

TÉCNICAS DE RECONOCIMIENTO DE PAQUETES PARA GESTIÓN MEJORADA DE TRÁFICO DE RED

5 CAMPO TÉCNICO

La presente solicitud se refiere en general a gestión de tráfico de paquetes en una red de comunicación y, más en particular, se refiere a técnicas de reconocimiento de paquetes que implican un servidor proxy.

10 ANTECEDENTES

Conexiones Rápidas de Internet de Protocolo de Datagramas de Usuario (UDP) (QUIC) es un protocolo de transporte seguro basado en UDP, multiplexado por flujos, con una cabecera protegida de integridad y carga útil cifrada. A diferencia de la pila de protocolos de transporte tradicional que utiliza TCP (Protocolo de Control de Transmisión) que reside en el núcleo del sistema operativo, QUIC se puede implementar fácilmente en la capa de aplicación. Esto aporta mucha flexibilidad en términos de evolución, control de congestión e implementación de nuevas características. QUIC está trabajando en el Grupo de Trabajo de Ingeniería de Internet (IETF).

QUIC es un candidato probable para convertirse en el principal protocolo de transporte en el plano de usuario. Se espera que la mayoría de las aplicaciones web (que hoy en día se ejecutan principalmente a través de HTTP/HTTPS) migren a QUIC.

Una distinción importante entre QUIC y TCP es que QUIC es un protocolo cifrado. Los protocolos que se ejecutan en la parte superior de TCP están comúnmente protegidos por el protocolo de Seguridad de Capa de Transporte (TLS). Aunque TLS proporciona protección de la carga útil, las cabeceras TCP permanecen expuestas. QUIC, por otro lado, proporciona protección tanto de la carga útil como de las cabeceras de paquetes.

Un paquete QUIC se protege usando dos claves separadas, una para proteger la cabecera de paquete y otra para proteger la carga útil de paquete. Además, una cabecera de paquete QUIC consta de menos campos que la contraparte TCP. En cambio, QUIC hace uso de tramas que forman parte de la carga útil de paquetes protegidos para información relacionada con transporte tal como reconocimientos.

QUIC permite la migración de una conexión entre diferentes direcciones IP y puertos. Esta migración puede ocurrir implícitamente (por ejemplo, debido a la reenlace del estado de Traducción de Direcciones de Red (NAT)) o explícitamente (por ejemplo, porque el cliente decide usar una red de acceso separada a medida que está disponible).

El control de congestión es un componente importante de protocolos de transporte confiables tal como TCP o QUIC. Un objetivo del control de congestión es evitar que la red se congestione y, si se produce la congestión, evitar que la red permanezca congestionada, en

tanto que se utiliza el ancho de banda disponible. Los algoritmos de control de congestión generalmente se implementan solo en el emisor y no requieren ningún soporte de protocolo específico. Aunque hay algoritmos que requieren soporte tanto del emisor como del receptor, estos algoritmos no se implementan ampliamente en Internet. Muchos algoritmos de control de congestión se diferencian por el tipo de señales que utilizan para inferir la congestión, cómo responden a la congestión y cómo sondean el ancho de banda.

Los algoritmos clásicos de control de congestión se basan en pérdida. Es decir, estos algoritmos utilizan la pérdida de paquetes como un indicador de congestión, típicamente implementando un modelo de Disminución Multiplicable de Incremento Aditivo (AIMD) como una respuesta de congestión. Una respuesta de control de congestión basada en AIMD incrementa la Ventana de Congestión (CWIN) linealmente hasta que se detecta un evento de congestión, momento en el cual reduce el CWIN por un factor multiplicativo. El algoritmo AIMD más común es Reno (Nuevo). Otro algoritmo común de control de congestión basado en pérdida es CUBIC, que incrementa la CWIN como una función cúbica con un punto de inflexión alrededor del tamaño de ventana del evento de pérdida más reciente.

Otros algoritmos utilizan mediciones de retardo y/o velocidad de bits conjuntamente como un indicador de congestión a fin de controlar la velocidad de envío y la respuesta de congestión. Estos algoritmos tienden a ser más proactivos en la mitigación de congestión de red, ya que las mediciones de retardo generalmente están más comúnmente disponibles que la pérdida de paquetes, y también funcionan mejor en redes donde la pérdida de paquetes ocurre por razones distintas de la congestión (por ejemplo, debido a la degradación de canal en entornos inalámbricos).

Un proxy es un programa intermediario o dispositivo de red que maneja el tráfico de red en nombre de otro (por ejemplo, uno o más extremos de comunicación, tal como un dispositivo de envío y/o un dispositivo de recepción). En la comunicación entre los extremos de cliente y servidor, un proxy puede actuar como un servidor para el puntos terminal de cliente y/o un cliente para el puntos terminal de servidor, por ejemplo, con el fin de realizar solicitudes en nombre de uno o ambos extremos. Las solicitudes se pueden atender internamente o pasándolas, posiblemente con traducción, a otros servidores. Además, se puede usar un proxy para proporcionar control de congestión y/u otros servicios beneficiosos.

Un "proxy transparente" es un proxy que no modifica la solicitud o respuesta más allá de lo que se requiere para autenticación e identificación de proxy.

Un "proxy no transparente" es un proxy que modifica la solicitud o respuesta para proporcionar algún servicio añadido al agente de usuario, tal como servicios de anotación de grupo, transformación de tipo de medios, reducción de protocolo y/o filtrado de anonimato.

Un "proxy inverso" básicamente es un proxy que pretende ser el servidor real (en lo que respecta a cualquier cliente o proxy de cliente), pero pasa la solicitud al servidor real que generalmente está detrás de una capa de cortafuegos.

Un "Proxy de Mejora de Rendimiento (PEP)" se utiliza para mejorar el rendimiento de los protocolos en rutas de red donde el rendimiento nativo sufre debido a las características de un enlace o subred en la ruta.

En vista de lo anterior, es evidente que un proxy puede desempeñar un papel importante en la red, por ejemplo, al mejorar el control de congestión entre los extremos de comunicación. Aunque muchas soluciones para implementar estos proxys se comprenden bien cuando la comunicación se basa en TCP, estos proxys pueden ser incapaces de proporcionar los mismos o similares beneficios cuando la comunicación se basa en QUIC, dadas las diferencias entre los protocolos TCP y QUIC. En consecuencia, pueden ser necesarias nuevas técnicas de comunicación en red que impliquen QUIC.

BREVE DESCRIPCIÓN

Las modalidades de la presente divulgación se refieren en general a reconocer un paquete en función de una promesa de entregar el paquete antes de que el paquete se entregue realmente. Estas modalidades pueden permitir que se realice el control de congestión para comunicación entre los extremos usando una variedad de protocolos diferentes. En particular, un proxy puede almacenar de forma intermedia un paquete recibido y enviar una trama de promesa a un dispositivo de recepción que promete entregar ese paquete más tarde, y recibir un reconocimiento del paquete del dispositivo de recepción en respuesta, es decir, antes de haber recibido realmente el paquete. Este reconocimiento se puede reenviar al emisor del paquete, que puede, por ejemplo, continuar enviando paquetes como si el paquete se hubiera entregado, incluso si el proxy retiene el paquete para fines de control de congestión u otras razones relacionadas con gestión de red.

Por consiguiente, las modalidades de la presente divulgación incluyen un método, implementado por un dispositivo de recepción. El método comprende recibir, de un servidor proxy, una trama de promesa que indica que el servidor proxy entregará un paquete al dispositivo de recepción desde un dispositivo de envío posteriormente. El método comprende además enviar un reconocimiento del paquete al dispositivo de envío a través del servidor proxy. El método comprende además recibir el paquete del servidor proxy después de haber enviado el reconocimiento del paquete al dispositivo de envío a través del servidor proxy.

En algunas modalidades, el método comprende además recibir, desde el servidor proxy, un primer paquete de encapsulación que comprende un paquete adicional enviado por el dispositivo de envío y la trama de promesa. Recibir la trama de promesa del servidor proxy comprende recibir la trama de promesa dentro del primer paquete de encapsulación. En algunas de estas modalidades, enviar el reconocimiento del paquete comprende enviar el reconocimiento del paquete en un segundo paquete de encapsulación, el segundo paquete de encapsulación comprende además un reconocimiento del primer paquete de encapsulación, y el reconocimiento del paquete también reconoce el paquete adicional.

En algunas modalidades, el método comprende además recibir un número de paquete del paquete en la trama de promesa. El número de paquete indica que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente. En algunas de estas modalidades, el método comprende además recibir, en la trama de promesa, una indicación de uno o más números de paquete adicionales correspondientes a paquetes adicionales recibidos por el servidor proxy que el servidor proxy promete entregar al dispositivo de recepción posteriormente. La indicación del uno o más números de paquete adicionales comprende un número que especifica cuántos paquetes contiguos inmediatamente precedentes al paquete están comprendidos en los paquetes adicionales y se promete que se entregarán, de acuerdo con la trama de promesa, al dispositivo de recepción posteriormente. En algunas de estas modalidades, el método comprende además recibir, en la trama de promesa, una indicación adicional de uno o más números de paquete adicionales correspondientes a otros paquetes no recibidos por el servidor proxy del dispositivo de envío e indicar que, de acuerdo con la trama de promesa, el servidor proxy no promete la entrega de los otros paquetes al dispositivo de recepción. En algunas de estas modalidades, los otros paquetes son contiguos e inmediatamente preceden a los paquetes contiguos inmediatamente precedentes al paquete, y la indicación del uno o más números de paquete adicionales comprende además un número adicional que especifica cuántos paquetes contiguos inmediatamente precedentes a los otros paquetes están comprendidos en los paquetes adicionales y se promete que se entregarán, de acuerdo con la trama de promesa, por el servidor proxy al dispositivo de recepción (600) posteriormente.

En algunas modalidades, la trama de promesa comprende un campo de recuento de intervalo que indica cuántos intervalos de paquetes se indican por la trama de promesa.

En algunas modalidades, el método comprende además recibir, en la trama de promesa, una parte cifrada del paquete que codifica un número de paquete del paquete para indicar que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente. El método comprende además descifrar la parte cifrada del paquete comprendido en la trama de promesa para identificar el paquete que el servidor proxy entregará al dispositivo de recepción posteriormente. En algunas de estas modalidades, el método comprende además recibir, en la trama de promesa, un recuento de paquetes que indica cuántos paquetes se promete, de acuerdo con la trama de promesa, que se entregarán al dispositivo de recepción posteriormente.

Otras modalidades se refieren a un método, implementado por un servidor proxy. El método comprende recibir, de un dispositivo de envío, un paquete destinado a un dispositivo de recepción. El método comprende además enviar, al dispositivo de recepción, una trama de promesa que indica que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente. El método comprende además reenviar un reconocimiento del paquete del dispositivo de recepción al dispositivo de envío. El método comprende además entregar el

paquete al dispositivo de recepción después de haber reenviado el reconocimiento al dispositivo de envío.

En algunas modalidades, el método comprende además recibir un paquete adicional destinado al dispositivo de recepción y enviar un primer paquete de encapsulación que comprende el paquete adicional y la trama de promesa al dispositivo de recepción. Enviar la trama de promesa al dispositivo de recepción comprende enviar la trama de promesa dentro del primer paquete de encapsulación. En algunas de estas modalidades, reenviar el reconocimiento del paquete comprende recibir, del dispositivo de recepción, un segundo paquete de encapsulación que comprende el reconocimiento del paquete y una porción restante. La porción restante comprende un reconocimiento del primer paquete de encapsulación. Reenviar el reconocimiento del paquete comprende además enviar el reconocimiento del paquete al dispositivo de envío sin la porción restante. El reconocimiento del paquete también reconoce el paquete adicional.

En algunas modalidades, el método comprende además incluir un número de paquete del paquete en la trama de promesa para indicar que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente. En algunas de estas modalidades, el método comprende además incluir, en la trama de promesa, una indicación de uno o más números de paquete adicionales correspondientes a paquetes adicionales recibidos del dispositivo de envío para indicar que el servidor proxy entregará los paquetes adicionales al dispositivo de recepción posteriormente. La indicación del uno o más números de paquete adicionales comprende un número que especifica cuántos paquetes contiguos inmediatamente precedentes al paquete están comprendidos en los paquetes adicionales y se promete que se entregarán, de acuerdo con la trama de promesa, al dispositivo de recepción posteriormente. En algunas de estas modalidades, el método comprende además incluir, en la trama de promesa, una indicación adicional de uno o más números de paquete adicionales correspondientes a otros paquetes no recibidos del dispositivo de envío para indicar que, de acuerdo con la trama de promesa, el servidor proxy no promete la entrega de los otros paquetes al dispositivo de recepción. En algunas de estas modalidades, los otros paquetes son contiguos e inmediatamente preceden a los paquetes contiguos inmediatamente precedentes al paquete. La indicación del uno o más números de paquete adicionales comprende además un número adicional que especifica cuántos paquetes contiguos inmediatamente precedentes a los otros paquetes están comprendidos en los paquetes adicionales y se promete que se entregarán, de acuerdo con la trama de promesa, al dispositivo de recepción posteriormente.

En algunas modalidades, el método comprende además que la trama de promesa comprende un campo de recuento de intervalo que indica cuántos intervalos de paquetes se indican por la trama de promesa.

En algunas modalidades, el método comprende además incluir, en la trama de promesa, una parte cifrada del paquete que codifica un número de paquete del paquete para indicar que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente.

En algunas de estas modalidades, el método comprende además incluir, en la trama de promesa, un recuento de paquetes que indica cuántos paquetes se promete, de acuerdo con la trama de promesa, que se entregarán al dispositivo de recepción posteriormente.

Otras modalidades incluyen dispositivos correspondientes. Una o más de estas modalidades incluyen un dispositivo de recepción configurado para recibir, de un servidor proxy, una trama de promesa que indica que el servidor proxy entregará un paquete al dispositivo de recepción desde un dispositivo de envío posteriormente y enviará un reconocimiento del paquete al dispositivo de envío a través del servidor proxy. El dispositivo de recepción se configura además para recibir el paquete del servidor proxy después de haber enviado el reconocimiento del paquete al dispositivo de envío a través del servidor proxy.

En algunas modalidades, el dispositivo de recepción se configura además para realizar cualquiera de los métodos descritos anteriormente con respecto a un dispositivo receptor.

En algunas modalidades, el dispositivo de recepción comprende circuitería de procesamiento y circuitería de interfaz acoplada comunicativamente a la circuitería de procesamiento, la circuitería de procesamiento que se configura para realizar cualquiera de los métodos descritos anteriormente con respecto al dispositivo de recepción.

Otras modalidades incluyen un servidor proxy configurado para recibir, de un dispositivo de envío, un paquete destinado a un dispositivo de recepción y enviar, al dispositivo de recepción, una trama de promesa que indica que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente. El servidor proxy se configura además para reenviar un reconocimiento del paquete del dispositivo de recepción al dispositivo de envío y entregar el paquete al dispositivo de recepción después de haber reenviado el reconocimiento al dispositivo de envío.

En algunas modalidades, el servidor proxy se configura además para realizar cualquiera de los métodos descritos anteriormente con respecto a un servidor proxy.

En algunas modalidades, el servidor proxy comprende circuitería de procesamiento y circuitería de interfaz acoplada comunicativamente a la circuitería de procesamiento, la circuitería de procesamiento que se configura para realizar cualquiera de los métodos descritos anteriormente con respecto al servidor proxy.

Otras modalidades incluyen sistemas correspondientes, programas de computadora y portadores.

Aunque se han descrito modalidades particulares más adelante, se debe señalar que algunas de estas modalidades pueden, en algunos casos, incluir una o más características descritas más adelante, como se explicará adicionalmente a continuación.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Aspectos de la presente divulgación se ilustran a manera de ejemplo y no se limitan por las figuras adjuntas con referencias similares que indican elementos similares. En general, se debe considerar que el uso de un número de referencia se refiere a la materia objeto ilustrada de acuerdo con una o más modalidades, mientras que el análisis de una instancia específica de un elemento ilustrado anexará una designación de letra a la mismas (por ejemplo, el análisis de una conexión 140, generalmente, a diferencia del análisis de instancias particulares de conexiones 140a, 140b).

La figura 1 es un diagrama de bloques esquemático que ilustra un ejemplo de comunicación TCP entre un cliente, servidor y proxy, de acuerdo con una o más modalidades de la presente divulgación.

La figura 2 es un diagrama de señalización que ilustra un ejemplo de mensajes intercambiados entre cliente, servidor y proxy, de acuerdo con una o más modalidades de la presente divulgación.

La figura 3 es un diagrama de bloques esquemático que ilustra un ejemplo de comunicación QUIC entre un cliente, servidor y proxy, de acuerdo con una o más modalidades de la presente divulgación.

La figura 4 y la figura 5 son diagramas de señalización que ilustran ejemplos de mensajes intercambiados entre cliente, servidor y proxy, de acuerdo con una o más modalidades de la presente divulgación.

Las figuras 6-8 son diagramas de bloques esquemáticos que ilustran estructuras de trama de ejemplo, de acuerdo con modalidades de la presente divulgación.

Las figuras 9 y 10 son diagramas de flujo que ilustran métodos de ejemplo, de acuerdo con modalidades de la presente divulgación.

La figura 11 es un diagrama de bloques esquemático que ilustra un dispositivo de recepción de ejemplo, de acuerdo con una o más modalidades de la presente divulgación.

La figura 12 es un diagrama de bloques esquemático que ilustra un servidor proxy de ejemplo, de acuerdo con una o más modalidades de la presente divulgación.

DESCRIPCIÓN DETALLADA

Los PEP basados en TCP generalmente se implementan en redes celulares con el propósito de acelerar el tráfico y realizar un control de congestión que se adapte para funcionar bien con la dinámica de redes de acceso tal como las Redes de Acceso de Radio (RAN). Tradicionalmente funcionan dividiendo las conexiones TCP de extremo a extremo en dos conexiones separadas, una entre el servidor y el proxy y la otra entre el proxy y el cliente. De esta manera, se establecen dos bucles de control separados, y el proxy puede regular cuántos datos se reconocen y/o se necesitan almacenar de forma intermedia para realizar un control de congestión adecuado en cada uno de los bucles de control. Dado que las cabeceras TCP y la

carga útil generalmente no están ofuscadas y la seguridad (por ejemplo, protección de integridad) no está integrada en el protocolo, un proxy TCP generalmente no requiere ninguna configuración o consentimiento explícito de ninguno de los extremos de comunicación.

La figura 1 ilustra un ejemplo de un proxy TCP tradicional 110 que divide lo que de otro modo sería una conexión individual en dos conexiones separadas 140a, 140b, cada una de las cuales tiene su propio bucle de control de congestión. El proxy TCP 110 reenvía datos de capa de aplicación de una conexión 140b a la otra 140a (y en algunas modalidades, viceversa). Los extremos de comunicación (es decir, el cliente TCP 120 y el servidor TCP 130, en este ejemplo) no requieren conocimiento explícito del proxy TCP 110.

La figura 2 ilustra un ejemplo de mensajería entre el proxy TCP 110, el cliente TCP 120 y el servidor TCP 130. Como se muestra en la figura 2, el proxy TCP 110 puede transmitir reconocimientos ACK2, ACK4 para paquetes P1-P4 recibidos del servidor, y puede recibir reconocimientos ACK2, ACK4 para paquetes P1-P4 reenviados al cliente. Aunque no se muestra, el proxy TCP 110 puede transmitir adicional o alternativamente reconocimientos para paquetes recibidos del cliente TCP 120, y puede recibir reconocimientos para paquetes enviados al servidor TCP 130. Además, el proxy TCP 110 puede regular la velocidad de transmisión de paquetes que recibe y puede asegurar que tenga suficientes datos almacenados de forma intermedia para realizar el control de congestión en el tramo sobre el cual se reenvían los paquetes. Por consiguiente, el control de congestión se puede realizar individualmente para cualquiera o ambas de las conexiones 140a y 140b.

En el ejemplo particular ilustrado en la figura 2, el proxy TCP 110 recibe paquetes del servidor TCP 130 a una velocidad de transmisión particular, y almacena los paquetes recibidos en una memoria intermedia para que el proxy TCP 110 pueda reenviarlos al cliente TCP 120 más lentamente de lo que se recibieron. En otras modalidades (no mostradas), el proxy TCP 110 puede usar una memoria intermedia de paquetes recibidos previamente para reenviar paquetes más rápidamente de lo que se recibieron (es decir, siempre que la memoria intermedia no se haya quedado sin paquetes). Como se analizó anteriormente, en algunas modalidades, el proxy TCP 110 puede realizar funciones similares para paquetes que se originan desde el cliente TCP 120 y están destinados al servidor TCP 130.

Debido a que QUIC viene con seguridad integrada tal como autenticación y cifrado, un proxy QUIC (en contraste con el proxy TCP 110 analizado anteriormente) típicamente requiere configuración explícita y consentimiento de al menos uno de los extremos de comunicación. En una modalidad típica de un proxy de reenvío QUIC, hay múltiples contextos de seguridad en su lugar, es decir, un contexto "interno" (que proporciona protección de extremo a extremo entre el cliente y el servidor) y al menos un contexto "externo" (que protege la comunicación entre un extremo y el proxy).

Aunque una implementación de proxy QUIC común tendrá un contexto de seguridad externa para una conexión 140d establecida entre el cliente QUIC 220 y el proxy QUIC 210,

como se muestra en la figura 3, no existe este contexto de seguridad externa entre el proxy QUIC 220 y el servidor QUIC 230. En cambio, los paquetes QUIC de una conexión de extremo a extremo 140c se tunelizan entre el cliente QUIC 220 y el proxy QUIC 210 a través de la conexión 140d, y utilizan datagramas UDP entre el proxy QUIC 210 y el servidor QUIC 230. En esta implementación, la conexión de extremo a extremo 140c no se divide en dos conexiones separadas 140a, 140b como se describió anteriormente, y puede tener lugar un bucle de control de congestión de extremo a extremo entre el servidor QUIC 230 y el cliente QUIC 220.

No obstante, el contexto de seguridad externa entre el proxy QUIC 210 y el cliente QUIC 220 también se puede controlar la congestión sin que el proxy QUIC 210 tenga la oportunidad de influir en el comportamiento de envío del servidor QUIC 230. En algunas de estas modalidades, esto puede conducir a un rendimiento e interacción subóptimos entre el bucle de control de congestión de extremo a extremo y el bucle de control de proxy a cliente. Debido a que el proxy QUIC 210 encapsula parte de una conexión de extremo a extremo 140c en una conexión QUIC externa 140d que tiene su propio contexto de seguridad externa, el proxy QUIC 210 no tiene la oportunidad de reconocer directamente los datos recibidos del servidor QUIC 230. En cambio, como se muestra en la figura 4, el proxy QUIC 210 espera los reconocimientos del cliente y envía los reconocimientos correspondientes al servidor en respuesta.

La figura 4 ilustra un ejemplo particular de un proxy QUIC 210 que encapsula parte de una conexión QUIC de extremo a extremo 140c en una conexión externa 140d entre un cliente QUIC 220 y el proxy QUIC 210. Una diferencia significativa en este tipo de proxy en comparación con un PEP TCP (por ejemplo, como se analizó anteriormente) es que debido a que la conexión de extremo a extremo 140c no está dividida, el proxy no tiene la oportunidad de reconocer los paquetes recibidos del servidor QUIC 230 directamente, y los dos bucles de control de congestión están en capas entre el cliente QUIC 220 y el proxy QUIC 210, lo que potencialmente conduce a una interacción subóptima.

Se señala que aunque las modalidades particulares de la presente divulgación se dibujarán específicamente en el protocolo QUIC, otras modalidades se pueden aplicar fácilmente a otros protocolos, proxys, extremos y/o dispositivos que utilizan conceptos inventivos similares. Por lo tanto, no se debe considerar esencial que cualquier proxy, extremo o señalización particular analizado más adelante soporte el protocolo QUIC. Dicho esto, las modalidades particulares descritas en la presente se pueden prestar particularmente bien a entornos en los que se utiliza el protocolo QUIC.

En vista de lo anterior, una o más modalidades de la presente divulgación se refieren a un proxy QUIC 210 que notifica a un compañero sobre paquetes que ha recibido de uno o más compañeros de comunicación y promete entregar esos paquetes de forma confiable. Al recibir esta información, el compañero puede reconocer la recepción de estos paquetes hacia el otro compañero de comunicación inmediatamente, lo que permite que el emisor abra la CWIN más

rápidamente. Esto, a su vez, permite que el proxy almacene datos y gestione el bucle de control de congestión, reduciendo así el riesgo de interacción deficiente del bucle de control de congestión, y aun permitiendo el progreso de la transferencia de datos entre los compañeros de comunicación de extremo a extremo.

5 En particular, las modalidades de la presente divulgación incluyen un cliente QUIC 220 que establece una o más conexiones QUIC 140c hacia uno o más servidores QUIC 230 a través de un proxy QUIC 210. Entre el cliente QUIC 220 y el proxy QUIC 210, la una o más conexiones QUIC de extremo a extremo 140c se encapsulan en una única conexión QUIC 140d. Es decir, el cliente QUIC 220 y el proxy QUIC 210 son extremos de la conexión QUIC
10 encapsulante 140d, de modo que los datos transferidos desde el servidor QUIC 230 se encapsulan por el proxy QUIC 210 antes de que se transmitan hacia el cliente QUIC 220. Cuando el proxy QUIC 210 recibe datos del servidor QUIC 230 a una velocidad mayor que la que transmite el proxy QUIC 210 hacia el cliente QUIC 220, el proxy QUIC 210 envía información al cliente QUIC 220 con respecto a los paquetes que ha recibido y promete
15 entregar. En respuesta a la recepción de esta información con respecto a los paquetes recibidos por el proxy QUIC 210 que el proxy QUIC 210 promete entregar, el cliente QUIC 220 envía los reconocimientos QUIC correspondientes en la conexión QUIC interna relevante 140c al servidor QUIC 230.

 Estas modalidades pueden permitir que el proxy QUIC realice la funcionalidad PEP
20 como se soporta por un extremo individual de conexión configurado adecuadamente. En el ejemplo anterior, el proxy QUIC 210 puede aplicar mecanismos de control de congestión adaptados a redes específicas que soportan el cliente QUIC 220 (por ejemplo, RAN, redes satelitales) en tanto que reduce la interacción subóptima con el control de congestión de lado servidor de la capa interna. Además, el uso del proxy QUIC 210 no rompe la confidencialidad
25 de extremo a extremo de la carga útil de paquete QUIC. En cambio, de acuerdo con modalidades, un número de paquete cifrado de la conexión interna 140c es visible para el proxy QUIC 210. El reporte que utiliza este número es suficiente para que el proxy QUIC 210 identifique y prometa entregar los paquetes mantenidos en el proxy QUIC 210, de modo que el cliente QUIC 220 puede enviar reconocimientos al servidor QUIC 230 en soporte del control de
30 congestión. Además, se pueden gestionar múltiples conexiones de extremo a extremo en paralelo 140c dentro de un único bucle de control de congestión entre el proxy QUIC 210 y el cliente QUIC 220, que puede permitir que se mejore el intercambio de recursos entre múltiples conexiones 140.

 En vista de lo anterior, y como se muestra en la figura 5, las modalidades de la presente
35 divulgación incluyen un proxy QUIC 210 que encapsula paquetes que recibe de uno o más servidores QUIC 230 en una capa QUIC externa que se envía hacia un cliente QUIC 220. Si (como se muestra en la figura 5) el proxy QUIC 210 recibe y almacena de forma intermedia paquetes adicionales desde el servidor QUIC 230 para su posterior transmisión al cliente QUIC

210 (por ejemplo, el proxy QUIC aún no es capaz de reenviar estos paquetes adicionales por razones de control de congestión), el proxy QUIC 210 añade una "trama de promesa" que contiene los números de paquete de esos paquetes que aún no se han reenviado. El cliente QUIC 220, al recibir una trama de promesa, reconoce esos paquetes hacia el o los servidores QUIC 230 que los enviaron como si esos paquetes se hubieran realmente recibido por el cliente QUIC 220. Esto permite que los servidores QUIC 230 progresen con la transferencia de datos entre el cliente QUIC 220 y los servidores QUIC 230. Es decir, el servidor QUIC 230 puede continuar transmitiendo paquetes como si se hubieran entregado al cliente QUIC 220, evitando así que el control de congestión realizado en el tramo de cliente ralentice la velocidad de transmisión en el tramo de servidor de la comunicación de extremo a extremo 140c.

En el ejemplo particular de la figura 5, hay cuatro tipos de paquetes, que para propósitos de ilustración se etiquetan como paquetes SC, PC, CP y CS. Los paquetes SC son paquetes enviados por el servidor QUIC 230 y portan datos destinados al cliente QUIC 220. Los paquetes de PC son paquetes enviados por el proxy QUIC 210 al cliente QUIC 220. Estos paquetes PC pueden comprender uno o más paquetes SC y, en algunos casos, incluyen una trama que especifica los números de paquete de paquetes que el proxy QUIC 210 ha recibido, promete entregar y aún no ha transmitido. Los paquetes CP son paquetes enviados por el cliente QUIC 220 al proxy QUIC 210. Los paquetes CP comprenden reconocimientos de paquetes realmente recibidos por el cliente QUIC 220. En algunos casos, estos paquetes CP encapsulan uno o más paquetes CS, que son paquetes enviados por el cliente QUIC 220 y destinados al servidor QUIC 230. Los paquetes CS comprenden reconocimientos de paquetes realmente recibidos por el cliente QUIC 220, y los paquetes que el proxy QUIC 210 prometió entregar.

Ahora, siguiendo el flujo del diagrama que se muestra en la figura 5, los datos se transmiten en paquetes desde el servidor QUIC 230 hacia un cliente QUIC 220 a través de un proxy QUIC 210 (es decir, en los paquetes SC). El proxy QUIC 210 encapsula los paquetes SC en paquetes PC y reenvía esos paquetes PC hacia el cliente QUIC 220. El proxy QUIC 210 incluye una trama (es decir, una trama de promesa) en algunos de los paquetes PC encapsulantes (es decir, en PC1 y PC2, en este ejemplo). Estas tramas adicionales indican los números de paquete de paquetes que el proxy QUIC 210 ha recibido y tiene la intención de entregar al cliente QUIC 220 (es decir, los paquetes SC1-4 recibidos hasta ahora). El cliente QUIC 220 envía, hacia el servidor QUIC 230, paquetes CS que incluyen reconocimientos de los paquetes SC recibidos por el cliente QUIC 220 y los paquetes SC que el proxy QUIC 210 ha prometido entregar. Los paquetes CS se encapsulan por paquetes CP, que reconocen los paquetes SC realmente recibidos (es decir, pero no los paquetes que se prometieron por el proxy QUIC 210 y aún no han llegado). En respuesta a la recepción de paquetes CP que incluyen paquetes CS, el proxy QUIC 210 desencapsula los paquetes CP y reenvía los paquetes CS al servidor QUIC 230.

En vista de lo anterior, el proxy QUIC 210 puede necesitar alguna manera de identificar los paquetes individuales. Como se analizó anteriormente, los números de paquete QUIC son típicamente parte de la cabecera de paquete QUIC, que se cifra usando una clave que es diferente de la que se utiliza para cifrar la carga útil de paquete QUIC. Por lo tanto, la estructura de una trama de promesa puede depender de si el proxy QUIC 210 tiene acceso o no a la clave de protección de cabecera de extremo a extremo. Si el cliente QUIC 220 expone la clave de protección de cabecera de extremo a extremo al proxy QUIC 210, el cliente QUIC 220 puede añadir la clave de protección de cabecera a un paquete enviado hacia el proxy QUIC 210 después de que se haya derivado la clave (por ejemplo, preferentemente añadir esta clave al primer paquete de este tipo). En general, cada conexión de extremo a extremo probablemente tendría su propia clave de protección de cabecera.

Como se analizó anteriormente, un trama de promesa comprende un conjunto de uno o más números de paquete QUIC de extremo a extremo que el proxy QUIC 210 ha recibido del servidor QUIC 230 y promete entregar al cliente QUIC 220. La estructura de la trama de promesa puede depender de si el proxy QUIC 210 tiene acceso o no a la clave de protección de cabecera de extremo a extremo. Aunque las estructuras de ejemplo de las tramas de promesa se analizan más adelante y se ilustran en las figuras 6-8, otros ejemplos pueden tener otras estructuras. Dicho esto, varias modalidades aplican un concepto general similar de cómo se pueden manejar números de paquete cifrados y no cifrados de una o más conexiones QUIC 140. Se debe señalar además que una trama de promesa puede ser parte del protocolo de transporte QUIC básico o se puede enviar como una carga útil QUIC, dependiendo de la modalidad. Además, como se analizó anteriormente, las modalidades que se aplican a protocolos no QUIC pueden tener características y variaciones similares.

En algunas modalidades, el proxy QUIC 210 descifra las cabeceras de paquete de extremo a extremo (por ejemplo, después de haber recibido la clave de protección de cabecera QUIC del cliente) y crea una trama de promesa que identifica uno o más paquetes que el proxy QUIC 210 promete entregar. En algunas modalidades, la trama de promesa comprende uno o más campos para indicar los paquetes prometidos. Por ejemplo, la trama de promesa puede incluir un campo que especifica la cantidad de un paquete particular prometido para entrega, un campo que indica un intervalo de paquetes contiguos que suceden o preceden al paquete particular, y/o uno o más campos que indican uno o más intervalos adicionales de paquetes contiguos, por ejemplo, que se prometen (y/o no se prometen).

La figura 6 ilustra este ejemplo. La trama de promesa 300 comprende un campo recibido más grande 310, un campo de recuento de intervalo 320, un primer campo de intervalo 330 y un campo de intervalos 340. El campo recibido más grande 310 indica el número de paquete más grande recibido por el proxy y que se promete para entrega futura al cliente. El campo de recuento de intervalo 320 indica cuántos intervalos de números de paquete indica la trama de promesa 300 usando el primer campo de intervalo y/o campo de intervalos 330, 340.

Si el proxy ha recibido todos los paquetes hasta el paquete recibido más grande, este campo puede ser cero.

El primer intervalo indica un número de paquetes contiguos que preceden al paquete indicado en el campo recibido más grande 310. Por lo tanto, si el campo recibido más grande 310 se establece al valor entero 999, y el primer campo de intervalo 330 se establece a 6, esto puede indicar que el proxy QUIC 210 ha recibido el número de paquete 999 y sus seis paquetes anteriores (es decir, paquetes 993-999), y promete entregar esos paquetes al cliente QUIC 220 más tarde.

El campo de intervalos 340 indica uno o más intervalos adicionales de paquetes contiguos que alternativamente no se prometen y se prometen. Este campo puede ser útil cuando hay espacios entre paquetes que el proxy QUIC 210 puede prometer al cliente QUIC 220. Por ejemplo, puede haber paquetes que el proxy QUIC 210 aún no ha recibido, o paquetes que se recibieron que contienen errores y no sería apropiado reenviar.

Por lo tanto, si el campo recibido más grande 310 indica el valor 999, el primer campo de intervalo 330 indica el valor 6, y el campo de intervalos 340 indica los valores 4, 6 y 5, entonces se prometen los números de paquete 993-999 (como se analizó en el ejemplo anterior, y como se indica por los campos recibidos más grandes y el primer intervalo 310, 330), 989-992 no se prometen para entrega (como se indica por el primer valor del campo de intervalos 340), se prometen los paquetes 983-988 (como se indica por el segundo valor del campo de intervalos 340), y se prometen los paquetes 978-982 (como se indica por el tercer valor en el campo de intervalos 340). En este ejemplo, el campo de recuento de intervalo 320 se puede establecer a cuatro para indicar el intervalo indicado en el primer campo de intervalo 330, y los tres intervalos indicados en el campo de intervalos 340.

Aunque en algunas modalidades, cada uno del campo recibido más grande 310, campo de recuento de intervalo 320, primer campo de intervalo 330 y campo de intervalos 340 se incluye en la trama de promesa 300, en otras modalidades, se pueden incluir campos adicionales, menos u otros en la trama de promesa 300. Por ejemplo, si solo se promete un paquete, el campo de promesa puede comprender el campo recibido más grande 310, y puede omitir el campo de recuento de intervalo, el primer campo de intervalo y el campo de intervalos 320, 330, 340. En otras modalidades, el campo recibido más grande 310 puede especificar el paquete que se promete y uno o más de los otros campos pueden ser cero.

En otras modalidades, la trama de promesa 300 puede comprender el campo recibido más grande 310 y el primer campo de intervalo 330, y puede omitir el campo de recuento de intervalo y el campo de intervalo (por ejemplo, cuando solo se promete un intervalo).

En otras modalidades, la trama de promesa 300 puede omitir el campo de recuento de intervalo y el primer campo de intervalo 320, 330 y terminar el conjunto de intervalos indicados en el campo de intervalos 340 con un cero cuando no se indican intervalos adicionales.

Cada uno de estos campos puede ser de cualquier tipo apropiado para transmitir sus respectivos significados. Por ejemplo, el campo recibido más grande, el campo de recuento de intervalo y el primer campo de intervalo 310, 320, 330 pueden ser campos enteros de longitud variable, y el campo de intervalos 340 puede incluir uno o más subcampos enteros de longitud variable. Otras modalidades pueden incluir campos de longitud fija o representar otros tipos de datos (por ejemplo, texto).

En algunas modalidades, el proxy no tiene acceso a las cabeceras de los paquetes QUIC recibidos. Por ejemplo, el proxy QUIC 210 puede no tener acceso a la clave de protección de cabecera de extremo a extremo. En al menos algunas de estas modalidades, el proxy QUIC 210 anexa datos de cabecera cifrados a la trama de promesa 300. Para cada paquete sobre el que el proxy QUIC 210 notificará al cliente QUIC 220, el proxy QUIC 210 extrae los datos de paquete sin procesar requeridos para descifrar la cabecera. Los datos necesarios para descifrar el número de paquete consisten en 4 octetos desde el inicio del campo de número de paquete y N octetos de datos de paquete cifrados que siguen la cabecera de paquete. El valor exacto de N depende del algoritmo de cifrado en uso. Por ejemplo, en al menos una versión de la especificación QUIC, ese valor siempre es 16.

Como el número de paquete no forma parte de las invariantes de la cabecera QUIC, es posible que nuevos algoritmos de criptografía (como se especifica en una nueva versión QUIC) cambien el número de octetos necesarios para descifrar el número de paquete. Por consiguiente, en algunas modalidades, el cliente QUIC 220 señala al proxy QUIC 210 donde se encuentran las porciones de la cabecera requeridas para descifrar el número de paquete dentro del paquete. Por ejemplo, el cliente QUIC 220 puede señalar al proxy QUIC 210 un desplazamiento desde el inicio de la cabecera que indica dónde comienzan los octetos de datos de paquete cifrado requeridos, y el valor de N que indica el número de octetos de datos de paquete cifrados requeridos para descifrar la cabecera. Estas modalidades pueden permitir que el proxy QUIC 210 copie las secciones relevantes de la cabecera de paquete en una trama de promesa 300 de modo que el cliente QUIC 220, al recibir esta trama de promesa 300, pueda descifrar estas secciones y determine el número del paquete recibido y prometido para entrega futura por el proxy QUIC 210. Estas modalidades pueden permitir que el proxy QUIC 210 proporcione tramas de promesa 300 al cliente QUIC 220 incluso si ciertos aspectos del esquema de protección de cabecera cambian en versiones futuras de QUIC.

La figura 7 ilustra un ejemplo de una estructura de trama de promesa 300 en la que se cifran los números de paquete. La trama de promesa 300 comprende un campo de recuento de paquetes 350 y un campo de partes de cabecera de paquete protegido 360. En este ejemplo, el campo de recuento de paquetes 350 indica un número entero de paquetes que el proxy QUIC 210 ha observado, por ejemplo, en total o además del paquete con el que se proporciona la trama de promesa 300. En particular, como se muestra en la figura 5, la trama de promesa 300 puede identificar uno o más paquetes distintos del paquete SC encapsulado en el mismo

paquete PC enviado al cliente QUIC 220. En este ejemplo, los números de paquete cifrados y el conjunto asociado de datos requeridos para el descifrado se anexan después del campo de recuento de paquetes 350 (es decir, en el campo de partes de cabecera de paquete protegido 360) para tantos paquetes como se indica en el campo de recuento de paquetes 350, de modo que el cliente QUIC 220 puede usar esas piezas para determinar el uno o más paquetes que se prometen.

La figura 8 ilustra otro ejemplo de una trama de promesa 300 para prometer la entrega de paquetes cuando el proxy QUIC 210 no puede acceder a los números de paquete (por ejemplo, porque están cifrados y el proxy QUIC 210 no tiene la clave de protección de cabecera requerida para descifrar la porción relevante de la cabecera). En este ejemplo, la longitud de los campos necesarios para descifrar un número de paquete es la misma para cada paquete. Por consiguiente, un campo de longitud específica no se codifica en esta trama de promesa de ejemplo 300. En cambio, esta trama de promesa 300 incluye, para cada paquete prometido, un campo de parte de cabecera protegido 370 y un campo de parte de carga útil protegido muestreado 380. En este ejemplo, el campo de parte de cabecera protegido 370 y el campo de parte de carga útil protegido muestreado 380 tienen longitudes de 32 bits y 128 bits, respectivamente. Otros ejemplos pueden incluir un campo de parte de cabecera protegido individual 370 y uno o más campos de parte de carga útil protegidos muestreados 380. Por ejemplo, si la parte de cabecera protegida es la misma (o se puede derivar) para cada paquete, no es necesario incluir explícitamente la información en el campo de parte de cabecera protegida 370 para cada paquete, ya que se puede confiar en que el cliente QUIC 220 utilice la información correcta para descifrar el número de paquete con base en la información en el campo de parte de cabecera protegido individual proporcionado.

Como se analizó anteriormente, con el fin de explicar varias características y conceptos de modalidades particulares, se analizaron ejemplos en el contexto de un proxy, cliente y servidor que soportan el protocolo QUIC. Otras modalidades pueden utilizar otros protocolos. De hecho, otros protocolos que cifran u ofuscan de otro modo las identidades de paquetes (por ejemplo, debido a ciertas características de seguridad integradas) pueden ser adecuados para aplicar los mismos conceptos o conceptos similares.

En vista de todo lo anterior, y como se muestra en la figura 9, las modalidades de la presente divulgación incluyen un método 800 implementado por un dispositivo de recepción (por ejemplo, una conexión de extremo a un QUIC). El método 800 comprende recibir, de un servidor proxy, una trama de promesa 300 que indica que el servidor proxy entregará un paquete al dispositivo de recepción desde un dispositivo de envío posteriormente (bloque 810). El método 800 comprende además enviar un reconocimiento del paquete al dispositivo de envío a través del servidor proxy (bloque 820) y recibir el paquete del servidor proxy después de haber enviado el reconocimiento del paquete al dispositivo de envío a través del servidor proxy (bloque 830).

Otras modalidades incluyen un método 850 implementado por un servidor proxy, como se muestra en la figura 10. El método 850 comprende recibir, de un dispositivo de envío, un paquete destinado a un dispositivo de recepción (bloque 860). El método 850 comprende además enviar, al dispositivo de recepción, una trama de promesa 300 que indica que el servidor proxy entregará el paquete al dispositivo de recepción posteriormente (bloque 870). El método 850 comprende además reenviar un reconocimiento del paquete del dispositivo de recepción al dispositivo de envío (bloque 880) y entregar el paquete al dispositivo de recepción después de haber reenviado el reconocimiento al dispositivo de envío (bloque 890).

Además, como se muestra en la figura 11, otras modalidades incluyen un dispositivo de recepción 600. El dispositivo de recepción 600 de la figura 11 comprende circuitería de procesamiento 610 y circuitería de interfaz 630. La circuitería de procesamiento 610 se acopla comunicativamente a la circuitería de interfaz 630, por ejemplo, a través de uno o más buses. En algunas modalidades, el nodo de red 600 comprende además circuitería de memoria 620 que se acopla comunicativamente a la circuitería de procesamiento 610, por ejemplo, a través de uno o más buses. De acuerdo con modalidades particulares, la circuitería de procesamiento 610 se configura para realizar uno o más de los métodos descritos en la presente (por ejemplo, el método 800 ilustrado en la figura 9). De manera adicional o alternativa, el dispositivo de recepción 600 puede ser una conexión de extremo a un QUIC (tal como un cliente QUIC 220, un servidor QUIC 230 y/o un Equipo de Usuario (UE)), de acuerdo con modalidades.

Además, como se muestra en la figura 12, otras modalidades incluyen un servidor proxy 700 (por ejemplo, un proxy QUIC 210). El servidor proxy 700 de la figura 12 comprende circuitería de procesamiento 710 y circuitería de interfaz 730. La circuitería de procesamiento 710 se acopla comunicativamente a la circuitería de interfaz 730, por ejemplo, a través de uno o más buses. En algunas modalidades, el servidor proxy 700 comprende además circuitería de memoria 720 que se acopla comunicativamente a la circuitería de procesamiento 710, por ejemplo, a través de uno o más buses. De acuerdo con modalidades particulares, la circuitería de procesamiento 710 se configura para realizar uno o más de los métodos descritos en la presente (por ejemplo, el método 850 ilustrado en la figura 10). De manera adicional o alternativa, el servidor proxy 700 puede ser un nodo de red, proxy QUIC 210, nodo de acceso, estación base, puerta de enlace, enrutador, conmutador y/u otro dispositivo de red.

La circuitería de procesamiento 610, 710 de cualquiera de los dispositivos 600, 700 puede comprender uno o más microprocesadores, microcontroladores, circuitos de hardware, circuitos lógicos discretos, registros de hardware, procesadores de señales digitales (DSP), arreglos de compuertas programables en el campo (FPGA), circuitos integrados de aplicación específica (ASIC) o una combinación de los mismos. Por ejemplo, la circuitería de procesamiento 610, 710 puede ser un hardware programable capaz de ejecutar instrucciones de software de un programa de computadora 660, 760 almacenado en la circuitería de memoria 620, 720 por lo cual se configura la circuitería de procesamiento 610, 710. La circuitería de

memoria 620, 720 de las diversas modalidades puede comprender cualquier medio legible por máquina no transitorio conocido en la técnica o que se puede desarrollar, ya sea volátil o no volátil, que incluye pero no se limita a, medios de estado sólido (por ejemplo, SRAM, DRAM, DDRAM, ROM, PROM, EPROM, memoria flash, unidad de estado sólido, etc.), dispositivos de almacenamiento extraíbles (por ejemplo, tarjeta Digital Segura (SD), tarjeta miniSD, tarjeta microSD, tarjeta de memoria, memoria USB, unidad flash USB, cartucho ROM, disco de medios universal), unidad fija (por ejemplo, unidad de disco duro magnético) o similares, en su totalidad o en cualquier combinación.

La circuitería de interfaz 630, 730 puede ser un concentrador de control configurado para controlar las rutas de datos de entrada y salida (I/O) de su dispositivo respectivo 600, 700. Estas rutas de datos E/S pueden incluir rutas de datos para intercambiar señales a través de una red de comunicaciones, rutas de datos para intercambiar señales con un usuario y/o rutas de datos para intercambiar datos internamente entre componentes del nodo de red 600, 700. Por ejemplo, la circuitería de interfaz 630, 730 puede comprender un transceptor configurado para enviar y recibir señales de comunicación a través de una o más de una red celular, red Ethernet o red óptica. La circuitería de interfaz 630, 730 se puede implementar como un componente físico unitario, o como una pluralidad de componentes físicos que se disponen contigua o separadamente, cualquiera de los cuales se puede acoplar comunicativamente a cualquier otro, o se puede comunicar con cualquier otro a través de la circuitería de procesamiento 610, 710. Por ejemplo, la circuitería de interfaz 630, 730 puede comprender circuitería de transmisor 640, 740 configurada para enviar señales de comunicación a través de una red de comunicaciones y circuitería de receptor 650, 750 configurada para recibir señales de comunicación a través de la red de comunicaciones. De manera similar, la circuitería de interfaz 630, 730 puede comprender una pantalla y/o teclado. Otras modalidades pueden incluir otras permutaciones y/o disposiciones de lo anterior y/o sus equivalentes.

De acuerdo con modalidades del dispositivo de recepción 600 ilustrado en la figura 11, la circuitería de procesamiento 610 se configura para recibir, de un servidor proxy 700, una trama de promesa 300 que indica que el servidor proxy 700 entregará un paquete al dispositivo de recepción 600 desde un dispositivo de envío posteriormente y enviará un reconocimiento del paquete al dispositivo de envío a través del servidor proxy 700. La circuitería de procesamiento 610 se configura además para recibir el paquete del servidor proxy 700 después de haber enviado el reconocimiento del paquete al dispositivo de envío a través del servidor proxy 700.

De acuerdo con modalidades del servidor proxy 700 ilustrado en la figura 12, la circuitería de procesamiento 710 se configura para recibir, de un dispositivo de envío, un paquete destinado a un dispositivo de recepción 600 y enviar, al dispositivo de recepción 600, una trama de promesa 300 que indica que el servidor proxy 700 entregará el paquete al dispositivo de recepción 600 posteriormente. La circuitería de procesamiento 710 se configura además para reenviar un reconocimiento del paquete del dispositivo de recepción 600 al

dispositivo de envío y entregar el paquete al dispositivo de recepción 600 después de haber reenviado el reconocimiento al dispositivo de envío.

Otras modalidades de la presente divulgación incluyen programas de computadora correspondientes. En una de estas modalidades, el programa de computadora comprende instrucciones que, cuando se ejecutan en circuitería de procesamiento de un dispositivo de recepción 600, hacen que el dispositivo de recepción 600 lleve a cabo cualquiera de los respectivos procesos descritos anteriormente. En otra modalidad, el programa de computadora comprende instrucciones que, cuando se ejecutan en circuitería de procesamiento de un servidor proxy 700, hacen que el servidor proxy 700 lleve a cabo cualquiera de los respectivos procesos descritos anteriormente. Un programa de computadora en cualquier sentido puede comprender uno o más módulos de código correspondientes a los medios o unidades descritos anteriormente.

Las modalidades incluyen además un producto de programa de computadora que comprende porciones de código de programa para realizar los pasos de cualquiera de las modalidades en la presente cuando el producto de programa de computadora se ejecuta mediante un dispositivo informático. Este producto de programa de computadora se puede almacenar en un medio de grabación legible por computadora.

Cualquier paso, método, característica, función o beneficio adecuados divulgados en la presente se puede realizar a través de una o más unidades funcionales o módulos de uno o más aparatos virtuales. Cada aparato virtual puede comprender un número de estas unidades funcionales. Estas unidades funcionales se pueden implementar mediante circuitería de procesamiento, que puede incluir uno o más microprocesadores o microcontroladores, así como otro hardware digital, que puede incluir procesador de señales digitales (DSP), lógica digital de propósito especial y similares. La circuitería de procesamiento se puede configurar para ejecutar código de programa almacenado en memoria, que puede incluir uno o varios tipos de memoria tal como memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), memoria caché, dispositivos de memoria flash, dispositivos de almacenamiento óptico, etc. El código de programa almacenado en memoria incluye instrucciones de programa para ejecutar uno o más protocolos de telecomunicaciones y/o comunicaciones de datos, así como instrucciones para llevar a cabo una o más de las técnicas descritas en la presente. En algunas implementaciones, la circuitería de procesamiento se puede usar para hacer que la unidad funcional respectiva realice funciones correspondientes de acuerdo con una o más modalidades de la presente divulgación.

Generalmente, todos los términos utilizados en la presente se deben interpretar de acuerdo con su significado ordinario en el campo técnico relevante, a menos que se dé y/o esté implicado claramente un significado diferente a partir del contexto en el que se usa. Todas las referencias al elemento, aparato, componente, medio, paso, etc. Se deben interpretar abiertamente como referencias a al menos una instancia del elemento, aparato, componente,

medio, paso, etc., a menos que se indique explícitamente lo contrario. Los pasos de cualquier método divulgados en la presente no se tienen que realizar en el orden exacto descrito, a menos que se describa explícitamente un paso como siguiente o anterior a otro paso y/o donde está implícito que un paso debe seguir o preceder a otro paso. Cualquier característica de cualquiera de las modalidades divulgadas en la presente se puede aplicar a cualquier otra modalidad, cuando sea apropiado. De manera similar, cualquier ventaja de cualquiera de las modalidades se puede aplicar a cualquier otra modalidad y viceversa. Otros objetivos, características y ventajas de las modalidades anexas serán evidentes a partir de la descripción.

El término unidad puede tener un significado convencional en el campo de la electrónica, dispositivos eléctricos y/o dispositivos electrónicos y puede incluir, por ejemplo, circuitería eléctricos y/o electrónicos, dispositivos, módulos, procesadores, memorias, dispositivos lógicos de estado sólido y/o discretos, programas de computadora o instrucciones para llevar a cabo tareas respectivas, procedimientos, cálculos, salidas y/o funciones de visualización, y así sucesivamente, tal como aquellos descritos en la presente.

Algunas de las modalidades contempladas en la presente se describen más completamente con referencia a las figuras anexas. Otras modalidades, sin embargo, están contenidas dentro del alcance de la materia objeto divulgada en la presente. La materia objeto divulgada no se debe considerar como limitada solo a las modalidades expuestas en la presente; más bien, estas modalidades se proporcionan a modo de ejemplo para transmitir el alcance de la materia objeto a aquellos expertos en la técnica.