

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 638 212**

51 Int. Cl.:

G06F 13/42 (2006.01)

H04J 3/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.05.2014 PCT/US2014/037035**

87 Fecha y número de publicación internacional: **20.11.2014 WO14186179**

96 Fecha de presentación y número de la solicitud europea: **06.05.2014 E 14730016 (4)**

97 Fecha y número de publicación de la concesión europea: **14.06.2017 EP 2997485**

54 Título: **Sincronización de contador de tramas USB basado en tiempo de medios para bus serie wifi**

30 Prioridad:

15.05.2013 US 201361823805 P
05.05.2014 US 201414270124

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
19.10.2017

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

TSFATY, YOSSEF;
WANG, XIAODONG;
RAISSINIA, ALIREZA;
RAVEENDRAN, VIJAYALAKSHMI,
RAJASUNDARAM y
HUANG, XIAOLONG

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 638 212 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sincronización de contador de tramas USB basado en tiempo de medios para bus serie wifi

5 **CAMPO TÉCNICO**

Los presentes modos de realización se refieren en general a sistemas de comunicación, y, específicamente, al envío de paquetes de protocolo de bus serie universal (USB) a través de un canal wifi.

10 **ANTECEDENTES DE LA TÉCNICA RELACIONADA**

Se emplean varias normas de interfaz para conectar ordenadores, periféricos externos y redes para proporcionar conectividad simple a altas velocidades. Por ejemplo, el bus serie universal (USB) es un protocolo de bus serie de alta velocidad comúnmente utilizado para conectar ordenadores como PC y ordenadores portátiles a una amplia variedad de dispositivos periféricos como ratones, teclados, impresoras, unidades flash, altavoces y similares. Más específicamente, el protocolo USB fue diseñado para ofrecer a los usuarios una interfaz mejorada y fácil de usar para conectar una gama increíblemente diversa de periféricos a sus ordenadores con el procedimiento de enchufar y usar.

Más recientemente, se han diseñado unos protocolos de comunicación inalámbrica para permitir que dispositivos de comunicación tales como PC, portátiles, tabletas y teléfonos inteligentes se conecten con otros dispositivos de comunicación sin hilos ni cables físicos. Por ejemplo, las normas IEEE 802.11 definen varios protocolos wifi que permiten la comunicación inalámbrica mediante señales RF. Por ejemplo, un punto de acceso wifi puede proporcionar una red de área local inalámbrica (WLAN) que permite que uno o más dispositivos cliente, tales como estaciones móviles (STA), se comuniquen entre sí y/o se conecten a redes alámbricas externas, tales como Internet.

Por lo tanto, sería deseable que un dispositivo principal (por ejemplo, un ordenador, un ordenador portátil, una tableta, un teléfono inteligente) intercambiara datos USB con una serie de dispositivos cliente (por ejemplo ratones, teclados, impresoras, unidades flash, altavoces), por ejemplo, para obviar la necesidad de cableado entre el dispositivo principal y los dispositivos cliente. Aunque la tecnología Bluetooth® puede permitir comunicaciones inalámbricas entre el dispositivo principal y los dispositivos cliente, las comunicaciones WLAN tienen un alcance mucho mayor que las comunicaciones Bluetooth.

El documento US 7 734 758 B1 divulga un controlador de control principal remoto de bus serie universal (USB) que incluye una conexión a una red local, una pila de protocolos de red y una memoria para almacenar direcciones de red de adaptadores de dispositivos USB y dispositivos USB en la red. Un adaptador de dispositivo USB tiene uno o más puertos USB, una conexión de red, una dirección de red y una pila de protocolos de red. En su conjunto, el controlador de control principal remoto USB y el adaptador de dispositivo USB permiten la interacción entre un ordenador de red y unos dispositivos USB periféricos de la red.

El documento US 2010/086093 A1 divulga un procedimiento y un aparato para regenerar un reloj local dentro de un módulo inalámbrico y sincronizar el reloj local con un reloj principal inalámbrico. Para un modo de realización, el módulo inalámbrico genera un reloj local, cuenta los ciclos del reloj durante un periodo de referencia de temporización común mantenido de forma inalámbrica entre el módulo inalámbrico y el principal, recibe un recuento del reloj principal durante el mismo período de referencia de temporización común y ajusta la señal de reloj local basándose en una comparación de los dos recuentos. Para un modo de realización, el módulo inalámbrico recibe además referencias de temporización del principal y, además, recibe paquetes de muestras de audio del principal acompañadas de una marca de tiempo, estando basada la marca de tiempo en la referencia de temporización del principal, y emite la muestra de audio en el momento designado por la marca de tiempo.

50 **RESUMEN**

Este sumario se proporciona para introducir de forma simplificada una selección de conceptos que se describen en mayor detalle a continuación en la descripción detallada. Este sumario no pretende indicar características clave ni características esenciales del objeto reivindicado, ni tampoco pretende limitar el alcance del objeto reivindicado.

A continuación se divulga un procedimiento de transmisión de tramas de bus serie universal (USB) a través de una red. Un dispositivo USB recibe una o más tramas USB desde un dispositivo principal a través de la red, en el que la una o más tramas USB se encapsulan en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con la red. El dispositivo USB sincroniza además una señal de reloj local con una señal de reloj del dispositivo principal mediante un mecanismo de sincronización de reloj del protocolo de comunicaciones. Por ejemplo, el protocolo de comunicaciones puede corresponder a un protocolo inalámbrico IEEE 802.11. Por lo tanto, el dispositivo USB puede ajustar su señal de reloj local basándose en una función de sincronización de temporización (TSF) recibida desde el dispositivo principal (por ejemplo, en una o más tramas de baliza). El dispositivo USB determina a continuación una serie de tramas USB transmitidas por el dispositivo principal y procesa la una o más tramas USB basándose, al menos en parte, en la señal de reloj local sincronizada.

El dispositivo USB puede incluir una memoria intermedia de datos para almacenar temporalmente la una o más tramas USB recibidas desde el dispositivo principal. El dispositivo USB puede actualizar además un contador de tramas local basándose, en parte, en una serie de las tramas USB almacenadas en la memoria intermedia de datos.

5 En consecuencia, el dispositivo USB puede determinar el número de tramas USB transmitidas por el dispositivo principal basándose en el contador de tramas local. A continuación, el dispositivo USB puede empezar a procesar las tramas USB almacenadas en la memoria intermedia de datos cuando el contador de tramas local alcanza un primer umbral.

10 Para algunos modos de realización, el dispositivo USB puede recibir un valor de recuento de tramas, desde el dispositivo principal, que indica un número de tramas USB transmitidas por el dispositivo principal. El dispositivo USB también puede recibir un valor de tiempo de medios, en asociación con el valor de recuento de tramas, que indica una hora a la que se registra el valor de recuento de tramas. Por ejemplo, el valor de recuento de tramas y el valor de tiempo de medios pueden recibirse juntos, en al menos una de las tramas USB. El dispositivo USB puede sincronizar además su contador de tramas local con un contador de tramas USB en el dispositivo principal basándose, en parte, en la señal de reloj local, el valor de recuento de tramas y el valor de tiempo de medios.

15 Para otros modos de realización, un dispositivo informático puede encapsular una o más tramas USB en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con una red. El dispositivo informático puede transmitir a continuación uno o más paquetes de datos, que incluyen las tramas USB encapsuladas, a uno o más dispositivos a través de la red. El dispositivo informático puede transmitir además un conjunto de datos de sincronización al uno o más dispositivos para permitir que cada uno del uno o más dispositivos procesen la una o más tramas USB.

25 Los datos de sincronización pueden incluir datos de sincronización de reloj que permiten a cada uno del uno o más dispositivos sincronizar una señal de reloj local respectiva con una señal de reloj del dispositivo informático. Por ejemplo, el protocolo de comunicaciones puede corresponder a un protocolo inalámbrico 802.11. Por consiguiente, el dispositivo informático puede transmitir los datos de sincronización de reloj con una o más tramas de baliza, en el que los datos de sincronización de reloj incluyen un valor TSF.

30 Los datos de sincronización pueden incluir además un valor de recuento de tramas y un valor de tiempo de medios asociado con el valor de recuento de tramas. Por ejemplo, el dispositivo informático puede actualizar un contador de tramas USB basándose en una serie de tramas USB transmitidas a uno o más dispositivos. El valor de recuento de tramas puede indicar un valor actual del contador de tramas USB, mientras que el valor de tiempo de medios puede indicar una hora a la que se registra el valor de recuento de tramas. Para algunos modos de realización, el valor de recuento de tramas y el valor de tiempo de medios se pueden encapsular juntos en al menos una de la una o más tramas USB.

40 Para algunos modos de realización, el dispositivo informático puede transmitir además una petición de procesamiento a uno o más dispositivos. La petición de procesamiento puede incluir una orden para comenzar a procesar la una o más tramas USB. Por ejemplo, la petición de procesamiento puede incluir un umbral de recuento de tramas. El uno o más dispositivos deben comenzar a procesar la una o más tramas USB cuando una serie de tramas USB transmitidas por el dispositivo informático es igual al umbral de recuento de tramas.

45 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

Los presentes modos de realización se ilustran a modo de ejemplo y no se pretende que estén limitados por las figuras de los dibujos adjuntos, en los que:

50 La FIG. 1 es un diagrama de bloques de un dispositivo de red de acuerdo con algunos modos de realización.

La FIG. 2 muestra un sistema de comunicación USB inalámbricas de acuerdo con algunos modos de realización.

55 La FIG. 3 es un diagrama de temporización ilustrativo que representa problemas potenciales de la sincronización inalámbrica de dispositivos cliente USB con un dispositivo principal.

Las FIGS. 4A-4B son diagramas de secuencia que representan un ejemplo de operación de sincronización USB inalámbrica de acuerdo con algunos modos de realización.

60 La FIG. 5 es un diagrama de flujo ilustrativo que representa un ejemplo de procedimiento de funcionamiento de un dispositivo cliente USB de acuerdo con algunos modos de realización.

La FIG. 6 es un diagrama de flujo ilustrativo que representa un ejemplo de procedimiento de funcionamiento de un dispositivo principal USB de acuerdo con algunos modos de realización.

65 La FIG. 7 es un diagrama de flujo ilustrativo que representa una operación de sincronización USB inalámbrica de

acuerdo con algunos modos de realización.

La FIG. 8 es un diagrama de bloques de un dispositivo cliente habilitado para USB de acuerdo con algunos modos de realización.

La FIG. 9 es un diagrama de bloques de un dispositivo principal habilitado para USB de acuerdo con algunos modos de realización.

Los números de referencia similares se refieren a partes correspondientes en todas las figuras de los dibujos.

DESCRIPCIÓN DETALLADA

Los presentes modos de realización se describen a continuación, en el contexto del intercambio de datos USB mediante protocolos de comunicación WLAN, solamente con fines de simplificación. Debe entenderse que los presentes modos de realización son igualmente aplicables para el intercambio de datos USB a través de otras normas inalámbricas adecuadas. En el presente documento, los términos WLAN y wifi pueden incluir comunicaciones reguladas por las normas IEEE 802.11, Bluetooth, HiperLAN (conjunto de normas inalámbricas, comparables a las normas IEEE 802.11, utilizadas principalmente en Europa) y otras tecnologías que tienen un rango de propagación de radio relativamente corto. En el presente documento, el término "dispositivo principal" puede utilizarse para referirse a cualquier dispositivo habilitado para USB capaz de transmitir datos USB (por ejemplo, a través de un canal de comunicaciones inalámbricas) a otros dispositivos habilitados para USB. Además, el término "dispositivo cliente" puede utilizarse en la presente memoria para referirse a cualquier dispositivo habilitado para USB capaz de recibir datos USB (por ejemplo, a través de un canal de comunicaciones inalámbricas) desde un dispositivo principal. Debe tenerse en cuenta que, para algunos modos de realización, un dispositivo principal también puede ser operable como dispositivo cliente, y viceversa.

En la siguiente descripción, se exponen numerosos detalles específicos tales como ejemplos de componentes, circuitos y procedimientos específicos para facilitar una plena comprensión de la presente divulgación. El término "acoplado", en el presente documento, significa conectado directamente a o conectado a través de uno o más componentes o circuitos intermedios. Asimismo, en la siguiente descripción se expone, con fines explicativos, la nomenclatura específica con el objeto de facilitar una plena comprensión de los presentes modos de realización. Sin embargo, como resultará evidente a los expertos en la materia, estos detalles específicos pueden no ser necesarios para llevar a la práctica los presentes modos de realización. En otros casos, unos circuitos y dispositivos ampliamente conocidos se muestran en forma de diagrama de bloques para evitar que la presente divulgación resulte confusa. Cualquiera de las señales proporcionadas a través de varios buses descritos en el presente documento puede someterse a multiplexación en el tiempo con otras señales y proporcionarse a través de uno o más buses comunes. Además, la interconexión entre elementos de circuito o bloques de software puede representarse en forma de buses o líneas de señal únicas. De forma alternativa, cada uno de los buses puede ser una sola línea de señal, y cada una de las líneas de señal individuales puede ser de forma alternativa unos buses, y una sola línea o bus puede representar una cualquiera o más de una pluralidad de mecanismos físicos o lógicos para la comunicación entre componentes. No debe considerarse que los presentes modos de realización se limitan a los ejemplos específicos descritos en el presente documento, sino más bien que sus alcances incluyen todos los modos de realización definidos por las reivindicaciones adjuntas.

Como se ha mencionado anteriormente, sería deseable que un dispositivo principal (por ejemplo, un ordenador, un ordenador portátil, una tableta, un teléfono inteligente, etc.) intercambiara datos USB con uno o más dispositivos cliente (por ejemplo, altavoces, impresoras, teclados, ratones, etc.) a través de una WLAN correspondiente. Al menos uno de los problemas asociados con el intercambio de datos USB con uno o más dispositivos cliente a través de la WLAN es la sincronización de los relojes y/o contadores de los dispositivos cliente con los relojes y/o contadores del dispositivo principal sin la presencia de una conexión alámbrica. Por ejemplo, las normas USB definen protocolos de comunicación a través de un cable USB asociado, y típicamente se basan en el intercambio de información de reloj y contador a través del cable USB para asegurar que los relojes y/o contadores de los dispositivos cliente permanezcan sincronizados con los relojes y/o contadores del dispositivo principal. Por lo tanto, debido a que las comunicaciones WLAN no utilizan una conexión alámbrica (por ejemplo, un cable USB), las técnicas de sincronización de reloj definidas por las normas USB pueden no estar disponibles al intentar transmitir datos USB a través de un canal inalámbrico asociado con una WLAN.

Para la divulgación del presente documento, se puede hacer referencia al modelo de interconexión de sistemas abiertos (OSI), que incluye las 7 capas lógicas siguientes: capa 1, que es la capa física, capa 2, que es la capa de enlace de datos, capa 3, que es la capa de red, capa 4, que es la capa de transporte, capa 5, que es la capa de sesión, capa 6, que es la capa de presentación, y capa 7, que es la capa de aplicación. Cuanto más alta esté en la jerarquía una capa OSI, más cerca estará de un usuario final; cuanto más baja esté en la jerarquía una capa OSI, más cerca estará de un canal físico. Por ejemplo, en la parte superior de la jerarquía del modelo OSI está la capa de aplicación, que interactúa directamente con la aplicación de software del usuario final. En la parte inferior de la jerarquía del modelo OSI está la capa física, que define la relación entre un dispositivo de red y un medio de comunicación físico.

La capa física proporciona especificaciones eléctricas y físicas para el medio físico e incluye transceptores que pueden modular/demodular datos que se van a transmitir/recibir a través del medio. La capa de enlace de datos proporciona los detalles funcionales y/o procedimentales, tales como mecanismos de direccionamiento y control de acceso al canal, para transmisiones de datos entre dispositivos. La capa de enlace de datos incluye dos subcapas: la capa de control de enlace lógico (LLC) y la capa de control de acceso al medio (MAC).

Existe una interfaz entre la capa MAC y la capa PHY para facilitar el intercambio de información entre las dos capas. Esta interfaz se denomina interfaz independiente de medios (MII) porque la capa MAC puede funcionar con todos los medios físicos utilizados para la transmisión de datos (y por tanto puede funcionar independientemente del dispositivo PHY particular empleado). De esta manera, la MII permite que un dispositivo MAC dado se utilice con una amplia variedad de dispositivos PHY. El término MII también puede referirse a un tipo específico de interfaz independiente de medios, además de referirse a todo el género. En el presente documento, los términos "interfaz de acceso a medios" y "MII" se referirán al género completo de dichas interfaces, a menos que se indique lo contrario. Entre los ejemplos de MII se incluyen la interfaz de unidad de acoplamiento (AUI), MII, MII reducida, Gigabit MII (GMII), GMII reducida, GMII serie (SGMII), SGMII Quad (QSGMII), 10GMII y MII serie de fuente sincrónica (S3MII).

La FIG. 1 es un diagrama de bloques de un dispositivo de red 100 de acuerdo con algunos modos de realización. Específicamente, el dispositivo de red 100 puede ser capaz de comunicarse con otro dispositivo (no mostrado para simplificar) a través de un medio M1 (por ejemplo, un canal inalámbrico). El dispositivo de red 100 incluye un procesador 110, una memoria 120, un dispositivo PHY 130 y un dispositivo MAC 140. El dispositivo PHY 130 incluye un transceptor 135 que está acoplado al medio inalámbrico M1. Aunque el transceptor 135 ilustrado en la FIG. 1 está incluido en el dispositivo PHY 130, el transceptor 135 puede ser un dispositivo autónomo o un circuito integrado. La memoria 120 puede ser cualquier elemento o dispositivo de memoria adecuado que incluya, por ejemplo, EEPROM o memoria Flash. El procesador 110 puede ser cualquier procesador adecuado capaz de ejecutar secuencias de mandatos o instrucciones de uno o más programas de software almacenados, por ejemplo, en la memoria 120.

El dispositivo PHY 130 y el dispositivo MAC 140 incluyen, cada uno, una interfaz independiente de medios (MII) 150-1 y 150-2, respectivamente, para transmitir señales entre los dos dispositivos a través de un conjunto de trayectorias de señal 160. Para algunos modos de realización, la MII 150-1 puede incluir un primer y un segundo pares de transistores diferenciales de lado PHY (no mostrados para simplificar) para transmitir y recibir datos a y desde, respectivamente, el dispositivo MAC 140, y la MII 150-2 puede incluir un primer y un segundo pares de transistores diferenciales de lado MAC (no mostrados para simplificar) para transmitir y recibir datos a y desde, respectivamente, el dispositivo PHY 130. Por lo tanto, la MII 150-1 puede denominarse en el presente documento "MII de lado PHY", y la MII 150-2 puede denominarse en el presente documento "MII de lado MAC».

El dispositivo MAC 140 puede ser cualquier dispositivo o circuito integrado que implemente las funciones de una subcapa MAC OSI, y puede ser un dispositivo autónomo o puede estar integrado en el dispositivo de red 100. De manera similar, el dispositivo PHY 130 puede ser cualquier dispositivo o circuito integrado que implemente las funciones de la capa física OSI, y puede ser un dispositivo autónomo o puede estar integrado en el dispositivo de red 100.

Durante las operaciones de transmisión de datos, cuando una aplicación de software de usuario final del dispositivo de red 100 transmite datos a otro dispositivo a través del medio M1, el procesador 110 procesa los datos de acuerdo con las capas superiores del modelo OSI y a continuación transmite los datos, a través del dispositivo MAC 140, al dispositivo PHY 130. A continuación, el dispositivo PHY 130 transmite los datos al medio M1 a través del transceptor 135.

La FIG. 2 representa un sistema de comunicación 200 de acuerdo con algunos modos de realización. El sistema 200 representado incluye un dispositivo principal 210 y tres dispositivos cliente 220(1)-220(3) que pueden comunicarse entre sí mediante un canal de comunicaciones 230. Para algunos modos de realización, el canal de comunicaciones 230 puede ser un canal inalámbrico (por ejemplo, WLAN o wifi) que facilita las comunicaciones de acuerdo con las normas IEEE 802.11, aunque pueden utilizarse otros protocolos de comunicación inalámbrica adecuados. Para otros modos de realización, el canal de comunicaciones 230 puede ser un enlace alámbrico (por ejemplo, un canal Ethernet 10GBASE-T). Además, aunque para simplificar se muestran tres dispositivos cliente 220(1)-220(3) en la FIG. 2, el sistema 200 puede incluir cualquier número de dispositivos cliente 220 conectados al dispositivo principal 210 a través del canal de comunicaciones 230.

El dispositivo principal 210, que puede ser un ordenador, un ordenador portátil, una tableta, un teléfono inteligente u otro dispositivo informático o de comunicación adecuado, incluye un procesador 211, un controlador USB 212, una entidad de gestión de estaciones (SME) 213 de capa de adaptación de protocolo (PAL) y una interfaz WLAN 214. El procesador 211 está acoplado al controlador USB 212, y puede controlar operaciones del controlador USB 212 y/u otros componentes del dispositivo principal 210. El controlador USB 212 puede ser cualquier controlador USB adecuado capaz de facilitar las comunicaciones USB entre el procesador 211 y la PAL SME 213. Para al menos algunos modos de realización, el controlador USB 212 puede incluir o estar asociado con un circuito de subcapa de codificación física (PCS) (no mostrado para simplificar) a fin de proporcionar funciones de codificación para

comunicaciones USB. La PAL SME 213, que puede funcionar de acuerdo con los protocolos definidos por la Wi-Fi Gigabit Alliance (WGA), puede permitir el intercambio de datos USB entre el dispositivo principal 210 y los dispositivos cliente 220(1)-220(3) a través de un canal inalámbrico 230. La interfaz WLAN 214, que está acoplada a la PAL SME 213, puede incluir transceptores para transmitir y recibir datos desde y a los dispositivos cliente 220(1)-220(3) a través del canal inalámbrico 230.

El dispositivo cliente 220 puede ser cualquier dispositivo periférico USB adecuado (por ejemplo, un altavoz, una impresora, un teclado, un ratón, etc.). Aunque para simplificar no se muestra en la FIG. 2, cada uno de los dispositivos cliente 220(1)-220(3) puede incluir un procesador, un controlador USB, una PAL SME y una interfaz WLAN para facilitar el intercambio inalámbrico de datos USB a través del canal inalámbrico 230.

Para que el dispositivo principal 210 haga que los dispositivos cliente 220(1)-220(3) presenten datos USB isócronos en el instante correcto (por ejemplo, simultáneamente entre sí), el contador de tramas USB del controlador USB principal 212 debe sincronizarse con los contadores de tramas USB de los dispositivos cliente 220(1)-220(3). Por ejemplo, si los datos USB contienen información de audio que los dispositivos cliente van a reproducir (por ejemplo, actuando como altavoces), puede ser deseable habilitar la reproducción de la información de audio desde una pluralidad de dispositivos 220(1)-220(3), simultáneamente. En consecuencia, puede ser deseable sincronizar los dispositivos cliente 220(1)-220(3) con el dispositivo principal 210 y/o entre sí.

Para algunos modos de realización, las comunicaciones entre el dispositivo principal 210 y los dispositivos cliente 220(1)-220(3) pueden ser bidireccionales. Por ejemplo, cada uno de los dispositivos cliente 220(1)-220(3) puede actuar como un dispositivo principal, y el dispositivo principal 210 puede actuar como un dispositivo cliente, en ciertos casos. Más específicamente, el dispositivo principal 210 y los dispositivos cliente 220(1)-220(3) pueden ser dispositivos informáticos sustancialmente similares que son capaces de transmitir y recibir tramas USB a través de un canal de comunicaciones inalámbrico (por ejemplo, el canal inalámbrico 230).

La FIG. 3 es un diagrama de temporización ilustrativo 300 que representa problemas potenciales con la sincronización de los dispositivos cliente 220(1)-220(3) con el dispositivo principal 210. Con referencia también a la FIG. 2, cuando el controlador USB 212 recibe datos desde el procesador 211, el controlador USB 212 puede adjuntar una marca de tiempo (T_p) a los datos y encapsular los datos en un paquete para su subsiguiente transmisión a los dispositivos cliente 220(1)-220(3). Sin embargo, puede haber un retardo de transmisión (T_d) entre las capas MAC y PHY (por ejemplo, entre el controlador USB 212 y la interfaz WLAN 214) del dispositivo principal 210. Este retardo de transmisión, que puede no ser constante para paquetes de datos diferentes, puede causar errores de temporización asociados con uno o más de los dispositivos cliente 220(1)-220(3) que reciben, procesan y/o visualizan paquetes de datos transmitidos desde el dispositivo principal 210.

Una solución a este problema de temporización puede implicar analizar el paquete de datos en el dispositivo principal 210, extraer la marca de tiempo original (T_p), calcular el retardo de transmisión (T_d) y finalmente actualizar la marca de tiempo original para reflejar el retardo de transmisión. Este proceso, que puede implicar operaciones de inspección profunda de paquetes (DPI) dentro del dispositivo principal 210, consume valiosos recursos de procesamiento y puede llevar mucho tiempo. Además, muchos dispositivos principales existentes pueden no incluir circuitos o módulos de software capaces de realizar operaciones DPI en paquetes/tramas de datos USB recibidos desde el controlador USB 212.

Para algunos modos de realización, el dispositivo principal 210 y/o los dispositivos cliente 220(1)-220(3) pueden utilizar operaciones WLAN definidas por las normas IEEE 802.11 para sincronizar los dispositivos cliente 220(1)-220(3) con el dispositivo principal 210. Más específicamente, el dispositivo principal 210 puede emplear técnicas incorporadas por las normas IEEE 802.11 para asegurar que los contadores y/o relojes de los dispositivos cliente 220(1)-220(3) se sincronicen (y permanezcan sincronizados) con respectivos contadores y/o relojes del dispositivo principal 210. Por ejemplo, las FIGS. 4A-4B son diagramas de secuencia que representan un ejemplo de operación de sincronización USB inalámbrica de acuerdo con algunos modos de realización.

Con referencia, por ejemplo, a la FIG. 4A, el dispositivo principal 210 genera una señal de reloj interno (Clk_H) que puede utilizarse para medir el tiempo de una transmisión (y/o recepción) de datos a (y/o desde) los dispositivos cliente 220(1)-220(3). De manera similar, cada uno de los dispositivos cliente 220(1)-220(3) puede generar una señal de reloj local correspondiente (Clk_1 - Clk_3) que puede utilizarse para medir el tiempo de una recepción (y/o transmisión) de datos desde (y/o a) el dispositivo principal 210. Para algunos modos de realización, el dispositivo principal 210 puede transmitir datos de sincronización de reloj (Clk_Sync) a cada uno de los dispositivos cliente 220(1)-220(3) con el propósito de sincronizar las señales de reloj locales Clk_1 - Clk_3 con la señal de reloj principal Clk_H . Cada uno de los dispositivos cliente 220(1)-220(3) puede utilizar a continuación los datos Clk_Sync para alinear su reloj local con el reloj principal Clk_H (por ejemplo, aplicando respectivos desplazamientos de temporización a los relojes locales Clk_1 - Clk_3).

Para algunos modos de realización, la operación de sincronización de reloj puede realizarse mediante uno o más mecanismos o técnicas de sincronización de reloj que ya son proporcionados por un protocolo de comunicaciones asociado con el canal de comunicaciones 230. Por ejemplo, la señal de reloj principal Clk_H y las señales de reloj

5 local Clk₁-Clk₃ pueden generarse basándose en una función de sincronización de temporización (TSF) definida por los protocolos inalámbricos IEEE 802.11. De forma alternativa, y/o adicional, la señal de reloj principal Clk_H puede sincronizarse con las señales de reloj locales Clk₁-Clk₃ mediante otras técnicas de sincronización de reloj (por ejemplo, tales como las definidas por las normas 802.1AS, 802.11 v y/o 802.11 mc), dependiendo del nivel de precisión deseado para una aplicación particular.

10 Un temporizador TSF puede ser un contador de módulo 2⁶⁴ que funciona (por ejemplo, varía) en incrementos de microsegundos y, por lo tanto, tiene un valor de recuento máximo de 2⁶⁴ = 102 400 microsegundos (aunque pueden utilizarse otros temporizadores, relojes y/o contadores adecuados). Para algunos modos de realización, el dispositivo principal 210 puede transmitir su valor de temporizador TSF a los dispositivos cliente 220(1)-220(3) en una o más tramas de baliza. Al recibir una trama de baliza, cada uno de los dispositivos cliente 220(1)-220(3) puede establecer su propio temporizador TSF local en el valor del temporizador TSF recibido. Esto asegura que los temporizadores TSF locales de los dispositivos cliente 220(1)-220(3) (por ejemplo, Clk₁-Clk₃) se sincronicen con un temporizador TSF del dispositivo principal 210 (por ejemplo, Clk_H).

15 Para algunos modos de realización, los datos Clk_Sync pueden radiodifundirse y/o transmitirse de forma periódica para mantener la sincronización de reloj entre el dispositivo principal 210 y los dispositivos cliente 220(1)-220(3). Por ejemplo, cuando el canal de comunicaciones 230 es un canal wifi, la especificación 802.11 define un mecanismo de sincronización de reloj en el que el dispositivo principal 210 transmite periódicamente mediante radiodifusión tramas de baliza a los dispositivos cliente 220(1)-220(3). Por consiguiente, los dispositivos cliente 220(1)-220(3) y el dispositivo principal 210 pueden establecer una serie de tiempos de transmisión de baliza objetivo (TBTT) que están separados por un intervalo de baliza (por ejemplo, proporcionado dentro de la trama de baliza). Los dispositivos cliente 220(1)-220(3) pueden, a continuación, resincronizar sus temporizadores TSF locales con el temporizador TSF del dispositivo principal 210 cada vez que se recibe una nueva trama de baliza.

25 Una vez sincronizadas las señales de reloj local Clk₁-Clk₃ con la señal de reloj principal Clk_H, el dispositivo principal 210 puede, a continuación, comenzar a transmitir tramas USB encapsuladas a los dispositivos cliente 220(1)-220(3) a través del canal de comunicaciones 230. Por ejemplo, las tramas USB pueden corresponder a datos USB que de otro modo se transmitirían a uno o más de los dispositivos cliente 220(1)-220(2) a través de un cable USB. En cambio, para algunos modos de realización, las tramas USB pueden encapsularse en (por ejemplo, escribirse y/o transportarse mediante) uno o más paquetes de datos, basándose en un protocolo de comunicaciones inalámbricas, que se van a transmitir a través del canal de comunicaciones 230. Los dispositivos cliente 220(1)-220(3) pueden decapsular los paquetes de datos recibidos para recuperar las tramas USB originales. Para algunos modos de realización, cada uno de los dispositivos cliente 220(1)-220(3) puede actualizar un contador de tramas local (por ejemplo, Cnt₁-Cnt₃) para cada trama USB recibida desde el dispositivo principal 210.

30 Con referencia, por ejemplo, a la FIG. 4B, el dispositivo principal 210 puede transmitir información que indica un valor de recuento de tramas (F_Count) y un tiempo de medios (M_Time) asociado con el valor de recuento a los dispositivos cliente 220(1)-220(3) para sincronizar un contador de tramas Cnt_H del dispositivo principal 210 con los contadores de tramas locales Cnt₁-Cnt₃ de los dispositivos cliente 220(1)-220(3), respectivamente. El contador de tramas Cnt_H puede indicar un número de tramas USB transmitidas por el dispositivo principal 210. Por ejemplo, el dispositivo principal 210 puede actualizar el contador de tramas Cnt_H para cada trama USB transmitida a los dispositivos cliente 220(1)-220(3).

35 El valor F_Count puede indicar el número total de tramas USB transmitidas por el dispositivo principal 210 (por ejemplo, basándose en el contador de tramas principal Cnt_H) en un momento dado. Más específicamente, el valor M_Time puede especificar la hora particular (por ejemplo, basándose en la señal de reloj principal Clk_H) a la que se determinó el valor F_Count y/o se integró en un paquete de datos y/o trama USB salientes. Para algunos modos de realización, el valor M_Time puede corresponder a una marca de tiempo de transmisión (por ejemplo, como la definida por la especificación 802.11) que está integrada en cada paquete de datos y/o trama transmitidos por el dispositivo principal 210. Para algunos modos de realización, los valores F_Count y M_Time pueden codificarse sustancialmente al mismo tiempo y, de este modo, "acoplarse" entre sí. Por ejemplo, los valores F_Count y M_Time pueden encapsularse en una o más tramas USB y proporcionarse con otros datos USB encapsulados. De forma alternativa, los valores F_Count y M_Time pueden transmitirse por separado de los datos USB (por ejemplo, como parte de una trama de gestión independiente).

45 Cada uno de los contadores de tramas locales Cnt₁-Cnt₃ puede sincronizarse con el contador de tramas principal Cnt_H basándose en el valor F_Count, el valor M_Time y una señal de reloj local correspondiente Clk₁-Clk₃, respectivamente. Por ejemplo, puede realizarse una operación de sincronización de recuento de tramas para cualquier dispositivo cliente 220 basándose en la siguiente ecuación:

$$60 \quad Cnt_n = F_Count + (Clk_n - M_Time) * \left(\frac{\text{Tramas USB}}{\text{Ciclo del reloj}} \right) \quad (1)$$

Por lo tanto, después de sincronizar cada una de las señales de reloj local Clk₁-Clk₃ con la señal de reloj principal

- 5 Clk_H (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A), se puede determinar la cantidad de tiempo transcurrido entre la codificación del valor F_Count por el dispositivo principal 210 y su recepción por uno de los dispositivos cliente 220(1)-220(3) obteniendo la hora a la que el valor F_Count se recibe (por ejemplo, basándose en las señales de reloj local Clk₁-Clk₃) y restando el valor M_Time. El número de tramas USB adicionales que es probable que hayan sido transmitidas por el dispositivo principal 210 durante este tiempo transcurrido se puede determinar multiplicando el tiempo transcurrido por la velocidad de transmisión del dispositivo principal 210 (por ejemplo, el número total de tramas USB que pueden transmitirse por ciclo de reloj). El resultado se puede sumar al valor F_Count para determinar el valor de recuento actual del contador principal Cnt_H.
- 10 Los dispositivos cliente 220(1)-220(3) pueden actualizar, a continuación, sus respectivos contadores locales Cnt₁-Cnt₃ basándose en tramas USB recibidas subsiguientemente. Debe tenerse en cuenta, sin embargo, que la velocidad a la que los dispositivos cliente individuales 220(1)-220(3) reciben tramas USB pueden variar. Por ejemplo, debido a retardos en la transmisión de datos por el dispositivo principal 210, retardos de propagación a lo largo del canal de comunicaciones 230 y/o condiciones de trayectorias múltiples, cada uno de los dispositivos cliente 220(1)-220(3) puede recibir tramas USB en tiempos diferentes. De este modo, en cualquier momento dado, el número de tramas USB almacenadas en memoria intermedia por uno de los dispositivos cliente 220(1)-220(3) puede ser diferente del número de tramas USB almacenadas en memoria intermedia por otro dispositivo cliente. Por ejemplo, en un tiempo T particular, el contador principal Cnt_H puede tener un valor de recuento de N, el contador local Cnt₁ puede almacenar un valor de recuento de N, el contador local Cnt₂ puede almacenar un valor de recuento de N-1, y el contador local Cnt₃ puede almacenar un valor de recuento de N-2. Para algunos modos de realización, el dispositivo principal 210 puede transmitir periódicamente nuevos valores F_Count y M_Time a los dispositivos cliente 220(1)-220(3) con el fin de mantener la sincronización entre el contador principal Cnt_H y cada uno de los contadores locales Cnt₁-Cnt₃. Esto permite que cada uno de los contadores de tramas locales Cnt₁-Cnt₃ mantenga un recuento del número total de tramas transmitidas por el dispositivo principal 210 y no solo el número de tramas USB recibidas y/o almacenadas en memoria intermedia por uno respectivo de los dispositivos cliente 220(1)-220(3).

Para algunos modos de realización, el dispositivo principal 210 puede transmitir un mensaje de petición de procesamiento a uno o más del dispositivo cliente 220(1)-220(3) con instrucciones para comenzar a procesar y/o presentar tramas USB recibidas basándose en un recuento de tramas particular. Por ejemplo, cada uno de los dispositivos cliente 220(1)-220(3) puede incluir un elemento de almacenamiento interno que almacena en memoria intermedia tramas USB recibidas desde el dispositivo principal 210. Los dispositivos cliente 220(1)-220(3) pueden continuar almacenando tramas USB en memoria intermedia hasta que el dispositivo principal 210 indique que deben comenzar a procesar o presentar las tramas USB almacenadas en memoria intermedia. Para algunos modos de realización, el mensaje de petición de procesamiento puede ordenar a los dispositivos cliente 220(1)-220(3) que comiencen a procesar las tramas USB después de que un contador de tramas local alcance un umbral de recuento de tramas particular. Sin embargo, como se ha descrito anteriormente, los contadores de tramas locales Cnt₁-Cnt₃ pueden desviarse con respecto al contador principal Cnt_H. Sin embargo, puede ser deseable que cada uno de los dispositivos cliente 220(1)-220(3) comience a procesar o presentar tramas USB almacenadas sustancialmente al mismo tiempo (por ejemplo, como en el caso en el que cada uno de los dispositivos cliente 220(1)-220(3) es un altavoz configurado para reproducir la misma señal de audio). Por lo tanto, la sincronización de cada uno de los contadores de tramas locales Cnt₁-Cnt₃ con el contador principal Cnt_H puede permitir que varios dispositivos cliente 220(1)-220(3) comiencen a procesar tramas USB simultáneamente.

Al aprovechar los protocolos de reloj existentes proporcionados para el canal de comunicación 230, el sistema 200 puede mantener de manera eficiente y precisa un recuento de tramas USB sincronizado entre el dispositivo principal 210 y los dispositivos cliente 220(1)-220(3) sin necesitar una inspección profunda de paquetes. Además, de acuerdo con los presentes modos de realización, cada uno de los dispositivos cliente 220(1)-220(3) puede determinar el recuento de tramas actual del dispositivo principal 210, mediante los valores F_Count y M_Time, aunque los dispositivos cliente 220(1)-220(3) reciban dichos valores en diferentes momentos.

La FIG. 5 es un diagrama de flujo ilustrativo que representa un ejemplo de procedimiento 500 de funcionamiento de un dispositivo cliente USB de acuerdo con algunos modos de realización. En ciertos ejemplos, el procedimiento 500 puede implementarse mediante cualquiera de los dispositivos cliente 220 que se han descrito anteriormente con respecto a las FIGS. 2 y 4A-4B. Específicamente, un dispositivo cliente 220 puede recibir una o más tramas USB desde un dispositivo principal 210 a través de una red (501). Para algunos modos de realización, la una o más tramas USB se pueden encapsular en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con la red. El dispositivo cliente 220 puede decapsular los paquetes de datos recibidos para recuperar la una o más tramas USB. Además, para algunos modos de realización, el dispositivo cliente 220 puede almacenar las tramas USB recibidas en una memoria intermedia de datos.

El dispositivo cliente 220 sincroniza además su señal de reloj local con una señal de reloj del dispositivo principal (502). Específicamente, el dispositivo cliente 220 puede sincronizar su señal de reloj local con la señal de reloj principal basándose en un mecanismo de sincronización de reloj del protocolo de comunicaciones. Por ejemplo, el dispositivo principal 210 puede transmitir periódicamente datos de sincronización (Clk_Sync) de reloj de radiodifusión que el dispositivo cliente 220 puede utilizar para ajustar su señal de reloj local. En una red inalámbrica 802.11, los datos de sincronización de reloj pueden corresponder a un valor TSF radiodifundido por el dispositivo principal 210

en una o más tramas de baliza. El dispositivo cliente 220 puede a continuación ajustar su propio valor TSF (si es necesario) para que coincida con el valor TSF del dispositivo principal 210 (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A).

5 El dispositivo cliente 220 determina a continuación un número de tramas USB transmitidas por el dispositivo principal 210 (503). Específicamente, el dispositivo cliente 220 puede determinar el número de tramas USB transmitidas por el dispositivo principal 210 basándose en la señal de reloj local sincronizada. Por ejemplo, el dispositivo cliente 220 puede actualizar un contador de tramas locales para cada trama USB recibida desde el dispositivo principal 210. Sin embargo, tal como se ha descrito anteriormente, el número de tramas USB recibidas por el dispositivo cliente 220
10 puede no coincidir siempre con el número de tramas USB transmitidas por el dispositivo principal 210 (por ejemplo, debido a paquetes de datos desviados y/o excluidos). Por lo tanto, para algunos modos de realización, el contador de tramas local puede rastrear un contador de tramas USB en el dispositivo principal 210 basándose en datos de sincronización transmitidos por el dispositivo principal 210. Los datos de sincronización pueden incluir, por ejemplo, un valor de recuento de tramas (F_Count) y un valor de tiempo de medios (M_Time). Más específicamente, el valor
15 F_Count puede indicar el número de tramas USB transmitidas por el dispositivo principal 210 en el momento especificado por el valor M_Time. El dispositivo cliente 220 puede utilizar, a continuación, la señal de reloj local sincronizada para determinar el número de tramas USB adicionales transmitidas desde el momento indicado por el valor M_Time (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4B). Para algunos modos de realización, el dispositivo cliente 220 puede resincronizar periódicamente su contador de tramas local con el del dispositivo principal 210.
20

Finalmente, el dispositivo cliente 220 puede procesar las tramas USB recibidas basándose en el número de tramas USB transmitidas por el dispositivo principal 210 (504). Por ejemplo, el dispositivo principal 210 puede ordenar al dispositivo cliente 220 que comience a procesar y/o presentar las tramas USB recibidas una vez que el contador de tramas local alcanza un umbral de recuento de tramas particular. Debido a que el contador de tramas local puede sincronizarse con el contador de tramas principal, el dispositivo cliente 220 puede comenzar a procesar tramas USB en cualquier momento indicado por el dispositivo principal 210. Esto puede permitir que varios dispositivos cliente 220 comiencen a procesar tramas USB sustancialmente al mismo tiempo, independientemente del número de tramas USB recibidas realmente por cada dispositivo cliente 220 en ese momento.
25
30

La FIG. 6 es un diagrama de flujo ilustrativo que representa un ejemplo de procedimiento 600 de funcionamiento de un dispositivo principal USB de acuerdo con algunos modos de realización. En ciertos ejemplos, el dispositivo principal 210, descrito anteriormente con respecto a las FIGS. 2 y 4A-4B, puede implementar el procedimiento 600. Específicamente, un dispositivo principal 210 de una red puede encapsular tramas USB en paquetes de datos basándose en un protocolo de comunicaciones asociado con la red (601). Como se ha descrito anteriormente, las tramas USB pueden corresponder a datos USB que de otro modo se transmitirían al uno o más dispositivos cliente 220 a través de un cable USB. El dispositivo principal 210 puede encapsular las tramas USB, por ejemplo, integrando datos USB correspondientes dentro de uno o más paquetes de datos que se van a transmitir a través de un tipo particular de canal de comunicaciones.
35
40

El dispositivo principal 210 puede transmitir a continuación los paquetes de datos, que incluyen las tramas USB encapsuladas, a uno o más dispositivos cliente 220 a través de la red (602). Por ejemplo, la red puede corresponder a una red wifi definida por una norma IEEE 802.11. En consecuencia, el dispositivo principal 210 puede transmitir los paquetes de datos a uno o más dispositivos cliente 220 basándose en uno o más protocolos inalámbricos 802.11. Para algunos modos de realización, el dispositivo principal 210 puede transmitir los paquetes de datos a varios dispositivos cliente 220, simultáneamente. Además, para algunos modos de realización, el dispositivo principal 210 puede incluir un contador de tramas USB para realizar un seguimiento del número de tramas USB transmitidas a uno o más dispositivos cliente 220. Por ejemplo, el dispositivo principal 210 puede actualizar su contador de tramas USB para cada trama USB transmitida.
45
50

El dispositivo principal 210 puede transmitir además un conjunto de datos de sincronización al uno o más dispositivos cliente 220 para permitir que cada dispositivo cliente 220 procese las tramas USB (603). Los datos de sincronización pueden incluir, por ejemplo, datos de sincronización de reloj (Clk_Sync), un valor de recuento de tramas (F_Count) y/o un valor de tiempo de medios (M_Time). Más específicamente, los datos Clk_Sync pueden utilizarse para sincronizar una señal de reloj del dispositivo principal 210 con una señal de reloj local del uno o más dispositivos cliente 220 (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A). Para algunos modos de realización, los datos Clk_Sync pueden proporcionarse mediante un mecanismo de sincronización de reloj asociado con el protocolo de comunicaciones inalámbricas. Los valores F_Count y M_Time se pueden utilizar para sincronizar el contador de tramas USB del dispositivo principal 210 con un contador de tramas local del uno o más dispositivos cliente 220 (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4B). Como se ha descrito anteriormente, el valor F_Count puede indicar el número de tramas USB transmitidas por el dispositivo principal 210 en un momento especificado por el valor M_Time. Para algunos modos de realización, el dispositivo principal 210 puede transmitir periódicamente datos de sincronización actualizados para mantener la sincronización con cada uno del uno o más dispositivos cliente 220.
55
60
65

La FIG. 7 es un diagrama de flujo ilustrativo que representa una operación de sincronización USB inalámbrica 700

de acuerdo con algunos modos de realización. Con referencia, por ejemplo, a las FIGS. 4A-4B, el dispositivo principal 210 transmite primero datos de sincronización de reloj a un dispositivo cliente (por ejemplo, dispositivo cliente 220(1)) a través del canal de comunicaciones inalámbricas 230 (701). Para algunos modos de realización, los datos de sincronización de reloj pueden corresponder a un valor de temporizador TSF proporcionado dentro de una trama de baliza. El dispositivo cliente 220(1) recibe los datos de sincronización de reloj a través del canal de comunicaciones inalámbricas 230 (702) y sincroniza su señal de reloj local Clk_1 con la señal de reloj principal Clk_H mediante un mecanismo de sincronización de reloj asociado con el canal de comunicaciones 230 (704). Por ejemplo, el dispositivo cliente 220(1) puede utilizar el valor del temporizador TSF del dispositivo principal para sincronizar su señal de reloj local Clk_1 (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A).

El dispositivo principal 210 encapsula tramas USB para su transmisión a través del canal de comunicaciones inalámbricas 230 (703). Por ejemplo, con referencia a la FIG. 2, la PAL SME 213 puede encapsular tramas USB (por ejemplo, recibidas a través del controlador USB 212) de acuerdo con un protocolo wifi, que se van a transmitir a través de un canal wifi. Para algunos modos de realización, la encapsulación puede realizarse de acuerdo con unos protocolos WGA. El dispositivo principal 210 integra el recuento de tramas y los valores de tiempo de medios actuales en la trama USB encapsulada (705). Por ejemplo, el valor F_Count se puede registrar a partir del contador de tramas principal Cnt_H , y el valor M_Time correspondiente se puede registrar a partir de la señal de reloj principal Clk_H . Para algunos modos de realización, los valores F_Count y M_Time se acoplan entre sí y se registran sustancialmente al mismo tiempo. Finalmente, el dispositivo principal 210 transmite las tramas USB encapsuladas al dispositivo cliente 220(1) a través del canal de comunicaciones inalámbricas 230 (707).

El dispositivo cliente 220(1) recibe las tramas USB encapsuladas a través del canal de comunicaciones inalámbricas (706) y analiza los valores F_Count y M_Time proporcionados con estas (708). Para algunos modos de realización, el dispositivo cliente 220 (1) puede analizar los valores F_Count y M_Time de un paquete de datos recibido sin decapsular los datos USB. El dispositivo cliente 220 (1) sincroniza, a continuación, su contador de tramas local Cnt_1 con el contador de tramas principal Cnt_H basándose en los valores F_Count y M_Time recibidos y la señal de reloj local sincronizada Clk_1 (710). Por ejemplo, el dispositivo cliente 220(1) puede actualizar su contador de tramas local Cnt_1 basándose en la ecuación 1 (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4B).

Para algunos modos de realización, el dispositivo principal 210 puede emitir subsiguientemente una orden de procesar y/o presentar tramas USB comenzando en un recuento de tramas particular (709). Por ejemplo, el recuento de tramas, especificado por el dispositivo principal 210, puede corresponder a un valor de recuento del contador de tramas principal Cnt_H . Al recibir dicha orden, el dispositivo cliente 220(1) puede proceder a procesar y/o presentar tramas USB almacenadas en memoria intermedia basándose en su contador de tramas local Cnt_1 (712). Específicamente, dado que el contador de tramas local Cnt_1 está sincronizado con el contador de tramas principal Cnt_H , el dispositivo cliente 220(1) puede comenzar a procesar sus datos USB almacenados en memoria intermedia a la hora deseada, independientemente del número de tramas USB recibidas y/o almacenadas en memoria intermedia por el dispositivo cliente 220(1).

La operación de sincronización anterior 700 se ha descrito con respecto al dispositivo cliente 220(1) solamente para simplificar. Sin embargo, la operación 700 se puede realizar de una manera sustancialmente similar entre el dispositivo principal 210 y cualquiera de los dispositivos cliente 220(1)-220(3). Además, la operación de sincronización 700 puede asegurar que los paquetes de datos USB transmitidos a varios dispositivos cliente 220(1)-220(3) a través del canal inalámbrico 230 se reproduzcan y/o presenten (por ejemplo, emitan desde altavoces asociados con los dispositivos cliente 220(1)-220(3)) en los momentos correctos (por ejemplo, sincrónicamente).

La FIG. 8 es un diagrama de bloques de un dispositivo cliente habilitado para USB 800 de acuerdo con algunos modos de realización. El dispositivo cliente 800 incluye una interfaz inalámbrica 810, una interfaz USB 820, un procesador 830 y una memoria 840. La interfaz inalámbrica 810 está acoplada al procesador 830 y puede utilizarse para recibir y/o transmitir señales de datos a través de un canal de comunicaciones inalámbricas. Por ejemplo, el canal de comunicaciones inalámbricas puede corresponder a un canal inalámbrico IEEE 802.11. La interfaz USB 820 también está acoplada al procesador 830 y puede utilizarse para comunicar datos USB a y/o desde el dispositivo cliente 800 (por ejemplo, a través de un cable USB). Para algunos modos de realización, la interfaz inalámbrica 810 puede recibir tramas USB, encapsuladas en paquetes de datos, que de otro modo se recibirían a través de la interfaz USB 820.

La memoria 840 puede incluir un almacén de datos 842 que puede utilizarse para almacenar en memoria intermedia tramas USB que el dispositivo 800 va a procesar y/o reproducir. Por ejemplo, las tramas USB recibidas a través de la interfaz inalámbrica 810 pueden almacenarse en el almacén de datos 842. Además, la memoria 840 puede incluir también un medio de almacenamiento no transitorio legible por ordenador (por ejemplo, uno o más elementos de memoria no volátil, tal como EPROM, EEPROM, memoria Flash, un disco duro, etc.) que pueden almacenar los módulos de software siguientes:

- un módulo de sincronización de reloj 844 para sincronizar una señal de reloj local con una señal de reloj de un dispositivo principal;

- un módulo de sincronización de tramas 846 para sincronizar un contador de tramas local con un contador de tramas USB de un dispositivo principal; y

5 • un módulo de procesamiento USB 848 para procesar tramas USB recibidas basándose en un valor de recuento de tramas del contador de tramas local.

10 Cada módulo de software puede incluir instrucciones que, cuando son ejecutadas por el procesador 830, pueden hacer que el dispositivo 800 desempeñe la correspondiente función. Por lo tanto, el medio de almacenamiento no transitorio legible por ordenador de la memoria 840 puede incluir instrucciones para realizar la totalidad o una parte de las operaciones descritas con respecto a las FIGS. 5 y 7.

15 El procesador 830, que está acoplado a la memoria 840, puede ser cualquier procesador adecuado capaz de ejecutar secuencias de mandatos de instrucciones de uno o más programas de software almacenados en el dispositivo cliente 800 (por ejemplo, en la memoria 840). Por ejemplo, el procesador 830 puede ejecutar el módulo de sincronización de reloj 844, el módulo de sincronización de tramas 846 y/o el módulo de procesamiento USB 848.

20 El módulo de sincronización de reloj 844 puede ser ejecutado por el procesador 830 para sincronizar una señal de reloj local con una señal de reloj de un dispositivo principal. Por ejemplo, el dispositivo cliente 800 puede recibir datos de sincronización de reloj (Clk_Sync) desde un dispositivo principal a través de la interfaz inalámbrica 810. Para algunos modos de realización, los datos Clk_Sync pueden corresponder a un valor TSF que el dispositivo principal transmite periódicamente mediante radiodifusión (por ejemplo, en tramas de baliza). El módulo de sincronización de reloj 844, ejecutado por el procesador 830, puede sincronizar su señal de reloj local con una señal de reloj principal, estableciendo su propio temporizador TSF local en el valor TSF recibido (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A).

25 El módulo de sincronización de tramas 846 puede ser ejecutado por el procesador 830 para sincronizar un contador de tramas local con un contador de tramas USB de un dispositivo principal. Por ejemplo, el dispositivo cliente 800 puede recibir además un valor de recuento de tramas (F_Count) y un valor de tiempo de medios (M_Time) desde el dispositivo principal a través de la interfaz inalámbrica 810. Como se ha descrito anteriormente, el valor F_Count puede indicar un número de tramas USB transmitidas por el dispositivo principal en un momento determinado indicado por el valor M_Time. El módulo de sincronización de tramas 846, ejecutado por el procesador 830, puede sincronizar su contador de tramas local con el contador de tramas principal basándose en los valores F_Count y M_Time, la señal de reloj local y la velocidad de transmisión del dispositivo principal (por ejemplo, como se describe anteriormente con respecto a la FIG. 4B). Para algunos modos de realización, el procesador 830, al ejecutar el módulo de sincronización de tramas 846, puede actualizar (por ejemplo, incrementar) continuamente el contador de tramas locales para cada trama USB recibida desde el dispositivo principal.

30 El módulo de procesamiento USB 848 puede ser ejecutado por el procesador 830 para procesar las tramas USB recibidas basándose en un valor de recuento de tramas del contador de tramas local. Por ejemplo, el módulo de procesamiento USB 848, ejecutado por el procesador 830, puede comenzar a procesar tramas USB almacenadas en el almacén de datos 842 una vez que el contador de tramas local alcanza un umbral de recuento de tramas particular. Para algunos modos de realización, el dispositivo principal puede especificar el umbral de recuento de tramas (por ejemplo, transmitiendo un mensaje de petición de procesamiento al dispositivo cliente 800). Debido a que el contador de tramas local puede estar sincronizado con el contador de tramas principal, el procesador 830, al ejecutar el módulo de procesamiento USB 848, puede comenzar a procesar tramas USB en cualquier momento indicado por el dispositivo principal.

35 La FIG. 9 es un diagrama de bloques de un dispositivo principal habilitado para USB 900 de acuerdo con algunos modos de realización. El dispositivo principal 900 incluye también una interfaz inalámbrica 910, un procesador 920 y una memoria 930. La interfaz inalámbrica 910 está acoplada al procesador 920 y puede utilizarse para transmitir y/o recibir señales de datos a través de un canal de comunicaciones inalámbricas. Por ejemplo, el canal de comunicaciones inalámbricas puede corresponder a un canal inalámbrico IEEE 802.11. Para algunos modos de realización, la interfaz inalámbrica 910 puede transmitir de forma inalámbrica tramas USB encapsuladas en paquetes de datos.

40 La memoria 930 puede incluir un almacén de datos 932 que puede utilizarse para almacenar en memoria intermedia tramas USB que se van a encapsular y transmitir a uno o más dispositivos cliente. Además, la memoria 930 puede incluir también un medio de almacenamiento no transitorio legible por ordenador (por ejemplo, uno o más elementos de memoria no volátil, tal como EPROM, EEPROM, memoria Flash, un disco duro, etc.) que pueden almacenar los módulos de software siguientes:

- un módulo de sincronización de reloj 934 para sincronizar una señal de reloj principal con una señal de reloj local de un dispositivo;

65 • un módulo de sincronización de tramas 936 para sincronizar un contador de tramas principal con un contador de tramas locales del dispositivo cliente; y

- un módulo de encapsulación USB 938 para codificar tramas USB en paquetes de datos para su transmisión a través de la interfaz inalámbrica 910.

- 5 Cada módulo de software puede incluir instrucciones que, cuando son ejecutadas por el procesador 920, pueden hacer que el dispositivo 900 desempeñe la correspondiente función. Por lo tanto, el medio de almacenamiento no transitorio legible por ordenador de la memoria 930 puede incluir instrucciones para realizar la totalidad o una parte de las operaciones descritas con respecto a las FIGS. 6 y 7.
- 10 El procesador 920, que está acoplado a la memoria 930, puede ser cualquier procesador adecuado capaz de ejecutar secuencias de mandatos de instrucciones de uno o más programas de software almacenados en el dispositivo principal 900 (por ejemplo, en la memoria 930). Por ejemplo, el procesador 920 puede ejecutar el módulo de sincronización de reloj 934, el módulo de sincronización de tramas 936 y/o el módulo de procesamiento USB 938.
- 15 El procesador 920 puede ejecutar el módulo de sincronización de reloj 934 para sincronizar una señal de reloj principal con una señal de reloj local de un dispositivo cliente. Por ejemplo, el módulo de sincronización de reloj 934, ejecutado por el procesador 920, puede generar datos de sincronización de reloj (Clk_Sync) que se van a transmitir a uno o más dispositivos cliente a través de la interfaz inalámbrica 910. Para algunos modos de realización, los datos Clk_Sync pueden corresponder a un valor TSF basado en un temporizador TSF del dispositivo principal 900
- 20 (por ejemplo, definido por una especificación IEEE 802.11). Cada uno del uno o más dispositivos cliente puede sincronizar, a continuación, su señal de reloj local con la señal de reloj principal mediante los datos Clk_Sync (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4A). Para algunos modos de realización, el procesador 920, al ejecutar el módulo de sincronización de reloj 934, puede transmitir periódicamente los datos Clock_Sync mediante radiodifusión (por ejemplo, dentro de tramas de baliza) a varios dispositivos cliente, de forma simultánea.
- 25

El módulo de sincronización de tramas 936 puede ser ejecutado por el procesador 920 para sincronizar un contador de tramas principal con un contador de tramas local del dispositivo cliente. Por ejemplo, el módulo de sincronización de tramas 936, ejecutado por el procesador 920, puede generar un valor de recuento de tramas (F_Count) y un valor de tiempo de medios (M_Time) que se van a transmitir o a uno o más dispositivos cliente a través de la interfaz inalámbrica 910. Para algunos modos de realización, el valor F_Count puede corresponder al valor de recuento del contador de tramas principal en un momento particular indicado por el valor M_Time. Cada uno del uno o más dispositivos cliente puede sincronizar, a continuación, su contador de tramas local con el contador de tramas principal basándose en los valores F_Count y M_Time, la señal de reloj local y la velocidad de transmisión del dispositivo principal (por ejemplo, como se ha descrito anteriormente con respecto a la FIG. 4B). Para algunos modos de realización, el procesador 920, al ejecutar el módulo de sincronización de tramas 936, puede actualizar continuamente (por ejemplo, incrementar) el contador de tramas principal para cada trama USB transmitida al uno o más dispositivos cliente.

30

35

- 40 El procesador 920 puede ejecutar el módulo de procesamiento USB 938 para codificar tramas USB en paquetes de datos para su transmisión a través de la interfaz inalámbrica 910. Por ejemplo, el módulo de procesamiento USB 938, ejecutado por el procesador 920, puede encapsular las tramas USB escribiendo datos USB correspondientes en uno o más paquetes de datos que se van a transmitir de forma inalámbrica. Para algunos modos de realización, la encapsulación puede realizarse de acuerdo con unos protocolos WGA. Además, para algunos modos de
- 45 realización, el procesador 920, al ejecutar el módulo de procesamiento USB 938, puede generar un mensaje de petición de procesamiento que se va a transmitir, junto con las tramas USB, al uno o más dispositivos cliente. Como se ha descrito anteriormente, el mensaje de petición de procesamiento puede incluir instrucciones para procesar las tramas USB correspondientes (por ejemplo, basándose en un recuento de tramas USB particular).

- 50 Debe observarse que, para algunos modos de realización, el dispositivo cliente 800 puede desempeñar una o más funciones del dispositivo principal 900, y viceversa. Por ejemplo, el dispositivo cliente 800 puede incluir uno o más módulos de software (por ejemplo, 934-938) del dispositivo principal 900. De forma similar, el dispositivo principal 900 puede incluir uno o más módulos de software (por ejemplo, 844-848) del dispositivo cliente 800.

- 55 En la memoria descriptiva anterior, la presente invención se ha descrito con referencia a unos ejemplos específicos de modos de realización de la misma. No obstante, resultará evidente que pueden hacerse diversas modificaciones y cambios a la misma sin apartarse del alcance más amplio de la divulgación, expuesto en las reivindicaciones adjuntas. La memoria descriptiva y los dibujos, en consecuencia, deben considerarse en un sentido ilustrativo, en lugar de en un sentido restrictivo. Por ejemplo, las etapas del procedimiento representadas en los diagramas de flujo de las FIGS. 5-7 se pueden realizar en otros órdenes adecuados, se pueden combinar varias etapas en una sola
- 60 etapa y/o se pueden omitir algunas etapas.

REIVINDICACIONES

1. Un procedimiento (500) para el funcionamiento de un dispositivo de bus serie universal, USB, en una red, comprendiendo el procedimiento:

5 recibir (501) una o más tramas USB desde un dispositivo principal (210) a través de la red, en el que la una o más tramas USB se encapsulan en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con la red; **caracterizado por** sincronizar (502) una señal de reloj local del dispositivo USB con una señal de reloj del dispositivo principal mediante un mecanismo de sincronización de reloj del protocolo de comunicaciones;

10 determinar (503) un número de tramas USB transmitidas por el dispositivo principal basándose, al menos en parte, en la señal de reloj local sincronizada; y

15 procesar (504) la una o más tramas USB basándose, al menos en parte, en el número de tramas USB transmitidas por el dispositivo principal.
2. El procedimiento de acuerdo con la reivindicación 1, en el que el protocolo de comunicaciones corresponde a un protocolo inalámbrico IEEE 802.11.
3. El procedimiento de acuerdo con la reivindicación 1, en el que sincronizar la señal de reloj local comprende:

 recibir una o más tramas de baliza desde el dispositivo principal, en el que al menos una de las tramas de baliza incluye un valor de función de sincronización de temporización, TSF; y

25 ajustar la señal de reloj local basándose en el valor TSF recibido.
4. El procedimiento de acuerdo con la reivindicación 1, que comprende además:

30 almacenar la una o más tramas USB en una memoria intermedia de datos.
5. El procedimiento de acuerdo con la reivindicación 4, en el que determinar el número de tramas USB transmitidas por el dispositivo principal comprende:

35 actualizar un contador de tramas local basándose, al menos en parte, en un número de las tramas USB almacenadas en la memoria intermedia de datos; y

 determinar el número de tramas USB transmitidas por el dispositivo principal basándose en el contador de tramas local.

40
6. El procedimiento de acuerdo con la reivindicación 5, que comprende además:

 recibir un valor de recuento de tramas desde el dispositivo principal, en el que el valor de recuento de tramas indica el número de tramas USB transmitidas por el dispositivo principal; y

45 recibir un valor de tiempo de medios desde el dispositivo principal, en el que el valor de tiempo de medios indica una hora a la que se registra el valor de recuento de tramas.
7. El procedimiento de acuerdo con la reivindicación 6, en el que el valor de recuento de tramas y el valor de tiempo de medios se reciben juntos en al menos una de las tramas USB.

50
8. El procedimiento de acuerdo con la reivindicación 6, en el que determinar el número de tramas USB transmitidas por el dispositivo principal comprende además:

55 sincronizar el contador de tramas local con un contador de tramas USB en el dispositivo principal basándose, al menos en parte, en la señal de reloj local, el valor de recuento de tramas y el valor de tiempo de medios.
9. El procedimiento de acuerdo con la reivindicación 8, en el que procesar la una o más tramas USB comprende:

60 procesar la una o más tramas USB cuando el contador de tramas local alcanza un primer umbral.
10. Un procedimiento (600) de funcionamiento de un dispositivo informático en una red, comprendiendo el procedimiento:

65

encapsular (601) una o más tramas de bus serie universal, USB, en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con la red;

5 transmitir (602) el uno o más paquetes de datos, que incluyen la una o más tramas USB encapsuladas, a uno o más dispositivos cliente (220) a través de la red; y

10 transmitir (603) un conjunto de datos de sincronización, al uno o más dispositivos cliente, que permite que el uno o más dispositivos cliente procesen la una o más tramas USB, en el que el conjunto de datos de sincronización incluye datos de sincronización de reloj que permiten que el uno o más dispositivos cliente sincronicen señales de reloj local respectivas con una señal de reloj del dispositivo informático.

11. El procedimiento de acuerdo con la reivindicación 10, en el que transmitir datos de sincronización de reloj comprende:

15 radiodifundir los datos de sincronización de reloj dentro de una o más tramas de baliza, en el que los datos de sincronización de reloj incluyen un valor de función de sincronización de temporización (TSF).

12. El procedimiento de acuerdo con la reivindicación 10, que comprende además:

20 actualizar un contador de tramas USB basándose en un número de tramas USB transmitidas al uno o más dispositivos cliente.

25 13. Un medio de almacenamiento legible por ordenador que contiene instrucciones de programa que, cuando son ejecutadas por un procesador proporcionado dentro de un dispositivo de bus serie universal, USB, hace que el dispositivo USB realice el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 12.

14. Un dispositivo de bus serie universal, USB, que comprende:

30 medios para recibir una o más tramas USB desde un dispositivo principal (210) a través de una red, en el que la una o más tramas USB se encapsulan en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con la red;

35 medios para sincronizar una señal de reloj local del dispositivo USB con una señal de reloj del dispositivo principal mediante un mecanismo de sincronización de reloj del protocolo de comunicaciones;

medios para determinar un número de tramas USB transmitidas por el dispositivo principal basándose, al menos en parte, en la señal de reloj local sincronizada; y

40 medios para procesar la una o más tramas USB basándose, al menos en parte, en el número de tramas USB transmitidas por el dispositivo principal.

15. Un dispositivo informático, que comprende:

45 medios para encapsular una o más tramas de bus serie universal, USB, en uno o más paquetes de datos basándose en un protocolo de comunicaciones asociado con una red;

medios para transmitir el uno o más paquetes de datos, que incluyen la una o más tramas USB encapsuladas, a uno o más dispositivos cliente (220) a través de la red; y

50 medios para transmitir un conjunto de datos de sincronización, al uno o más dispositivos cliente a través de la red inalámbrica, que permite que el uno o más dispositivos cliente procesen la una o más tramas USB, en el que el conjunto de datos de sincronización incluye datos de sincronización de reloj que permiten que el uno o más dispositivos cliente sincronicen señales de reloj local respectivas con una señal de reloj del dispositivo informático.

55 16. El procedimiento de acuerdo con la reivindicación 10, que comprende además:

60 transmitir una petición de procesamiento al uno o más dispositivos cliente, en el que la petición de procesamiento incluye una orden para comenzar a procesar la una o más tramas USB.

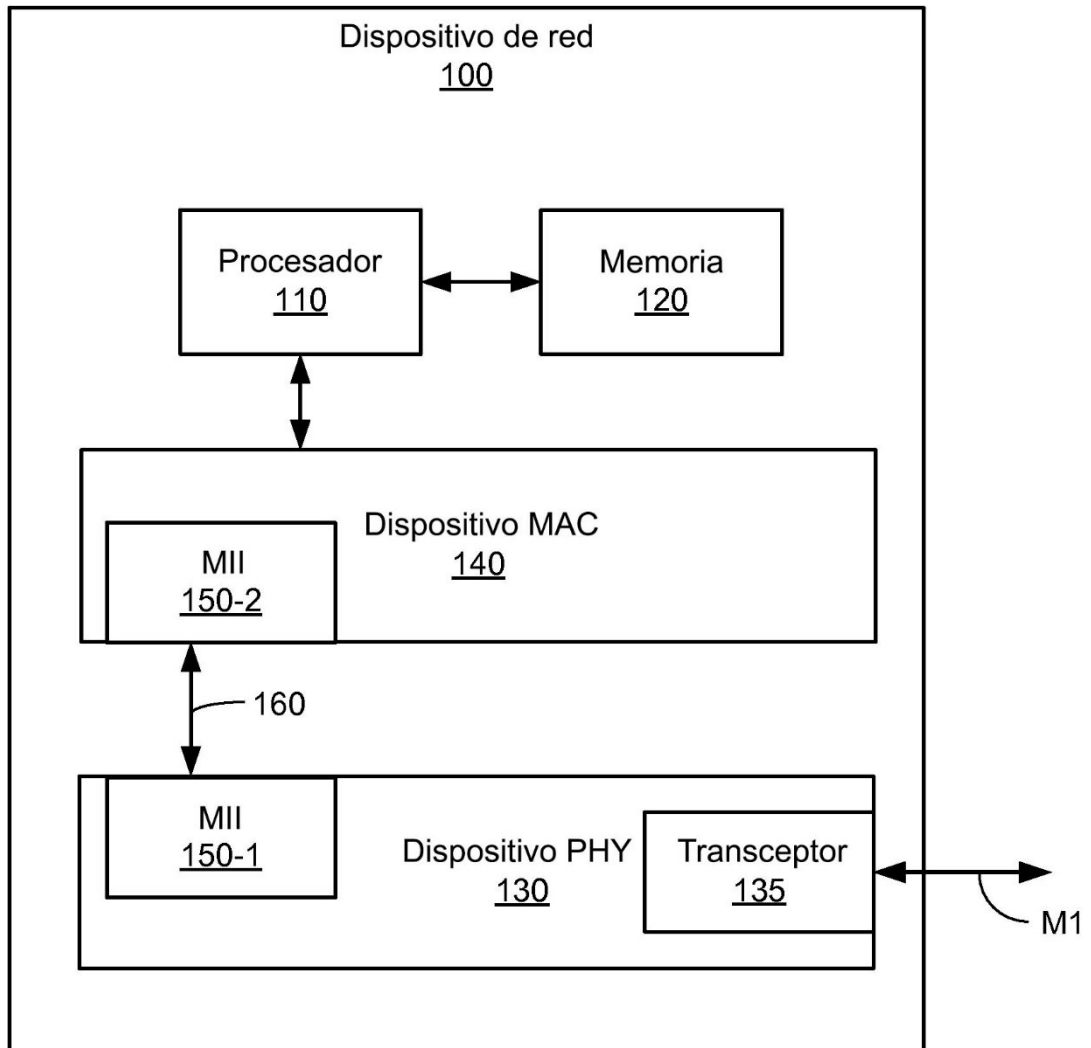


FIG. 1

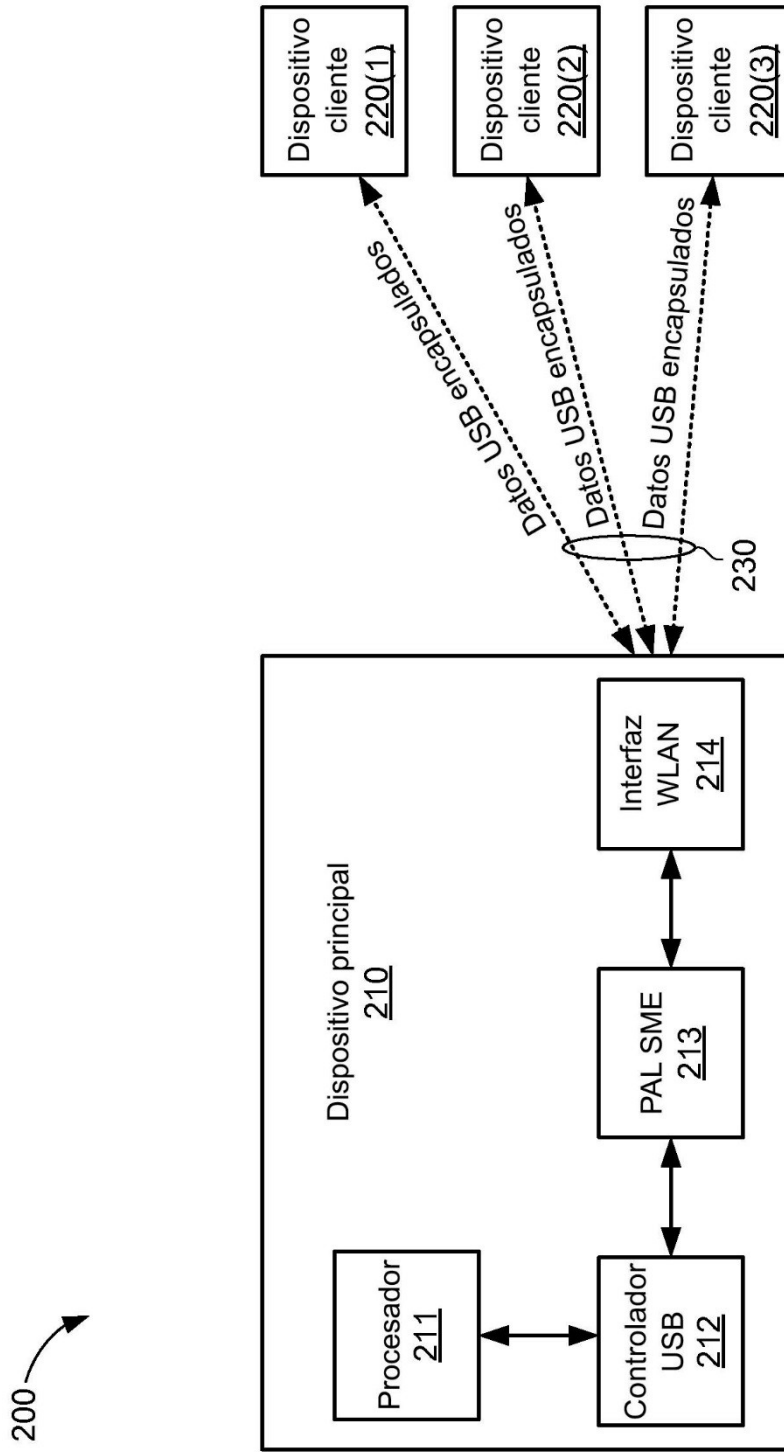


FIG. 2

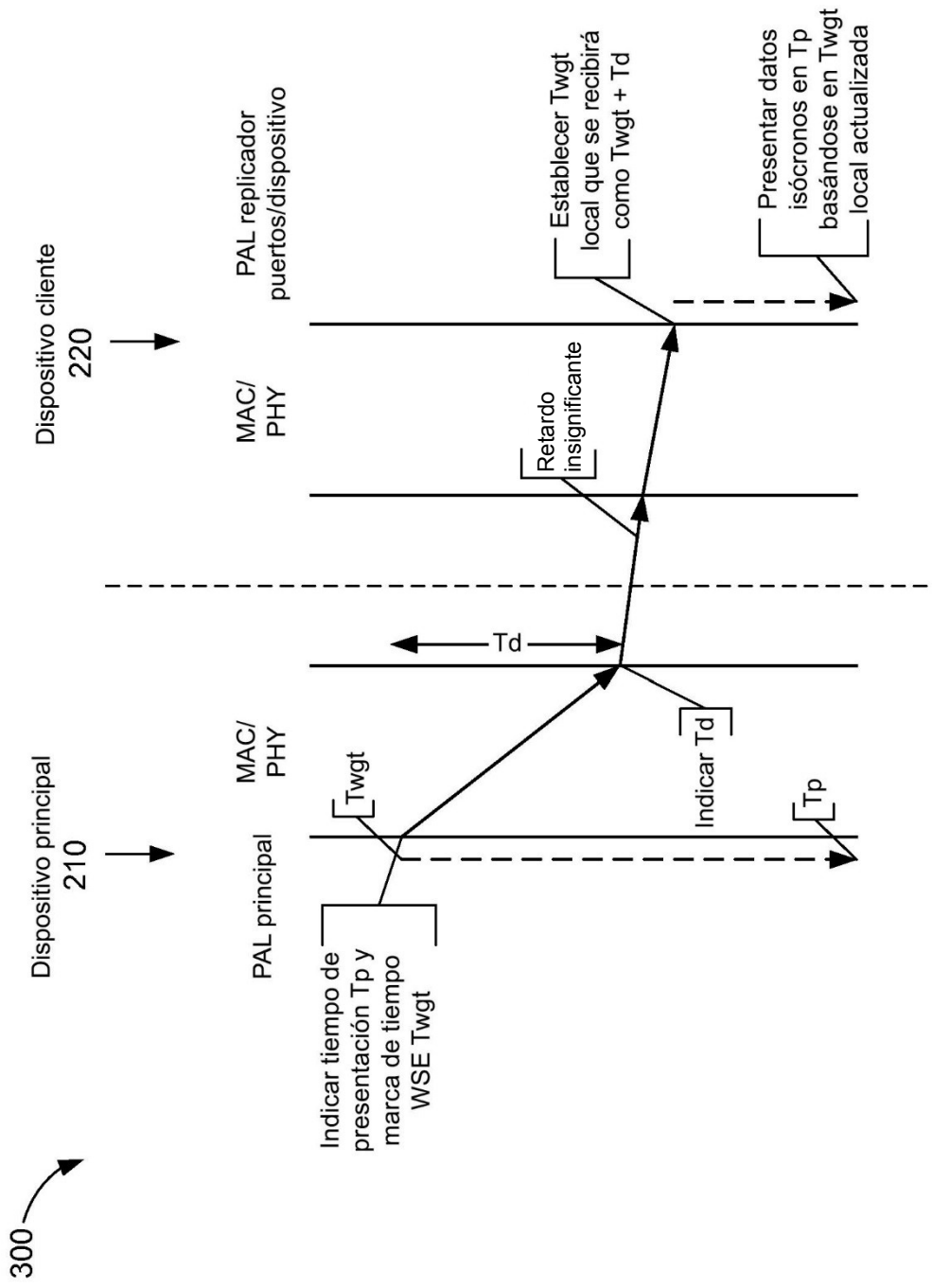


FIG. 3

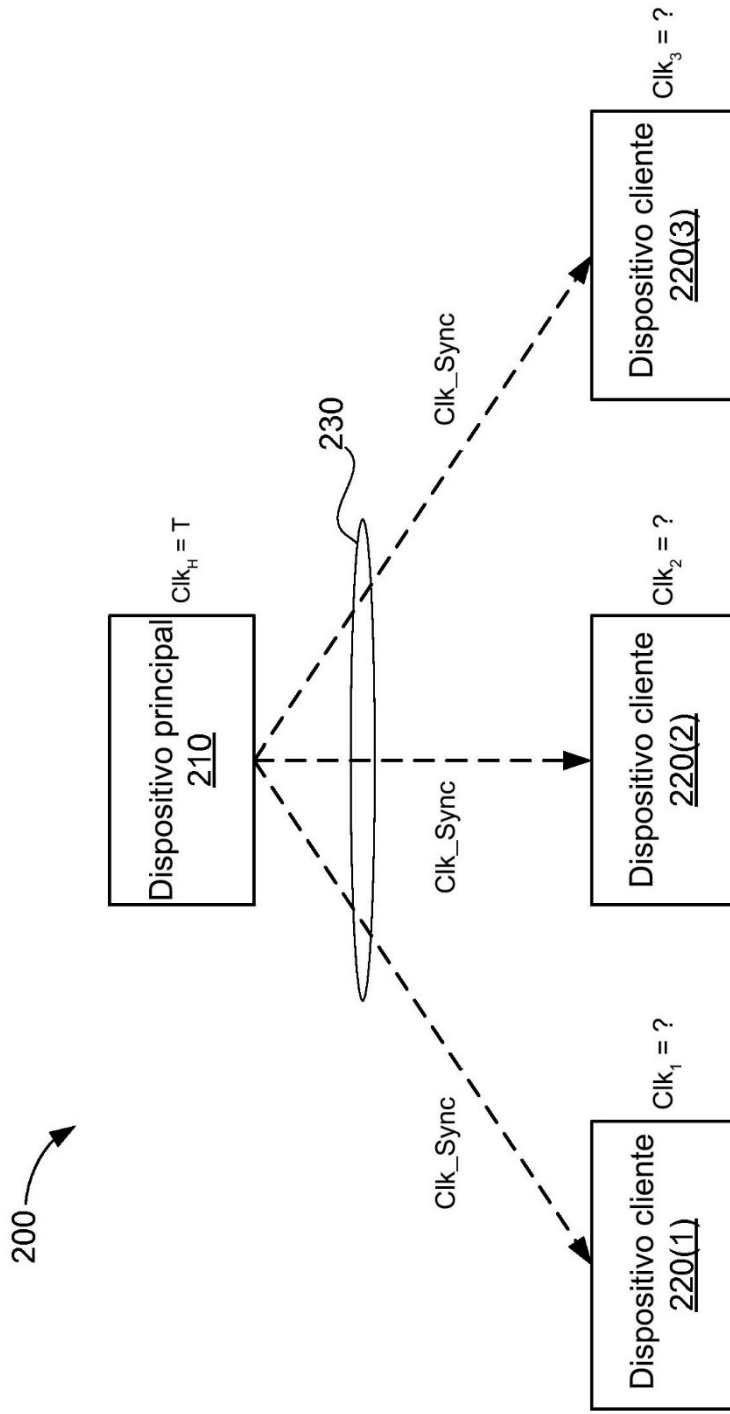


FIG. 4A

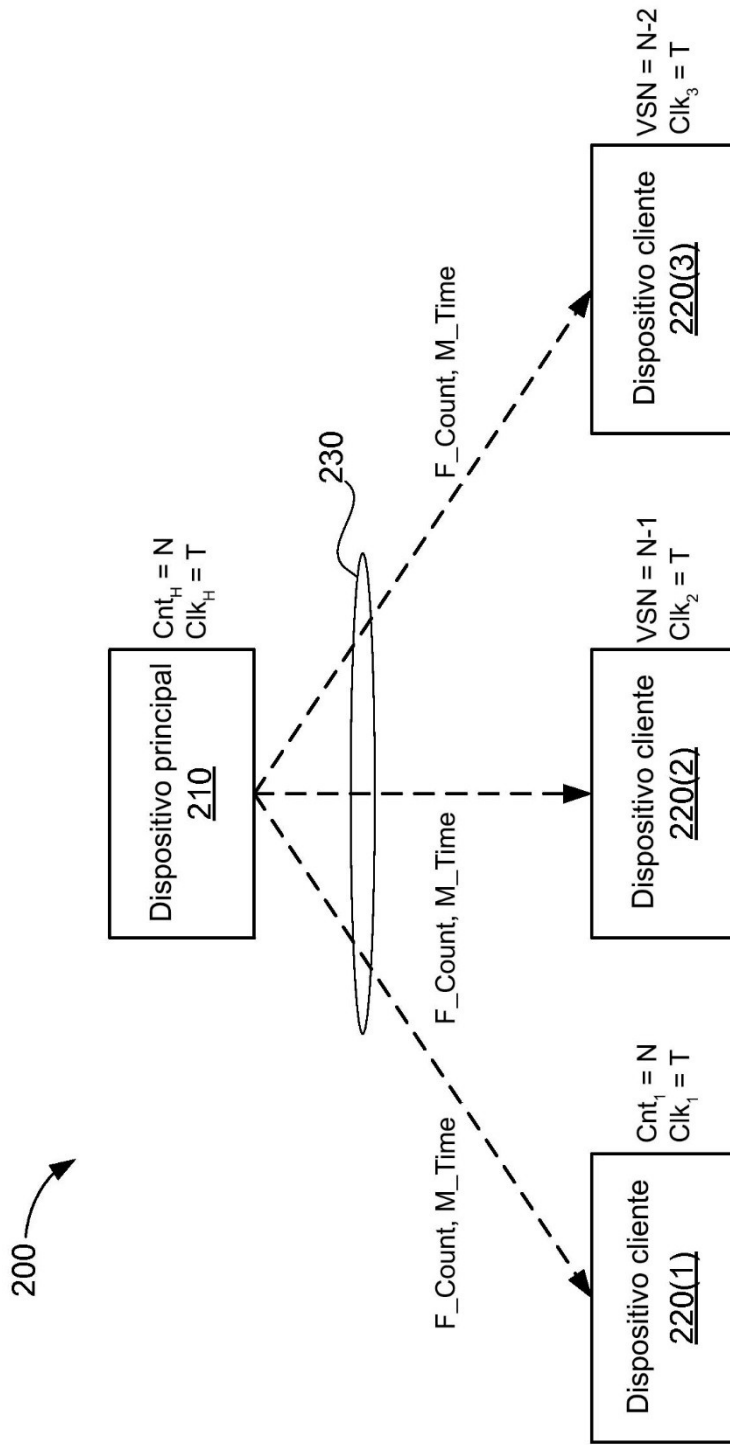


FIG. 4B

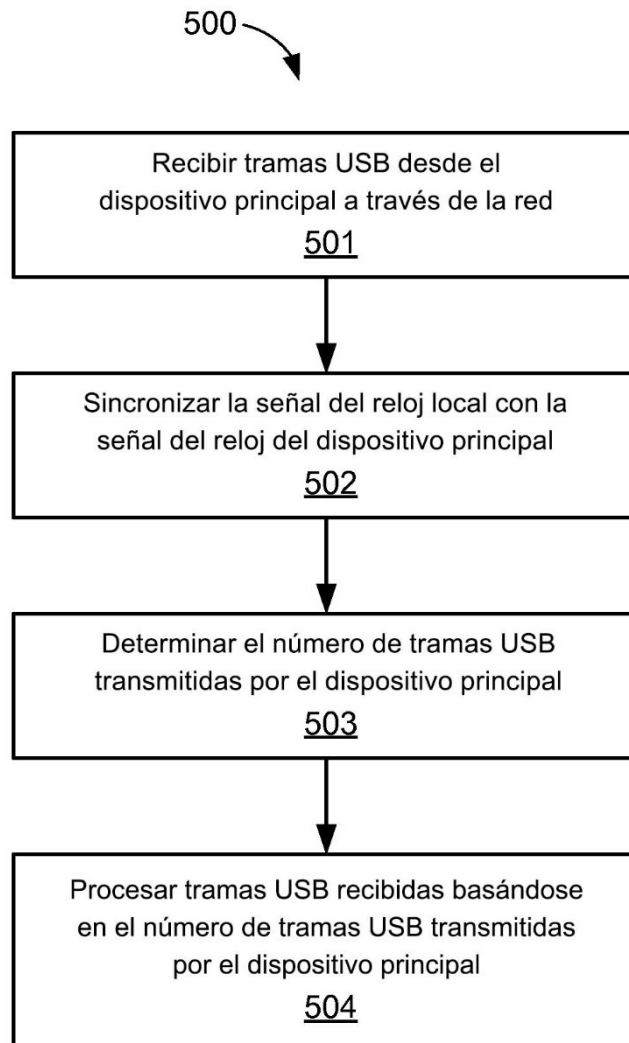


FIG. 5

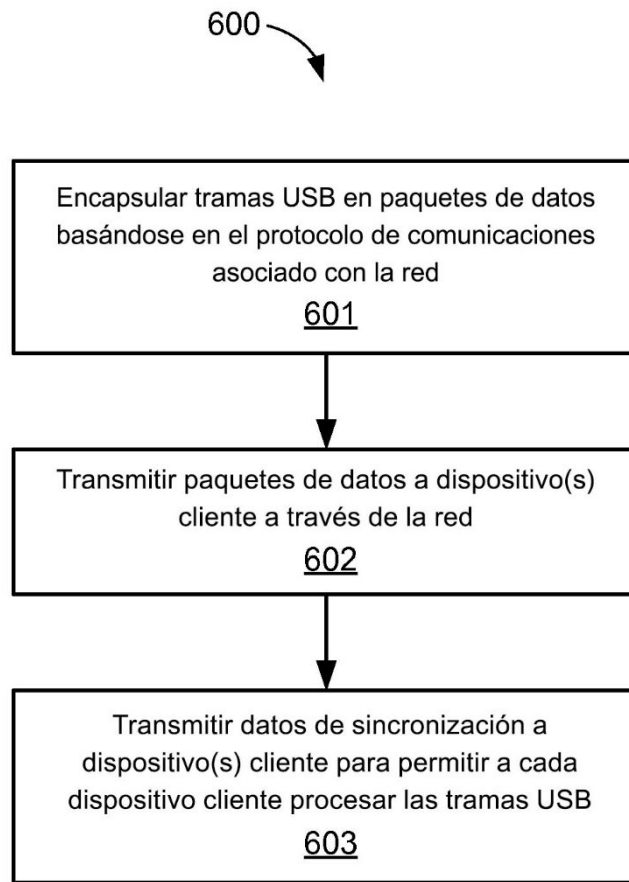


FIG. 6

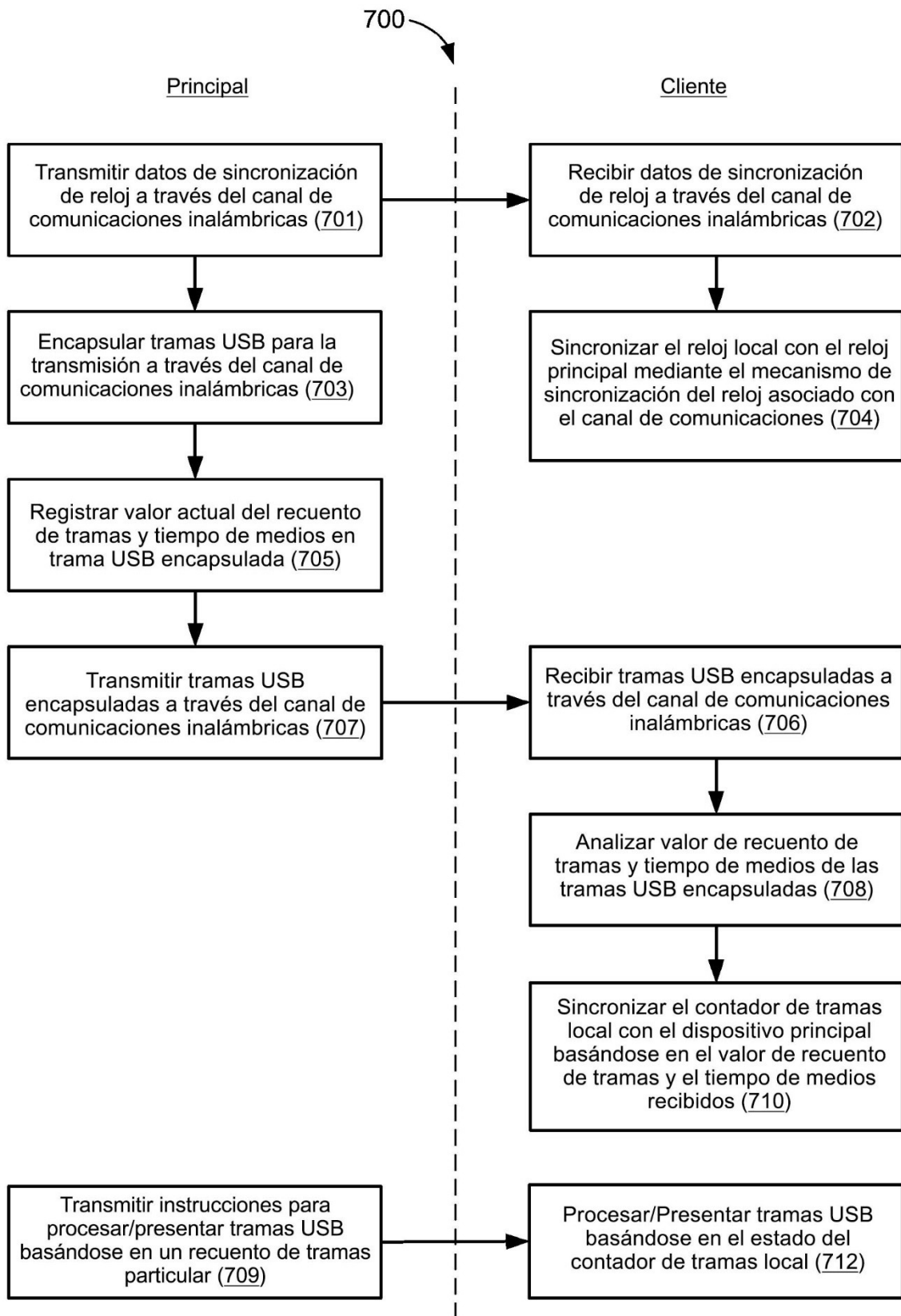


FIG. 7

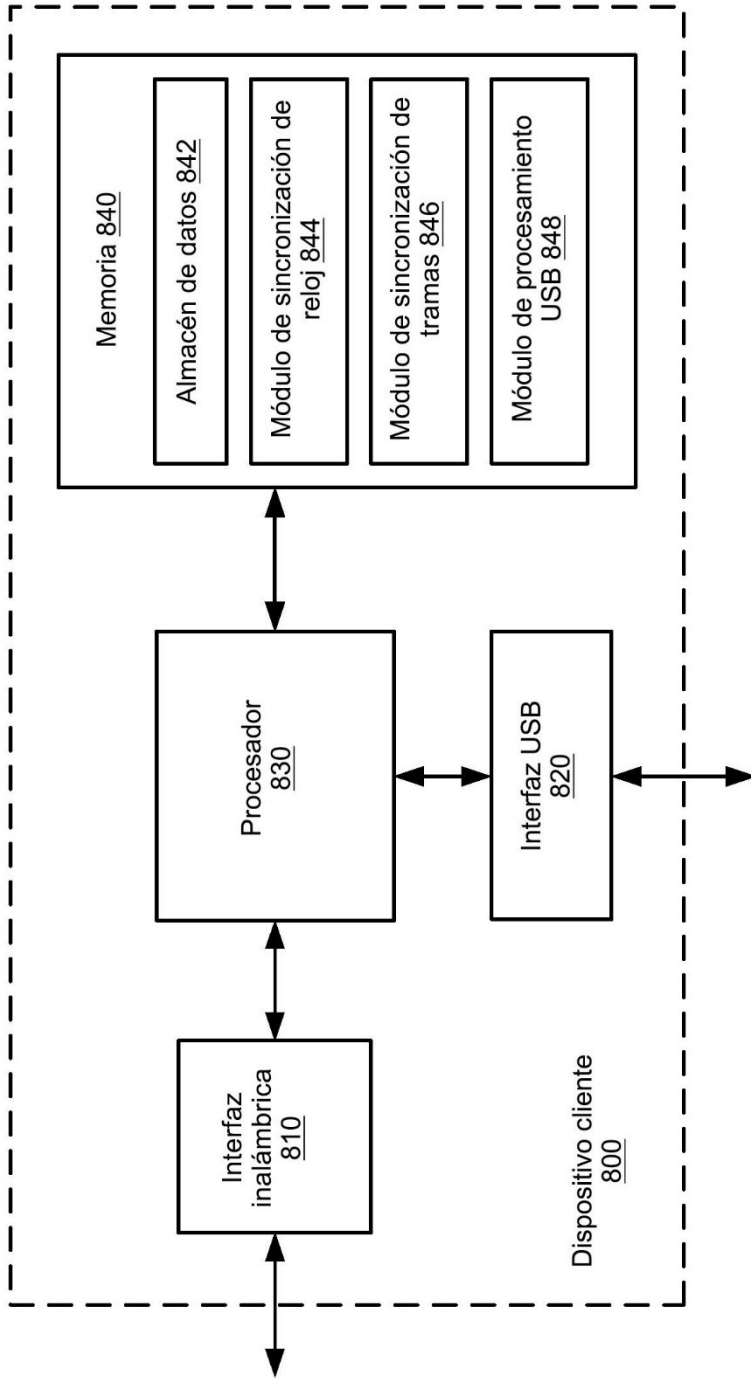


FIG. 8

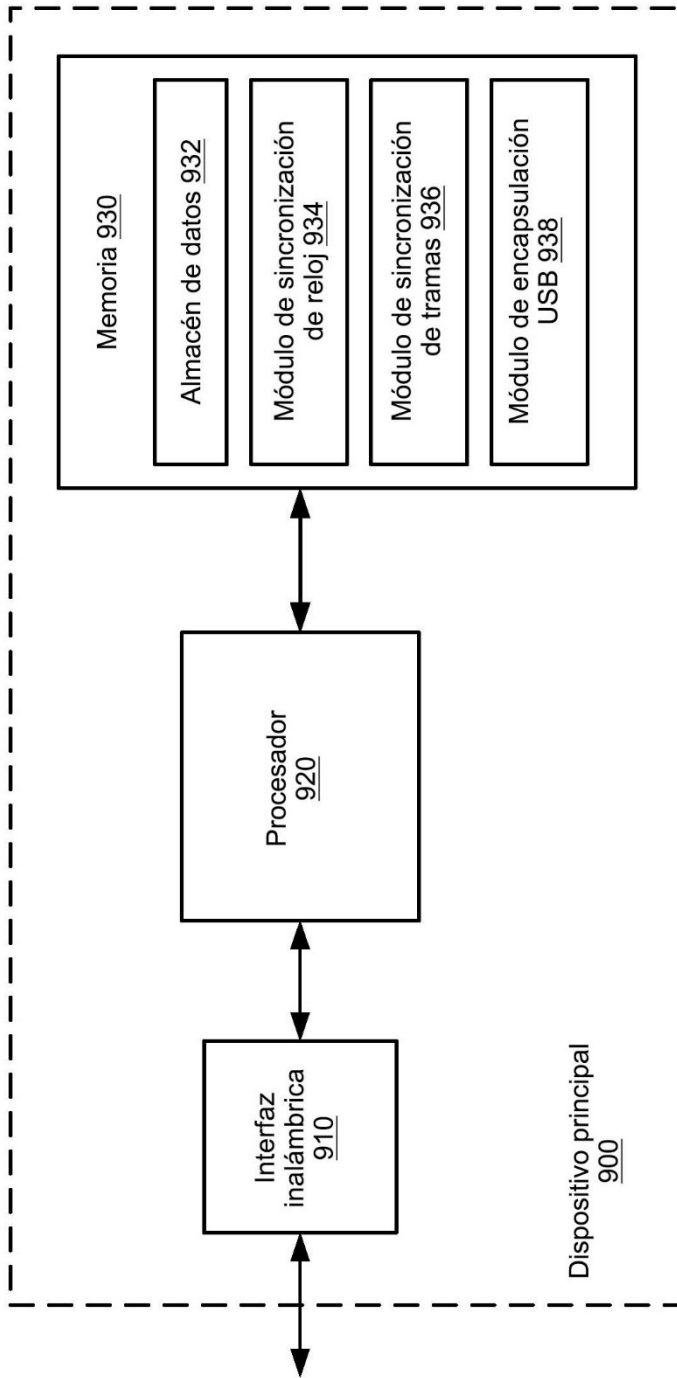


FIG. 9