interfaces está en una red diferente. Para la obtención de las métricas se han utilizado las herramientas justificadas en la sección 4.1 y se han configurado los equipos como se puede observar:

- wlx0022f7003d52: IP: 192.168.0.109 ; Máscara de Subred: 255.255.255.0 ; Puerta de enlace: 192.168.0.1.
- wlx0022f70043aa: IP: 192.168.1.25 ; Máscara de Subred: 255.255.255.0 ; Puerta de enlace: 192.168.1.1.

Tabla 4: Comparativa de MPTCP y TCP con dos interfaces inalámbricas (Rendimiento en Mb/s y Retraso en ms). Fuente: Elaboración propia

Interfaces	Protocolo	Métrica	Mínimo	Media	Máximo
wlx0022f7003d52	TCP	Rendimiento	5,801	6,688	7,113
		Retraso	61,897	65,639	94,565
	MPTCP	Rendimiento	2,256	2,8144	4,248
		Retraso	56,789	66,78	94,588
wlx0022f70043aa	TCP	Rendimiento	2,867	3,561	4,102
		Retraso	61,927	93,587	204,351
	MPTCP	Rendimiento	0,832	0,9648	1,304
		Retraso	53,324	91,535	151,537
wlx0022f7003d52 wlx0022f70043aa	MPTCP	Rendimiento	2,956	3,828	4,432
		Retraso	56,714	65,652	79,515

Este escenario arroja los resultados más bajos de los estudiados, y es debido a que se ha comprobado el protocolo con dos interfaces inalámbricas. Hay que indicar que la interfaz

“wlx0022f70043aa” conectada a la red “multimedia” ha obtenido datos mucho inferiores a la conectada a la red “telema2”, puesto que estaba alojado el punto de acceso en un laboratorio contiguo. Sin embargo, es interesante conocer el funcionamiento también en un escenario de este tipo, con una red más alejada. El resultado obtenido ha sido el esperado. Su rendimiento ha sido muy bajo y en cuanto al retraso en los paquetes medido, hay que indicar que se perdían aproximadamente un 13% de los paquetes enviados.

El rendimiento obtenido en modo MPTCP con las dos interfaces ha sido inferior que en el caso de la interfaz “wlx0022f7003d52” en modo TCP, pero muy superior al caso de la interfaz ”wlx0022f70043aa” en modo MPTCP. También se puede comprobar nuevamente como el rendimiento obtenido por ambas interfaces en modo MPTCP es la suma aproximada de cada una de ellas con el protocolo MPTCP.

En cuanto al retraso, se comprueba de nuevo que el resultado medio en MPTCP con las dos interfaces, se asemeja al mejor de los dos dados por cada interfaz de forma separada.

4.6 Escenario E – Cuatro interfaces en el PC del laboratorio y dos interfaces en el portátil

En este caso se ha conectado el equipo sobremesa con el portátil del alumno para realizar un intercambio con un servidor FTP propio, teniendo en un equipo un servidor MPTCP y en el otro un cliente. La configuración realizada es la siguiente:

- Ordenador laboratorio:
  - Enp2s0: IP: 192.168.0.104; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
  - Enp5s2: IP: 192.168.0.106; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
  - wlx0022f7003d52: IP: 192.168.0.109; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
  - wlx0022f70043aa: IP: 192.168.1.25; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
- Ordenador portátil alumno:
  - Enp9s0: 192.168.0.111; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
  - Wlp8sp: 192.168.0.110; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1

Para realizar este escenario se ha instalado en el equipo portátil el mismo sistema operativo y la modificación del kernel para MPTCP que en el equipo sobremesa. Además se ha

instalado un servidor FTP llamado *vsftpd* [16] para el intercambio de archivos en el portátil y un cliente en el sobremesa llamado *Filezilla* [25] y el cual viene instalado en la distribución usada por defecto. El proceso de configuración del servidor puede verse detallado en la sección 9.2.

Mediante este escenario se ha conseguido poner en práctica no solamente un equipo cliente de MPTCP para poder acceder a servidores compatibles con el protocolo, sino que además se ha logrado instalarlo en un servidor de modo que se pudiese estudiar el funcionamiento en este caso.

Con esta prueba se ha podido observar cómo se transferían recursos de un equipo a otro mediante MPTCP y se ha elaborado una tabla con los tiempos que se tardaba en obtener ese recurso.

Tabla 5: Comparativa de MPTCP y TCP descargando archivos mediante FTP de diferente tamaño (Tiempo en ms). Fuente: Elaboración propia

Interfaces Cliente	Interfaces Servidor	Protocolo	Tamaño Archivo	Mínimo	Media	Máximo
Enp2s0+ Enp5s2+ wlx0022f70043aa+ wlx0022f7003d52	Enp9s0+ Wlp8s0	MPTCP	6,2 MB	0,7	1	1,1
			306,2 MB	2	3,1	3,9
			2 GB	20	22,5	25
Enp2s0	Enp2s0	TCP	6,2 MB	0,7	1,1	1,2
			306,2 MB	2,1	3,2	3,8
			2 GB	21,1	23,2	24,9
Enp2s0+ wlx0022f70043aa	Enp9s0+ Wlp8s0	MPTCP	6,2 MB	0,7	1	1,2
			306,2 MB	2,4	3,6	5,0
			2 GB	20,1	24	29,4
Enp5s2	Enp9s0	TCP	6,2 MB	0,8	1,2	1,4
			306,2 MB	2,3	3,3	3,8
			2 GB	21,8	23,9	25,1
Enp2s0+ Enp5s2	Enp9s0+ Wlp8s0	MPTCP	6,2 MB	0,7	1	1.1
			306,2 MB	2,3	3,3	4,1
			2 GB	20,8	23,1	27,3
wlx0022f70043aa	Wlp8s0	TCP	6,2 MB	10	11,1	18,1

			306,2 MB	309	360	391
			2 GB	1918	2132	2341
wlx0022f70043aa+ wlx0022f7003d52	Enp9s0+ Wlp8s0	MPTCP	6,2 MB	8	9,7	15,1
			306,2 MB	298	340	383
			2 GB	1897	2021	2124
W2	Wlp8s0	TCP	6,2 MB	9,9	11,7	17,2
			306,2 MB	321	383	401
			2 GB	2039	2232	2513

Con estos resultados obtenidos, se observa que la mejora aportada mediante el protocolo MPTCP es más notable cuanto mayor es el archivo a descargar. También se pone de manifiesto el bajo rendimiento de las redes inalámbricas, como se ha comentado en escenarios anteriores, dando como resultado tiempos máximos en MPTCP con las dos redes inalámbricas de 2124 ms para descargar 2 GB de datos, frente a 25 ms con las cuatro interfaces en modo MPTCP en el peor caso.

4.7 Escenario F – Tres interfaces en el PC del laboratorio

En este escenario se ha simulado la caída de una red durante la descarga de un archivo, procediendo a desconectar algunas interfaces y descargando archivos de la página web que se ha utilizado para implementar el kernel modificado, ya que es el único sitio web que se ha encontrado con un servidor MPTCP implementado y archivos disponibles para su descarga. Hay que señalar que solamente disponía de archivos de 121 MB. También se ha decidido utilizar las dos interfaces cableadas y una inalámbrica debido a que la red “multimedia” usada para el resto de pruebas era muy inestable en este escenario y no influía en el resultado signficativamente, ya que al tratarse de la red más alejada la transferencia se realizaba por las otras tres interfaces en la mayoría de los casos. A continuación puede verse la configuración del ordenador del laboratorio.

- Enp2s0: IP: 192.168.0.104; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1
- Enp5s2: IP: 192.168.0.106; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1

- wlx0022f7003d52: IP: 192.168.0.109; Máscara de subred: 255.255.255.0; Puerta de enlace: 192.168.0.1

Tabla 6: Simulaciones de caídas de la red (Rendimiento en Mb/s). Fuente: Elaboración propia

Interfaces activas incialmente	**Interfaz/Interfaces desconectadas tras 10 segundos de descarga**	**Tamaño archivo**	**Mínimo**	**Medio**	**Máximo**
Enp2s0+Enp5s2+ wlx0022f7003d52	Ninguna	120,75 MB	25,817	27,652	29,732
Enp2s0+Enp5s2+ wlx0022f7003d52	Enp2s0	120,75 MB	26,014	28,102	32,158
Enp2s0+Enp5s2+ wlx0022f7003d52	Enp5s2	120,75 MB	25,961	29,107	35,862
Enp2s0+Enp5s2+ wlx0022f7003d52	wlx0022f7003d52	120,75 MB	27,834	29,917	33,812
Enp2s0+Enp5s2+ wlx0022f7003d52	Enp2s0+ wlx0022f7003d52	120,75 MB	28,293	28,701	31,988
Enp2s0+Enp5s2+ wlx0022f7003d52	Enp2s0 + Enp5s2	120,75 MB	55,439	61,547	72,597

Como se puede observar, el mejor resultado es con las tres interfaces en modo MPTCP en cualquier caso. Sin embargo es interesante comprobar como al hacer caer una red de modo "artificial"[7], la descarga prosigue y a penas toma unos segundos más para finalizar. En el útimo caso tarda aproximadamente el doble en la descarga, debido a que las redes que se "caen" son las dos cableadas y solamente se encuentra activa la red inalámbrica, que es la de menor desempeño. Aún así la descarga prosigue y se consigue finalizarla.

[7] Para simular este comportamiento, desactivamos una interfaz mediante:
`ifconfig "xx" down`

5 CONCLUSIONES Y LÍNEAS FUTURAS

Para finalizar este estudio, se presentan a continuación las conclusiones del mismo. Mediante el estudio teórico y las pruebas realizadas, se puede llegar a la conclusión de que el protocolo MPTCP es una mejora significativa de TCP, el cual aporta varias ventajas.

En primer lugar, y como solución más importante para el usuario final, sería el poder tener una conexión de respaldo, ya que como se ha visto en el escenario F, el protocolo permite continuar con una descarga o cualquier otra actividad con solamente una leve demora, y no interrumpiéndose la misma, como si tuviésemos una sola red hasta que ésta volviese a estar disponible. Además, esta ventaja sería de gran utilidad para cualquier empresa, puesto que le permite no dejar de ofrecer sus servicios a los clientes en el caso de fallo de una red. Más aún cuando la mayoría de empresas medianas o grandes disponen ya de varias conexiones contratadas con diferentes operadores para este fin, y la instalación del protocolo MPTCP en sus servidores no sería de gran dificultad, puesto que está disponible para numerosos sistemas operativos. También facilitaría su instalación el hecho de estar en la capa de transporte, sobre todo en aspectos concernientes con la seguridad, ya que normalmente la seguridad no se implementa en este nivel de red y por tanto no habría que modificarla. Solamente se necesitaría implementar en los firewalls las reglas requeridas para que no descartasen los paquetes de TCP con opciones, como se ha comentado en los problemas de este protocolo en la sección 4.4.

Esta visión del protocolo MPTCP como conexión de respaldo también sería de gran utilidad en los retos que encuentra actualmente la tecnología, como por ejemplo en los coches autónomos. Estos vehículos requieren de una latencia muy baja para poder tener mucha información y tomar decisiones en muy pocos milisegundos. Como se ha podido comprobar a lo largo de los escenarios A,B,C y D, el retardo que sufren los paquetes con este protocolo siempre es el mejor dadas todas las redes de las que dispone, por lo cual sería de gran utilidad en este tipo de servicios. Además, en estas situaciones es imprescindible tener esa conexión de respaldo debido a la seguridad que deben tener los ocupantes, ya que si el vehículo dispone de solamente una interfaz de red y se cae, dejaría de funcionar todo el sistema.

Otra gran ventaja del uso de este protocolo es que permite agregar ancho de banda de varias redes, por lo cual puede aportar una mayor velocidad. Este hecho en la actualidad no es tan relevante como hace unos años debido a que los usuarios disponen de redes de fibra óptica con velocidades de hasta 1 Gbp/s y de redes móviles 4G+ con velocidades teóricas de hasta 300 Mbp/s mediante agregación de canales y uso de varias antenas. Además, se está desarrollando el estándar 5G y las pruebas reales más recientes,

encontradas en Reino Unido, dan como resultado unas velocidades mínimas de 100 Mbp/s y máximas de 1 Gbp/s[19]. No obstante, esta ventaja sí que es de gran importancia en los centros de procesamiento de datos en los que se manejan gran cantidad de los mismos y se requiere un ancho de banda muy elevado, además de una baja latencia para la entrega de datos al usuario cuando lo requiere[37]. También es muy importante la mejora en el ancho de banda para el *Big Data*¸ ya que esta tecnología emergente depende de la recolección de muchos datos y requiere un ancho de banda considerable en según que aplicaciones[28].

A pesar de todas estas ventajas, el protocolo no esta implementado de forma extensiva en los sistemas operativos en la actualidad y solamente unos pocos servidores disponen de él. Por la información que se ha podido encontrar, en los dispositivos móviles iOS está habilitado mediante una opción para tener una red de respaldo y en los dispositivos Android depende de cada fabricante, pero Samsung, a modo de ejemplo, lo implementa solamente como una opción para descargas de Google Play Store y Samsung Galaxy Store de más de 30 MB. Por lo tanto, los servidores de estas compañías (Google, Apple y Samsung) sí que disponen de una implementación de este protocolo pero muy limitada a ciertos servicios.

Las desventajas de uso del protocolo son las comentadas en este documento anteriormente. En primer lugar, que algunos dispositivos de red pueden descartar estos paquetes TCP con opciones desconocidas pero, como se ha visto, el protocolo se encarga de volver a TCP normal y no se interrumpe la conexión, y en segundo lugar, la sobrecarga de cabeceras producida por las opciones introducidas por MPTCP en el campo opciones de TCP. El campo opciones de TCP suele tener un tamaño de 12 bytes, según los paquetes que se han podido observar con el analizador de tráfico *Wireshark*, y cuando llevan la información de MPTCP de 20 bytes. A la vista de los datos, la información de MPTCP supone un incremento del 40% en el campo opciones de TCP. No obstante, en la cabecera de TCP al completo supone un incremento del 23% aproximadamente, que es un dato destacable pero asumible dadas las ventajas explicadas anteriormente.

Otro aspecto importante en el que puede ser de gran utilidad este protocolo, es en la computación de juegos en la nube, una nueva forma de interpretar los videojuegos sin necesidad de tener una consola, sino con una aplicación únicamente y todo el procesado realizado en la nube. Este tipo de servicios emergentes está desarrollándose en la actualidad por la mayoría de compañías del sector y a finales de este año será lanzado por Google con su plataforma *google stadia* o Microsoft mediante su plataforma *xCloud* a comienzos del próximo año. En estos servicios será muy importante la conexión ya que se

requiere una latencia muy baja, que como se ha visto, nos puede aportar MPTCP seleccionando el menor retardo de todas las rutas posibles, y requerirán de un ancho de banda de al menos 35 Mb/s para poder jugar en 4k a 60 imágenes por segundo con Dolby Surround 5.1 [31].

En conclusión, se ha logrado mediante este trabajo la instalación de una implementación del protocolo en el sistema operativo de linux Ubuntu 18.04.2, que era el primer objetivo. Posteriormente se ha conseguido, a lo largo de los escenarios de pruebas de la sección 4, el análisis del protocolo en distintas circunstancias para poder realizar una evaluación del mismo. Con ellas, se ha llegado al estudio en profundidad del protocolo, otro de los objetivos marcados para este trabajo.

5.1 Líneas de futuro

Por otro lado, comentar acerca de las líneas de futuro, que una de las más interesantes de este proyecto sería la investigación de una posible instalación de forma sencilla en cualquier sistema operativo. Sería muy importante el hecho de conseguir una implementación para los sistemas operativos de Microsoft, ya que son de los más usados en el mundo actualmente. Esta línea de actuación puede ser muy interesante, si se tiene en cuenta que se ha conseguido que *Windows 10* tenga su propio *kernel* de *Linux* [20] [28]. Con la utilización de este kernel se podría modificar el funcionamiento de la pila de protocolos del sistema operativo, de forma similar a lo realizado para los sistemas basados en Linux, y así conseguir implementar el protocolo MPTCP.

6 ABREVIATURAS Y ACRÓNIMOS

APT: *Advanced Packaging Tool.*

ARPA: *Advanced Research Project Agency.*

ARPANET: *Advanced Research Projects Agency Network.*

DHCP: *Dynamic Host Configuration Protocol.*

DSN: *Data Sequence Number.*

DSS: *Data Sequence Signal.*

EC2: *Elastic Compute Cloud.*

HMAC: *Hash-based Message Authentication Code.*

IANA: *Internet Assigned Numbers Authority.*

IETF: *Internet Engineering Task Force.*

IoT: Internet de las Cosas

LTS: Soporte de larga duración.

MPTCP: *Multipath Operation Transport Control Protocol.*

NAT: Traducción de direcciones de red

RFC: *Request for Comments.*

RPM: *Red Hat Package Manager (*Herramienta de administración de paquetes GNU/Linux).

TCP: *Transport Control Protocol.*

7 BIBLIOGRAFÍA

[1] **Adrei, Victor S. 2013.** MultiPath TCP. [En línea] 9 de septiembre de 2013. [Citado el: 18 de mayo de 2019.] http://multipath-tcp.org/pmwiki.php/Users/AmazonEC2.

[2] **Alan Ford, C. R**. (30 de mayo de 2019). *IETF.* Obtenido de https://tools.ietf.org/html/draft-ietf-mptcp-rfc6824bis-17.

[3] **Alexander Frömmgen, Robert Rehner, Tobias Erbshäußer, and Maximilian Weller.** Multipath TCP Cheker. [En línea] [Citado el: 29 de mayo de 2019.] http://amiusingmptcp.de/#speedtest.

[4] **ÁLVAREZ MÉNDEZ, Sebastián**. *Análisis del protocolo MPTCP en plataformas Linux*. 2015. Tesis de Licenciatura.

[5] **Apple**. [En línea] agosto de 2017. [Citado el: 22 de mayo de 2019.]https://support.apple.com/es-es/HT201373.

[6] **Apple.** [En línea] [Citado el: 18 de mayo de 2019.] https://developer.apple.com/documentation/foundation/urlsessionconfiguration/improving_network_reliability_using_multipath_tcp.

[7] **ARZANI, Behnaz**, et al. Deconstructing MPTCP performance. En *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014. p. 269-274.

[8] **Bozakov, Z.** (21 de febrero de 2018). *Github*. Obtenido de https://github.com/multipath-tcp/iproute-mptcp/issues/6.

[9] **Caspar, Nils. 2013.** GitHub. [En línea] 28 de noviembre de 2013. [Citado el: 31 de mayo de 2019.] https://github.com/multipath-tcp/mptcp-scripts/blob/master/scripts/rt_table/mptcp_up.

[10] **Chabanois, Yannick. 2019.** Openmptcprouter. [En línea] 11 de mayo de 2019. [Citado el: 18 de mayo de 2019.] https://www.openmptcprouter.com/.

[11] **Chpalin, Clint, Montemurro, Michael y Marshall, Bill. 2008.** IEEE. [En línea] 15 de julio de 2008. [Citado el: 18 de mayo de 2019]. https://ieeexplore.ieee.org/document/4573292.

[12] **Christoph Paasch, Gregory Detal. 2015.** GitHub. [En línea] 14 de septiembre de 2015. [Citado el: 27 de mayo de 2019.] https://github.com/multipath-tcp/iproute-mptcp.

[13] **CORBILLON, Xavier**, et al. Cross-layer scheduler for video streaming over MPTCP. En *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016. p. 7.

[14] **C. Paasch, S. Barre**, et al., Multipath TCP in the Linux Kernel, disponible en http://www.multipath-tcp.org .

[15] **C. Paasch, S. Barre.** MultiPath TCP - Linux Kernel implementation. [En línea] [Citado el: 31 de mayo de 2019.] https://multipath-tcp.org/pmwiki.php/Users/HowToInstallMPTCP?.

[16] **Evans, C.** (s.f.). *vsftpd*. Obtenido de https://security.appspot.com/vsftpd.html#about .

[17] **FORD, Alan**, et al. *TCP extensions for multipath operation with multiple addresses*. 2013.

[18] **FORD, Alan**, et al. TCP Extensions for Multipath Operation with Multiple Addresses draft-ford-mptcp-multiaddressed-00. 2009.

[19] **García, A**. (31 de mayo de 2019). *ADSLZone.* Obtenido de https://www.adslzone.net/2019/05/31/prueba-5g-sin-datos/.

[20] **González, G.** (13 de junio de 2019). *Genbeta*. Obtenido de https://www.genbeta.com/windows/impensable-se-hace-realidad-windows-10-tiene-su-propio-kernel-linux .

[21] **Gregory Detal, Matthieu Baerts, Quentin De Coninck. 2015.** MulltPath TCP. [En línea] 16 de junio de 2015. [Citado el: 18 de mayo de 2019.] http://multipath-tcp.org/pmwiki.php/Users/Android?action=diff.

[22] **Grenville Armitage, Nigel Williams, Lawrence Stewart, Radika Veera Valli, Jason But.** Centre for Advanced Internet Architectures. [En línea] [Citado el: 18 de mayo de 2019.] http://caia.swin.edu.au/urp/newtcp/mptcp/.

[23] **Guittino, Andrea. 2016.** mytechpg.blogspot. [En línea] 04 de enero de 2016. [Citado el: 18 de mayo de 2019.] http://mytechpg.blogspot.com/2016/01/multipath-tcp-mptcp-for-openwrt.html.

[24] **Hoffman, C.** (17 de julio de 2017). *How-To Geek*. Obtenido dehttps://www.howtogeek.com/175641/how-to-boot-and-install-linux-on-a-uefi-pc-with-secure-boot/ .

[25] **Kosse, T.** (s.f.). *Filezilla. The free FTP solution*. Obtenido de https://filezilla-project.org/index.php .

[26] **Krüger, Mario. 2016.** OpenWrt – Wireless Freedom. [En línea] 19 de mayo de 2016. [Citado el: 18 de mayo de 2019.] https://oldwiki.archive.openwrt.org/doc/uci/mptcp.

[27] **LAW, David**, et al. Evolution of Ethernet standards in the IEEE 802.3 working group. *IEEE Communications Magazine*, 2013, vol. 51, no 8, p. 88-96.[28] Loewen, C. (30 de mayo de 2019). *Microsoft*. Obtenido de https://docs.microsoft.com/en-us/windows/wsl/wsl2-ux-changes#accessing-network-applications .

[28] **Marx, V.** (2013). The big challenges of big data. Nature. International journal of science, 255-260.

[29] **Nandi, Sunit. 2018.** Techno FAQ. [En línea] 2 de septiembre de 2018. [Citado el: 18 de mayo de 2019.] https://technofaq.org/posts/2018/09/how-to-compile-mptcp-linux-kernel-on-raspberry-pi-2-and-raspberry-pi-3-on-archlinux-arm/.

[30] **NÉMETH, Felición**, et al. A large-scale multipath playground for experimenters and early adopters. En *ACM SIGCOMM Computer Communication Review*. ACM, 2013.

[31] **Nieto, J. G.** (6 de junio de 2019). Xatakamóvil. Obtenido de https://www.xatakamovil.com/varios/estos-requisitos-conexion-para-usar-google-stadia-asi-puedes-comprobar-cumples

[32] **Paasch, Christoph. 2012.** GitHub. [En línea] 11 de enero de 2012. [Citado el: 31 de mayo de 2019.] https://github.com/multipath-tcp/mptcp-scripts/blob/master/scripts/rt_table/mptcp_down.

[33] **PERAHIA, Eldad**. IEEE 802.11 n development: History, process, and technology. *IEEE Communications Magazine*, 2008, vol. 46, no 7, p. 48-55.

[34] **PIRIS RUIZ**, Jaime, et al. Algoritmos para la búsqueda de múltiples rutas sobre redes inalámbricas malladas multi-interfaz. 2015.

[35] **POSTEL, Jon**, et al. RFC 791: Internet protocol. 1981.

[36] **RAICIU, Costin**, et al. Data center networking with multipath TCP. En *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010.

[37] **RAICIU, Costin**, et al. Improving datacenter performance and robustness with multipath TCP. En *ACM SIGCOMM Computer Communication Review*. ACM, 2011.

[38] **Samsung. 2019.** Samsung. [En línea] Samsung, 28 de 03 de 2019. [Citado el: 17 de 05 de 2019.] https://www.samsung.com/co/support/mobile-devices/galaxy-s10-what-is-it-and-how-to-activate-the-download-booster/.

[39] **SHIN, Sungjin**, et al. TCP and MPTCP retransmission timeout control for networks supporting WLANs. *IEEE Communications Letters*, 2016, vol. 20, no 5, p. 994-997.

8 ANEXO A. Instalación y configuración del kernel

8.1 Instalación

Para la instalación de esta modificación del núcleo del sistema operativo hay numerosas opciones que pueden consultarse [15]. En este caso se ha procedido a instalarlo mediante la opción “Instalación automática en tú equipo con nuestro repositorio-apt” y para ello son necesarios los siguientes pasos:

1. Agregar las claves necesarias para la instalación

```
sudo apt-key adv --keyserver hkp://keys.gnupg.net --recv-keys 379CE192D401AB61
```

2. Añadir el repositorio en el archivo fuente de repositorios admitidos por nuestro sistema.

```
sudo sh -c "echo 'deb https://dl.bintray.com/cpaasch/deb stretch main' > /etc/apt/sources.list.d/mptcp.list"
```

3. Proceder a la instalación

```
sudo apt-get update

sudo apt-get install linux-mptcp
```

Tras la realización de estos pasos es necesario reiniciar el sistema y elegir en el arranque la opción *Linux 4.14.110.mptcp.*

8.2 Configuración

Para realizar la configuración se han utilizado dos scripts que pueden usarse para que realicen la configuración de forma automática, tanto para cuando se activan [9] como cuando se desactivan [32] las interfaces. En cada uno de ellos se especifica el lugar donde hay que situarlos y son los siguientes:

MPTCP_UP

```
#!/bin/sh

# A script for setting up routing tables for MPTCP in the N950.

# Copy this script into /etc/network/if-up.d/

set -e

env > /etc/network/if_up_env

if [ "$IFACE" = lo -o "$MODE" != start ]; then
```

```
        exit 0
fi
if [ -z $DEVICE_IFACE ]; then
      exit 0
fi
# FIRST, make a table-alias
if [ `grep $DEVICE_IFACE /etc/iproute2/rt_tables | wc -l` -eq 0 ]; then
      NUM=`cat /etc/iproute2/rt_tables | wc -l`
      echo "$NUM  $DEVICE_IFACE" >> /etc/iproute2/rt_tables
fi
if [ $DHCP4_IP_ADDRESS ]; then
      SUBNET=`echo $IP4_ADDRESS_0 | cut -d \   -f 1 | cut -d / -f 2`
      ip route add table $DEVICE_IFACE to $DHCP4_NETWORK_NUMBER/$SUBNET dev
$DEVICE_IFACE scope link
      ip route add table $DEVICE_IFACE default via $DHCP4_ROUTERS dev
$DEVICE_IFACE
      ip rule add from $DHCP4_IP_ADDRESS table $DEVICE_IFACE
else
      # PPP-interface
      IPADDR=`echo $IP4_ADDRESS_0 | cut -d \   -f 1 | cut -d / -f 1`
      ip route add table $DEVICE_IFACE default dev $DEVICE_IP_IFACE scope link
      ip rule add from $IPADDR table $DEVICE_IFACE
fi
```

MPTCP_DOWN

```
#!/bin/sh
# A script for setting up routing tables for MPTCP in the N950.
# Copy this script into /etc/network/if-post-down.d/
set -e
env > /etc/network/if_down_env
if [ "$IFACE" = lo -o "$MODE" != stop ]; then
        exit 0
fi
ip rule del table $DEVICE_IFACE
```

```
ip route flush table $DEVICE_IFACE
```

9 ANEXO B. Instalación de herramientas y utilidades

9.1 Instalación de herramientas

9.1.1 Wireshark

Se puede instalar mediante la actualización de los repositorios del kernel utilizado, ya que viene incluido en ellos, mediante la ejecución de:

```
sudo apt-get update && apt-get dist-upgrade
```

9.1.2 Iproute2 (extensión de iproute)

Esta herramienta también viene incluída en el kernel instalado, sin embargo, como se ha explicado en la sección 3.4, no se puede hacer uso de ella porque el sistema operativo la sustituye por una versión de *Iproute* más actualizada. Por lo tanto, para su instalación es necesario:

1. Conocer dónde está ubicado el paquete ip

```
sudo which ip
```

2. Encontrar el paquete que nos proporciona el comando para instalar iproute2

```
sudo dpkg -S /sbin/ip
```

3. Encontrar las versiones disponibles

```
sudo apt-cache madison iproute2
```

4. Bajar de la versión instalada por el sistema operativo, a la versión necesaria de iproute2.

```
sudo apt-get install iproute2=XXXXXX-.
```

9.1.3 Netstat (extensión para MPTCP).

Para instalar esta herramienta se ha tenido que descargar y compilar el código fuente desde GitHub.

1. Descargar los archivos del código fuente

```
sudo git clone https://github.com/multipath-tcp/net-tools
```

2. Situarse en la carpeta creada y realizar la configuración para poder compilar

```
sudo cd net-tools
```

```
sudo ./configure
```

3. Pasamos a la compilar de los ficheros

```
sudo make
```

4. Por último, instalar la herramienta

```
sudo make install
```

9.1.4 Iperf3

1. Descargar el paquete de https://iperf.fr/iperf-download.php (.deb)

Seleccionar la primera opción de descarga en Ubuntu de 64 bits, son necesarios dos archivos.

2. Situarnos en la carpeta de descargas e instalar la herramienta

```
cd /home/jose/Descargas
```

```
sudo dpkg -i iperf3_3.1.3-1_amd64.deb
```

```
sudo dpkg -i libiperf0_3.1.3-1_amd64.deb
```

9.2 Instalación del servidor FTP y configuración

1. Instalación de la herramienta vsftpd

```
sudo apt-get -y install vsftpd
```

2. Realizar una copia de seguridad del archivo de configuración

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.original
```

3. Crear el directorio de usuarios

```
sudo adduser pruebaftp
```

4. Crear la carpeta para compartir los archivos

```
sudo mkdir /home/pruebaftp/ftp
```

5. Establecer la propiedad

```
sudo chown nobody:nogroup /home/alex/ftp
```

6. Crear el directorio contenedor de archivos y asignarle la propiedad

```
sudo mkdir /home/alex/ftp/files
```

```
sudo chown alex:alex /home/alex/ftp/files
```

7. Configuración de vsftpd

```
sudo nano /etc/vsftpd.conf
```

8. Descomentar las siguientes líneas para habilitar la escritura del usuario que se conecta y para garantizar que accede solamente a los archivos dentro del directorio permitido

```
write_enable=YES

chroot_local_user=YES
```

9. Añadir las dos primeras líneas para que la configuración funcione con el usuario configurado y cualquier otro que se introduzca posteriormente

```
user_sub_token=$USER

local_root=/home/$USER/ftp
```

10. Creación y adición del usuario creado al archivo

```
echo "pruebaftp" | sudo tee -a /etc/vsftpd.userlist
```

11. Reiniciar el servicio

```
sudo systemctl restart vsftpd
```