

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 987 741**

51 Int. Cl.:

H04L 67/12 (2012.01)

G06F 9/54 (2006.01)

G08G 1/0967 (2006.01)

H04W 4/80 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **02.07.2020 PCT/GB2020/051590**

87 Fecha y número de publicación internacional: **07.01.2021 WO21001655**

96 Fecha de presentación y número de la solicitud europea: **02.07.2020 E 20740381 (7)**

97 Fecha y número de publicación de la concesión europea: **04.09.2024 EP 3994869**

54 Título: **Modelo de coordinación distribuido basado en eventos**

30 Prioridad:

02.07.2019 GB 201909545

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

18.11.2024

73 Titular/es:

**PHINXT ROBOTICS LIMITED (100.0%)
Flat 201 Loren Apartments, 51 Blair Street
London E14 0HY, GB**

72 Inventor/es:

CHEN, YANWEN

74 Agente/Representante:

VÁZQUEZ FERNÁNDEZ-VILLA, Concepción

ES 2 987 741 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

Modelo de coordinación distribuido basado en eventos

5 **Campo**

La presente invención se refiere a la coordinación de tareas en una malla distribuida de sistemas, tal como entre vehículos autónomos, sistemas de cámaras múltiples, sistemas de nodos distribuidos IoT y/o fábricas modulares en una variedad de casos de uso. Más particularmente, la presente invención se refiere a coordinar eventos en una malla distribuida de sistemas basándose en parámetros y restricciones y puede usarse para proporcionar una disposición de mensajería que permite que los sistemas distribuidos se comuniquen y garanticen que se completan las tareas dentro del sistema distribuido y/o en uno o más nodos dentro del sistema distribuido en uno o más momentos sustancialmente predeterminados.

15 **Antecedentes**

Cada vez más, se automatizan tareas manuales mediante máquinas y las máquinas que se usan para proporcionar esta automatización se mejoran de tal manera que cada vez más puedan funcionar "de manera inteligente", por ejemplo, considerando riesgos tales como riesgos de infraestructura o de seguridad humana a partir de decisiones distribuidas automatizadas. En particular, en industrias como almacenamiento, logística y automoción, ya se han realizado grandes mejoras en la eficiencia, mientras que otras industrias como la minería sólo están volviéndose ahora más automatizadas.

El paradigma actual requiere conexiones de red centralizadas estables para coordinar centralmente todas las máquinas y dispositivos autónomos en una operación dada, sin embargo, esto no siempre es posible. Específicamente, si se usan protocolos de comunicación tales como LTE o 5G, entonces normalmente estos no son lo suficientemente rápidos o no son ubicuos en la mayoría de las áreas geográficas. Las limitaciones de tales redes centralizadas resultarán evidentes y, por ejemplo, incluyen el problema de tener que configurar una infraestructura celular en ubicaciones remotas donde no existe ninguna. En algunas situaciones, por ejemplo, en la exploración subacuática, minas en tajo cerrado, regiones montañosas y el espacio, podría no ser ni siquiera posible desplegar la infraestructura de manera económica. Además, operar un servidor de gestión centralizado puede proporcionar un único punto de debilidad en un sistema distribuido por lo demás.

Por tanto, parece deseable proporcionar un método para evitar la necesidad de coordinación central de máquinas autónomas en tales operaciones, que podría evitar entonces problemas que surgen de enlaces de comunicación deficientes entre una o más de las máquinas.

Sin embargo, las redes distribuidas actuales sólo se usan normalmente en casos de uso específicos tales como redes en malla ancladas, arquitecturas web y operaciones de desarrollador para ampliar a escala aplicaciones web. Sin embargo, estas implementaciones conocidas de redes distribuidas aún no se aplican o adaptan a aplicaciones críticas para la seguridad (por ejemplo, vehículos autónomos) porque la sincronización de tareas en una red distribuida no se considera fiable debido a la falta de un reloj común o "global". En tales redes distribuidas, cada máquina/sistema tiene su propio reloj interno, es decir, su propio registro de tiempo, pero es difícil garantizar que todas las máquinas/sistemas tengan relojes internos sincronizados. Como ejemplo, una desviación de 150 ms entre relojes internos entre nodos en un sistema distribuido puede significar que un nodo que es un vehículo autónomo pueda tener una posición estimada mal calculada que está a aproximadamente 5 metros de distancia, en una autopista, con respecto a su posición real en relación con otro nodo, de nuevo por ejemplo otro vehículo autónomo que es una máquina/sistema en la misma red. Como resultado, incluso lo que parecen ser pequeñas diferencias de tiempo o errores de sincronización pueden conducir a una alta probabilidad de errores y/o consecuencias graves cuando se trata de coordinar centralmente varios nodos dentro de un sistema distribuido.

Además, cuando la secuenciación se basa en sellos de tiempo, normalmente no se intercambia o se intercambia muy poca cantidad de información contextual entre cada máquina/sistema o dentro de un proceso de secuenciación distribuida. Los datos de sello de tiempo intercambiados pueden consumir una parte significativa del ancho de banda disponible en algunas configuraciones, especialmente si se incluye información de direccionamiento con los datos de sello de tiempo. Como resultado, para la toma de decisiones dentro de la secuenciación o entre máquinas/sistemas, esta falta de información contextual limita la toma de decisiones descentralizada. Sin embargo, aumentar la cantidad de información contextual proporcionada también agravaría el problema del consumo de ancho de banda por los intercambios de datos, que ya puede estar limitado por la cantidad de datos de sello de tiempo intercambiados.

La publicación de patente estadounidense número US2018/335784 describe un sistema para intercambiar información de desplazamiento contextual, donde las alertas y acciones sugeridas se reciben desde otros nodos.

65 **Sumario de la invención**

Las realizaciones tratan de proporcionar un método para coordinar máquinas/nodos/dispositivos de una red

distribuida, sin un componente de sistema centralizado, sincronizando dispositivos de un sistema distribuido sin usar un sello de tiempo absoluto. En su lugar, los nodos de una red distribuida pueden construir de manera colaborativa un árbol de dependencias lógico, basado en eventos para gestionar decisiones en cada nodo de una red distribuida.

5 Las realizaciones tratan de proporcionar un método para coordinar tareas en un sistema distribuido de nodos usando parámetros y restricciones locales para determinar el mejor actor en una malla para una tarea particular.

10 Las realizaciones tratan de proporcionar una resolución de seguridad distribuida para garantizar que un nodo de una red distribuida no comience a emprender una acción o nueva secuencia de eventos que pondrían uno o más de otros nodos dentro de la red en riesgo (por ejemplo, el riesgo de daño físico en el contexto de un sistema físico virtual o, más generalmente, el riesgo (lógico) de un escenario de interbloqueo).

15 Otras realizaciones tratan de proporcionar una disposición o un protocolo de mensajería complejo para que los nodos en un sistema distribuido se comuniquen para garantizar que las tareas se completen de manera eficaz.

Según un primer aspecto, se proporciona un método según la reivindicación 1.

20 El uso de una sincronización basada en eventos puede permitir la coordinación de nodos/máquinas/sistemas dentro de sistemas distribuidos eliminando la dependencia de tales sistemas de un reloj basado en el tiempo, y/o sin que los nodos/máquinas/sistemas tengan que pasar a través de un regulador o servidor central para gestionar tareas o eventos. Más particularmente, usando este enfoque, cuando los componentes de una red distribuida se comunican entre sí, no puede existir la dependencia de un reloj basado en el tiempo absoluto para sincronizar las comunicaciones.

25 De esta manera, no hay necesidad de que los nodos de una red estén conectados centralmente y, por tanto, pueden proporcionar la flexibilidad de desplegar un sistema distribuido en cualquier lugar independientemente de la disponibilidad de la red.

30 Opcionalmente, la sincronización basada en eventos comprende coordinar eventos para la pluralidad de nodos vecinos para adaptarse automáticamente a un cambio de estado para cualquier nodo vecino durante su secuencia de eventos.

35 Al hacerlo, cada nodo puede evaluar independientemente si un cambio en el estado de un nodo vecino afectará a su secuencia o estado y, si es así, cómo el estado cambiado puede afectar a su secuencia. Una vez determinado, cada nodo también puede cambiar su propia secuencia de eventos para asegurarse de que la tarea se ha completado, y preferiblemente de manera segura.

40 Opcionalmente, la etapa de establecer comunicación entre una pluralidad de nodos vecinos comprende además: enviar una solicitud de comunicación confirmatoria a la pluralidad de nodos vecinos; y recibir una o más respuestas a la solicitud de comunicación confirmatoria desde uno o más nodos vecinos que indican su presencia.

45 Antes de tomar cualquier decisión o cambio en la secuencia de cualquier nodo, es importante conocer qué nodos en el sistema están lo suficientemente cerca como para conectarse a o provocar interrupción. Esto también puede denominarse etapa de descubrimiento.

Opcionalmente, la etapa de establecer comunicación entre una pluralidad de nodos vecinos comprende además construir un mapa en tiempo real de la pluralidad de nodos vecinos.

50 En algunos casos, conocer la proximidad de los nodos o dispositivos circundantes de un sistema distribuido puede proporcionar una representación visual de la red en una determinada área y puede ser útil cuando se consideran nodos adicionales al sistema y/o cuando se planifica gestionar de manera eficiente los recursos del sistema.

55 Opcionalmente, un nodo puede detectar nodos externos que están en proximidad física, pero no forman parte de la red distribuida o la vecindad, y actualizar su propia secuencia de eventos (o actualizar el uno o más nodos externos) observando el comportamiento de tal nodo: como resultado, nodos como vehículos autónomos pueden atender situaciones híbridas donde están implicados vehículos no autónomos. Opcionalmente, los nodos externos están dentro de una proximidad geográfica de un nodo dentro de la red distribuida.

60 Opcionalmente, determinar una pluralidad de eventos interdependientes comprende crear árboles de dependencias de eventos de cada uno de la pluralidad de nodos vecinos. Opcionalmente, los árboles de dependencias de eventos se crean usando tiempo lógico.

65 El uso de tiempo lógico puede reducir el riesgo de retardos de tiempo de transmisión impredecibles que se producen en los sistemas tradicionales. Aunque se usa lógica basada en eventos, las mediciones tradicionales basadas en el tiempo pueden usarse internamente en un nodo pero no dentro/entre el resto del sistema.

Opcionalmente, se calcula el tiempo lógico basándose en la secuencia de eventos para cada uno de la pluralidad de nodos vecinos.

5 En algunos casos, el tiempo lógico se determina después de considerar la secuencia de eventos que deben producirse en orden para que se complete de manera segura una tarea particular. Alternativamente, el tiempo lógico puede determinarse en relación con una posición dentro de la secuencia de eventos.

10 Opcionalmente, la etapa de comunicar cada nodo su información de estado a los nodos vecinos comprende enviar una consulta de cambio de estado a los nodos vecinos. Opcionalmente, tras la recepción de la consulta de cambio de estado, los nodos vecinos determinan si la consulta de cambio de estado es conforme a su secuencia de eventos.

15 Cada nodo puede calcular si un cambio de estado de un nodo cercano daría como resultado una complicación o interrupción de su propio estado o secuencia y si detuviera al nodo de completar correctamente, y opcionalmente de manera segura, su tarea original.

20 Opcionalmente, la etapa de comunicar cada nodo su información de estado a los nodos vecinos comprende además procesar localmente la información de estado recibida desde nodos vecinos para determinar una nueva secuencia de eventos para que él mismo y nodos vecinos se adapten al cambio de solicitud de estado y opcionalmente enviar una solicitud para ejecutar las nuevas secuencias de eventos a nodos vecinos.

25 Esto también puede reducir la potencia de procesamiento necesaria para los componentes centralizados tradicionales puesto que las etapas de procesamiento pueden distribuirse entre los nodos o componentes en el sistema distribuido.

Opcionalmente, la nueva secuencia de eventos se determina basándose en una tasa de muestreo variable.

30 Por ejemplo, esto puede basarse en un vehículo que se desplaza durante una cierta cantidad de tiempo a una velocidad específica. Si se determina que este cambio, y las condiciones no son seguros, el método puede adaptarse actualizando/creando la secuencia de eventos con una nueva secuencia de eventos o deteniendo la actualización por completo.

35 Opcionalmente, cada nodo puede operarse para reaccionar a uno o más estados dependiendo de parámetros locales. Opcionalmente, los parámetros locales comprenden uno cualquiera o cualquier combinación de parámetros definidos por el usuario y/o parámetros predefinidos.

40 Los parámetros locales pueden ser varias propiedades tales como proximidad, velocidad, ubicación, o un requisito particular dependiendo del nodo. Los parámetros locales pueden usarse para crear un sistema de clasificación para determinar qué nodos podrían completar una tarea que se ha introducido en la red o una nueva solicitud de un nodo en la red. Estos datos se recopilan a menudo cuando se determinan los nodos vecinos y también pueden mejorar la asignación de tareas tradicional puesto que ya no es predictiva y/o se arbitra centralmente, pero se basa en condiciones locales de los componentes de la red.

45 Opcionalmente, cada nodo puede operarse para reaccionar a uno o más estados dependiendo de las restricciones. Opcionalmente, las restricciones comprenden cualquiera de una o cualquier combinación de restricciones condicionales, restricciones dependientes, restricciones exclusivas, restricciones de precedencia y/o restricciones de coincidencia, opcionalmente, las restricciones condicionales comprenden determinar cuándo se satisface una condición booleana.

50 Las restricciones de precedencia requieren que se produzca un evento después de un evento particular, mientras que las restricciones de coincidencia requieren que se produzcan eventos al mismo tiempo.

55 Las restricciones de dependencia pueden referirse a la circunstancia en la que un evento depende de otro evento. Como ejemplo, y específicamente para la industria del automóvil, no se produciría una solicitud de aumentar la velocidad de un vehículo si no se ha producido el evento de "arranque del motor". De manera similar, las restricciones exclusivas pueden referirse a asegurarse de que si tiene lugar un evento, entonces otro no puede. De nuevo, continuando con un ejemplo para la industria del automóvil, una solicitud de girar a la izquierda y a la derecha no pueden producirse al mismo tiempo.

60 Opcionalmente, el método funciona sin un sistema de sellos de tiempo.

65 Específicamente, cuando el método se implementa a través de todo el sistema distribuido, o globalmente a través del sistema distribuido, el método no depende de mediciones tradicionales basadas en el tiempo, por ejemplo, sellos de tiempo absolutos.

Opcionalmente, el envío y recepción de datos se procesan a través de canales de comunicación iguales y/o

independientes. Opcionalmente, el canal de comunicación comprende una cualquiera o cualquier combinación de conexiones inalámbricas o cableadas.

5 Opcionalmente, las secuencias de eventos comprenden una o más tareas que han de llevarse a cabo por cada nodo de la red distribuida.

10 Opcionalmente, cada uno de los nodos puede operarse para evaluar cambios de estado de cada uno de los nodos vecinos y determinar una secuencia de eventos que evite riesgos. Opcionalmente, los riesgos comprenden uno cualquiera o una combinación de riesgos físicos o lógicos.

Opcionalmente, la etapa de determinar uno o más estados de los nodos relevantes de la pluralidad de nodos vecinos comprende determinar un estado estacionario de cada uno de la pluralidad de nodos vecinos.

15 Puede proporcionarse un nodo transmisor que puede operarse para realizar una cualquiera o cualquier combinación de las etapas del método descritas anteriormente.

Puede proporcionarse un nodo receptor que puede operarse para realizar una cualquiera o cualquier combinación de las etapas del método descritas anteriormente.

20 Puede proporcionarse un sistema que comprende tal nodo transmisor según tal nodo receptor. Cada nodo en un sistema distribuido puede actuar como transmisor y/o receptor de datos dentro de un sistema distribuido. Por consiguiente, cada nodo también puede realizar sus propios datos de cálculo y de proceso.

25 Según un segundo aspecto, se proporciona un sistema según la reivindicación 14.

Según un tercer aspecto, se proporciona un programa informático según la reivindicación 15.

Breve descripción de los dibujos

30 A continuación, se describirán realizaciones, a modo de ejemplo únicamente y con referencia a los dibujos adjuntos que tienen números de referencia similares, en los que:

35 la figura 1 muestra una realización, en una fase de descubrimiento, en la que tres coches autónomos determinan de manera distribuida cómo permitir que uno de los coches se incorpore a un carril con otros coches autónomos;

la figura 2 muestra una realización, en una fase de intención, en la que tres coches autónomos determinan de manera distribuida cómo permitir que uno de los coches se incorpore a un carril con otros coches autónomos;

40 la figura 3 muestra una realización, en una fase de ejecución, en la que tres coches autónomos determinan de manera distribuida cómo permitir que uno de los coches se incorpore a un carril con otros coches autónomos;

la figura 4 muestra la tarea ejecutada de un coche que se incorpora a un carril con otros coches autónomos;

45 la figura 5 muestra una realización, en una fase de descubrimiento, en la que cuatro robots autónomos determinan de manera distribuida cómo acceder a una intersección entre corredores y evitar interbloqueos;

la figura 6 muestra una realización, en una fase de intención, en la que cuatro robots autónomos determinan de manera distribuida cómo acceder a una intersección entre corredores y evitar interbloqueos;

50 la figura 7 muestra una realización, en una fase de intención, en la que cuatro robots autónomos han comunicado sus intenciones entre sí y han identificado una sección crítica, que consiste en las ubicaciones CS1, CS2, CS3, CS4; y

55 la figura 8 muestra una realización, en una fase de ejecución, en la que dos de los robots se han movido hasta la intersección sin provocar un interbloqueo.

Descripción específica

60 En el artículo "Timed-pNets: Timed-pNets: a communication behavioural semantic model for distributed systems" publicado en *Frontiers of Computer Science* - febrero de 2015 (*Front. Comput. Sci.*, 2015, 9(1): 87-110; DOI 10.1007/s11704-014-4096-4) se propuso un enfoque matemático para construir un modelo semántico de comportamiento de comunicación para sistemas distribuidos heterogéneos que incluyen comunicaciones síncronas y asíncronas. Este artículo proporciona una descripción matemática de cómo construir dependencias de reloj basadas en eventos.

65 Basándose en la base matemática expuesta en este artículo, el siguiente ejemplo describe una realización de la

invención a través de una red distribuida:

En una red distribuida, varios nodos conectados (por ejemplo, a, b, c...) son capaces de comunicarse entre sí a través de un canal directo y/o indirecto. Puede usarse cualquier forma conocida de comunicación por cable y/o inalámbrica para permitir la comunicación entre nodos en la misma red distribuida. En algunas realizaciones, puede considerarse que los nodos dentro de una proximidad cercana o los nodos que afectan a una secuencia de tareas de uno o más de otros nodos dentro de una red distribuida, están dentro de una vecindad de nodos (vecinos). En algunas realizaciones, los nodos vecinos pueden determinarse de manera dinámica y/o considerarse como un subconjunto de nodos dentro de una red distribuida. Como ejemplo, puede considerarse que todos los robots autónomos dentro de un almacén son una red distribuida (donde cada robot es un nodo de la red) mientras que puede considerarse que un grupo o subconjunto más pequeño de estos robots es una vecindad de nodos de la red distribuida. En algunas realizaciones, puede operarse cada nodo de la red distribuida para determinar su propia vecindad de nodos, en lugar de una determinación global de nodos vecinos para cada nodo de la red distribuida. Por ejemplo, si dos nodos de una red distribuida el "nodo a" y el "nodo b" están cerca uno del otro, es probable que estén en la vecindad de nodos vecinos del otro, pero cada nodo puede tener diferentes nodos dentro de sus vecindades respectivas (por ejemplo, nodos vecinos del "nodo a": nodos b, e, l, r, c; y nodos vecinos del "nodo b": nodos a, t, n, f). En algunas realizaciones, una vecindad de nodos puede incluir nodos de más de una red distribuida.

En realizaciones de ejemplo, se monitorizan habitualmente los estados actuales de los nodos y se basan en un consenso definido previamente. Cuando el estado está cambiando según el consenso del sistema o la red distribuida, habitualmente se hace referencia a que está en un estado estacionario. En algunas realizaciones, un estado estacionario es una condición que existe cuando todos los nodos en una vecindad han alcanzado un consenso (acordado sobre qué nodo debe estar realizando qué tarea/acción y cuándo). Como ejemplo, si un nuevo nodo entrara o interfiriera con la vecindad e iniciara una nueva "fase de intención", toda la vecindad deja de estar en una condición de estado estacionario. En algunas realizaciones, los cambios de estado corresponden a acciones que se realizan por el nodo (por ejemplo, "MOVING_FORWARD" (avanzar) "TURNING_LEFT" (girar a la izquierda), etc.). Por tanto, cuando un nodo dentro de una vecindad de nodos está realizando diversas acciones de una secuencia y cambios de estados de una manera esperada, basándose en un consenso predefinido, se considera que está operando en un estado estacionario. En algunas realizaciones, las acciones llevadas a cabo por cada nodo de una red pueden depender del contexto puesto que se basan en las capacidades y/o la función de un nodo específico. Por ejemplo, cuando un nodo está acoplado a un componente dentro de un almacén robótico, el nodo podría realizar una acción "LIFT_SHELF" (elear un estante); sin embargo, este tipo de acción no sería relevante en el contexto de vehículos autónomos.

A lo largo de la red, cada nodo envía periódicamente un latido a sus nodos vecinos que permite que cada nodo respectivo monitoree los estados respectivos de los nodos vecinos. En algunas realizaciones, el estado de un nodo corresponde a un conjunto de parámetros internos para el nodo, tales como velocidad, dirección, ubicación, proximidad, etc. En algunas realizaciones, el latido puede permitir la construcción de un mapa en tiempo real de los nodos circundantes. En algunas realizaciones, los mensajes de latido se envían por cada nodo de manera continua, a través de todas las etapas del proceso y, de esta manera, cada nodo permite determinar si se cumplen todas las condiciones para un cambio de estado (por ejemplo, comenzar a ejecutar una nueva tarea/acción). Como ejemplo, un mensaje de latido puede contener un identificador de nodo y un conjunto de parámetros que son relevantes para la aplicación específica. Como ejemplo, en el contexto de algunos sistemas físicos virtuales, estos parámetros pueden incluir la posición actual, la última posición conocida y la rotación del nodo físico a través de un eje tridimensional.

Si uno o más nodos tienen que completar una nueva acción o una secuencia de acciones/eventos (que alterarán finalmente el estado estacionario de sus nodos vecinos), el uno o más nodos enviarán una comunicación de intención a los nodos vecinos que proporciona detalles sobre la acción o secuencias de acciones/eventos pretendidas. La comunicación de intención puede incluir un conjunto arbitrario de parámetros y restricciones. En algunas realizaciones, un nodo no conoce necesariamente que afectará al estado estacionario de un nodo vecino sino que, tan pronto como el nodo entra en su fase de intención para determinar si sus nodos vecinos pueden ejecutar una cierta acción o secuencias de acciones/eventos, se necesita un nuevo consenso y, por tanto, la vecindad ya no está en un estado estacionario.

Una vez que los nodos vecinos reciben esta información de carga útil desde el nodo (o nodos) de origen, evalúan el impacto que tendrá el cambio en su propio estado y sus estados de nodos vecinos. En algunas realizaciones, la comunicación de intención (información de carga útil) de las fases de intención incluye una descripción de la acción o secuencias de acciones/eventos que se requiere que lleve a cabo el nodo. Esta descripción se usa entonces por sus vecinos (destinatarios de la comunicación de intención) para determinar si puede ejecutar la acción o secuencias de acciones/eventos, y el impacto en sus propios estados y de nodos vecinos. En realizaciones de ejemplo, en la fase de intención, los nodos pueden intercambiar mensajes para alcanzar un consenso. Basándose en las evaluaciones por los nodos vecinos, si se considera seguro que prosiga la comunicación de intención (por ejemplo, sin interbloqueos, condiciones de carrera, condiciones inseguras, etc.) el/los nodo(s) vecino(s) enviará(n) de vuelta un consenso positivo al/a los nodo(s) de origen de intención, y opcionalmente a los otros nodos vecinos. Sin embargo, si se considera insegura la intención y su efecto en el nodo (o nodos) vecino(s), se rechazará el mensaje

de "intención". En algunas realizaciones, algunos nodos pueden considerar seguro ejecutar la intención y algunos nodos pueden considerar inseguro ejecutar la intención. Es necesario un consenso positivo cada vez que un nodo que forma parte de una red distribuida, vecina o lo suficientemente cercana como para afectar a una red distribuida, desea entrar en un nuevo estado (es decir, ejecutar una nueva acción) porque eso puede tener un impacto potencial en las secuencias de eventos que se mantienen en el interior de cada uno de los nodos vecinos. En algunas realizaciones, la secuencia de acciones de uno o más nodos podría depender de una secuencia de acciones de otro nodo vecino, por lo que es necesario reconocer un consenso positivo antes de alterar los estados de sus nodos vecinos.

Como ejemplo de evaluaciones de seguridad realizadas por los nodos, los nodos pueden determinar la probabilidad de un escenario de interbloqueo. Pueden producirse interbloqueos cuando hay una dependencia circular entre adquisiciones de recursos exclusivas realizadas por dos o más hilos o nodos concurrentes. Normalmente, es necesario que haya más de un hilo o nodo de ejecución concurrente para que se produzca un interbloqueo. Cuando estos hilos concurrentes intentan obtener acceso exclusivo a algún recurso, da como resultado un interbloqueo. A menudo se requiere acceso exclusivo a un recurso para la exactitud en programas concurrentes para impedir que múltiples hilos o nodos interfieran entre sí, pero es el requisito de exclusividad lo que puede conducir a un interbloqueo. Para que un interbloqueo se produzca realmente, hay habitualmente una dependencia circular: un anillo de hilos o nodos, cada uno de los cuales ha adquirido acceso a algún recurso exclusivo, pero desea el acceso a otro recurso. Debido a la circularidad, cada hilo está esperando finalmente a sí mismo que libere su recurso para proseguir. Los escenarios de interbloqueo comunes incluyen cuatro vehículos en una intersección de una carretera de doble vía, o el bien conocido "problema de la cena de los filósofos".

Otra evaluación que pueden realizar los nodos se conoce como condiciones de carrera, o peligros de carrera. Una condición de carrera es cuando la salida de un sistema o red depende de la temporización o secuencia de un evento incontrolable. Esto es muy frecuente en entornos de múltiples hilos y distribuidos. Las condiciones de carrera pueden producirse cuando dos (o más) nodos que comparten el mismo espacio de memoria quieren actualizar una variable total que ha completado sus eventos independientes sustancialmente al mismo tiempo. Como ejemplo, cuando dos hilos o nodos quieren completar una tarea, ambos nodos leerán en primer lugar la variable total inicial. Una vez que los dos nodos llevan a cabo sus tareas, realizarán operaciones sobre la variable total inicial en función del resultado de sus tareas y luego intentarán simultáneamente actualizar la variable total inicial con sus totales actualizados respectivos. Sin embargo, como la variable total se comparte entre los dos nodos, sólo el valor del nodo que envía su variable total actualizada en último lugar se conserva en la memoria compartida puesto que sobrescribirá cualquier total anterior, incluyendo una variable total actualizada del otro nodo. Así, en condiciones de carrera, no puede garantizarse la salida puesto que depende de qué hilo o nodo finalice su tarea y envíe su último total actualizado.

Las evaluaciones realizadas por los nodos también pueden ayudar a determinar qué nodo o actor en la red es el más adecuado para ejecutar una tarea o secuencia de eventos. Esto puede determinarse basándose en la eficiencia y/o la seguridad. Por ejemplo, en un almacén robótico varios robots (representados como nodos dentro de una red distribuida) pueden llevar a cabo tareas tales como transportar objetos desde una ubicación a otra, mediante un cierto plazo (por ejemplo, dentro de un intervalo de tiempo tal como de 5 minutos, 10 minutos, etc.). Tras evaluar las opciones, los nodos pueden seleccionar el primer nodo para proseguir con su transición de estado (es decir, la ejecución de una secuencia de acciones/eventos establecidos) basándose en la proximidad temporal a su plazo. Alternativamente, si la vecindad está enfrentándose a un riesgo de una congestión dentro del almacén, los nodos podrían seleccionar el nodo que está en un estado que aparece más a menudo en la lista de dependencias de otros nodos (es decir, bloquea el número más alto de nodos) para proseguir. Adicionalmente, de esta manera, la información de procesamiento se distribuye dentro de la red y no se lleva a cabo en un componente de sistema central (tal como un servidor o un controlador centralizado de software o hardware, por ejemplo, un nodo maestro central en una red en malla).

Una vez que el/los nodo(s) de origen recibe respuestas de los nodos vecinos, la acción se lleva a cabo por un nodo que se determina que es responsable de la primera acción o evento en la secuencia, siempre que no haya dependencia precedente para ese nodo de otra acción o acciones (por ejemplo, la acción puede llevarse a cabo independientemente o en paralelo a una o más de otras acciones en uno o más de otros nodos). Si hay una dependencia precedente, que puede hacer que la nueva acción, evento o secuencia sea inseguro, se abortará la intención original, y se descartará la secuencia. En algunas realizaciones, la secuencia puede descartarse o abortarse basándose en un mensaje de consenso de uno o más nodos que deniegan el consentimiento para la transición de estado en respuesta al mensaje de intención inicial.

Basándose en el mensaje de intención y las nuevas acciones/eventos, en el caso de que los nodos entren en un estado de ejecución, los nodos en ejecución monitorizarán sus estados para garantizar que las acciones o secuencia de eventos están llevándose a cabo en la secuencia correcta y según se acuerda para mantener la integridad y seguridad del sistema. Una vez que un nodo completa su evento en la secuencia, notificará a sus vecinos que lo ha completado y su estado actual. Esto activará el siguiente evento en la secuencia si está disponible. Una vez que todos los nodos implicados han completado la etapa de ejecución, el sistema vuelve a entrar en un estado estacionario.

En algunos casos, pueden insertarse nuevos mensajes de intención prioritaria en una red para ejecutar una acción o secuencia de eventos antes de una intención emitida previamente, basándose en cualquier parámetro arbitrario definido por el sistema/red. En otros casos, siempre que se envía un nuevo mensaje de intención al sistema, antes de que la acción pueda ejecutarse de manera segura, se considera la seguridad del nuevo evento y la nueva intención sólo se ejecutará siempre que sea seguro hacerlo.

Haciendo referencia ahora a las figuras 1 a 4, se describirá ahora una realización de ejemplo en relación con el uso de este enfoque en vehículos autónomos que proporciona un método para coordinar tareas en una red/malla distribuida basándose en parámetros y restricciones, que proporciona una disposición de mensajería compleja que permite que las máquinas en una red distribuida se comuniquen entre sí para garantizar que las tareas se completen de manera eficaz.

En las figuras 1 a 4, la realización de ejemplo muestra una situación que implica tres coches C1, C2, C3 autónomos que circulan a lo largo de una carretera de doble calzada (es decir, una carretera con dos carriles, teniendo ambos carriles tráfico que va en el mismo sentido). El primer coche C1 circula a lo largo del primer carril y pretende incorporarse al otro carril de la carretera. El segundo coche C2 circula a lo largo del segundo carril (al que pretende incorporarse el primer coche C1). El tercer coche C3 circula a lo largo del segundo carril detrás del segundo coche C2.

Con referencia a la figura 1, el primer coche C1 está en una fase de "descubrimiento" e intenta establecer una conexión con cualquier vehículo vecino. En algunas realizaciones, el árbol de dependencias basado en eventos se construye de manera colaborativa por los nodos de una red distribuida. En algunas realizaciones, la vecindad de los nodos puede determinarse por los nodos dentro de una cierta área o basarse en interdependencias (por ejemplo, nodos que son relevantes) para llevar a cabo una o más secuencias de acciones. En este ejemplo, los vehículos relevantes son el segundo coche C2 y el tercer coche C3. El primer coche C1 hace esto emitiendo un comando 130 de "latido" a través de un canal de comunicación común a cualquier vehículo cercano y en este ejemplo recibirá una respuesta del segundo coche C2 y el tercer coche C3 de vuelta a través del mismo canal de comunicación. Preferiblemente, las conexiones se establecen para obtener un estado actual de los otros vehículos antes de tomar cualquier decisión en cualquier nodo.

Con referencia a la figura 2, el primer coche C1 entra en la fase de "intención" y envía una solicitud 240 a través del canal de comunicaciones al segundo y tercer coches C2, C3 para entrar entre el segundo y tercer coches C2, C3 en su carril de la carretera 220. El segundo y tercer coches C2, C3 comunican a su vez hacia adelante la solicitud del primer coche C1 con los vehículos vecinos del segundo y tercer coches C2, C3, respectivamente, para garantizar que cada uno del segundo y tercer coches C2, C3 pueda completar las tareas requeridas de manera segura. Si el segundo y tercer coches C2, C3 están de acuerdo (así como cualquier coche vecino al segundo y tercer coches C2, C3) en que la tarea se considera segura (un consenso positivo), entonces puede comenzar la fase de ejecución.

Con referencia a la figura 3, el primer coche C1 entra en la fase de "ejecución" y solicita el estado actual (por ejemplo, velocidad, dirección, ubicación, etc.) del segundo y tercer coches C2, C3 a través del canal de comunicaciones. El primer coche C1 computa entonces, usando los estados adquiridos del segundo y tercer coches C2, C3, una determinación de la línea de acción correcta que es necesario completarse de manera segura con respecto a la tarea o tareas requeridas. En este ejemplo, el primer coche C1 determina que el segundo coche C2 debe acelerar en 5 millas por hora (mph) durante 30 segundos y el tercer coche C3 debe desacelerar en 5 mph durante 30 segundos para crear un punto de entrada seguro de una distancia de tres veces la longitud del primer coche C1 entre el segundo y tercer coches C2, C3 para que el primer coche C1 se desplace al segundo carril 320 de la carretera entre el segundo y tercer coches C2, C3. El primer coche C1 comunica entonces estas solicitudes determinadas de aceleración por el segundo coche C2 y de desaceleración por el tercer coche C3 a través del canal de comunicaciones. El segundo y tercer coches C2, C3 reciben entonces estas solicitudes y confirman con cualquiera de sus coches vecinos respectivos que garanticen que las solicitudes mantengan la seguridad. Una vez que se han realizado todas las comprobaciones con cualquier coche vecino y el segundo y tercer coches C2, C3, el segundo y tercer coches C2, C3 aprueban la fase de ejecución y se realizan las tareas distribuidas por los vehículos C1, C2, C3 respectivos hasta que los actores C1, C2, C3 participantes logran las posiciones ideales (relativas) calculadas. Durante la ejecución, cada nodo C1, C2, C3 muestrea el estado de sí mismo y cualquier nodo C1, C2, C3 vecino a una tasa de muestreo variable y puede activarse en cualquier inflexión (por ejemplo, si el estado estacionario del sistema se rompe, uno de los coches C1, C2, C3 frena repentinamente o uno de los coches C1, C2, C3 gira repentinamente).

Una vez que se ha completado la tarea, tal como se representa en la figura 4, el ciclo trifásico se repite en sí mismo de manera continua. Dicho de otro modo, una vez que se ha completado la tarea, el sistema/red entra en un nuevo estado estacionario en el que se envían latidos desde cada nodo de manera continua a nodos vecinos en el sistema/red. El ciclo completo se reinicia si hay un cambio en el estado estacionario o se propaga una nueva intención. Opcionalmente, al final de cada ciclo, puede determinarse la seguridad de la tarea que va a completarse.

Haciendo referencia ahora a las figuras 5 a 8, se describirá a continuación otra realización en relación con el uso de

este enfoque en almacenes robóticos.

En las figuras 5 a 8, la realización de ejemplo muestra una situación que implica cuatro robots R1, R2, R3, R4 autónomos (es decir, nodos), que se desplazan a lo largo de diferentes corredores de un almacén. En este ejemplo, cada robot se desplaza hacia la intersección de los cuatro corredores sin conocer la dirección de los otros robots, sus destinos o secuencias de acciones pretendidos.

Con referencia a la figura 5, a medida que cada robot R1, R2, R3, R4 se aproxima a la intersección, están en una fase de "descubrimiento" e intentan establecer una conexión con cualquier robot vecino. En este ejemplo, los robots relevantes son R1, R2, R3 y R4. Cada robot hace esto emitiendo un comando 510a, 510b, 510c, 510d de "latido", a través de un canal de comunicación común a cualquier robot cercano. Por ejemplo, tal como se mencionó anteriormente, la comunicación/comando de latido puede incluir el estado de los nodos. En este ejemplo, cada robot recibirá una respuesta de vuelta desde cada uno de los otros robots a través del mismo canal de comunicación.

A continuación, con referencia a la figura 6, cada robot entra en la fase de "intención" y envía una solicitud 610 (que indica la acción o secuencia de acciones pretendida), a los robots vecinos. Como ejemplo, la figura 6 también muestra un sistema de ejes cartesianos X/Y que los robots podrían usar para expresar sus intenciones o ubicaciones. El robot R1 declara que desea desplazarse según el eje X en sentido positivo (es decir, "MOVE_EAST" (desplazarse al este)) en una distancia d_{r11} (hasta que llega a la pared 620 del almacén), entonces quiere desplazarse a lo largo del eje Y en un sentido negativo (es decir, "MOVE_SOUTH" (desplazarse al sur)) en una distancia d_{r12} (hasta la pared 621 del almacén), entonces quiere "MOVE_EAST" en la distancia d_{r13} (hasta que la pared 622 ya no sea una obstrucción en su sentido norte) y finalmente quiere desplazarse a lo largo del eje Y en un sentido positivo (es decir, "MOVE_NORTH" (desplazarse al norte)) en una distancia d_{r14} . De la misma manera, el robot R2 declara que quiere "MOVE_SOUTH" hasta la pared 621, luego "MOVE_EAST" hasta que la pared 621 ya no sea una obstrucción en su sentido sur y, finalmente, "MOVE_SOUTH". El robot R3 declara que desea a lo largo del eje X en un sentido negativo (es decir, "MOVE_WEST" (desplazarse al oeste)) hasta que llega a la pared 623, entonces quiere "MOVE_NORTH" hasta que llega a la pared 624 y, finalmente, quiere "MOVE_WEST". El robot R4 declara que desea "MOVE_NORTH" hasta que llega a la pared 624, luego "MOVE_EAST" hasta que llega a la pared 620 y, finalmente, "MOVE_NORTH".

Con referencia a la figura 7, los robots han calculado ahora una lista común de eventos y todas las dependencias (por ejemplo, todos los eventos interdependientes entre todos los nodos). Por ejemplo, cada robot conoce a dónde pretende desplazarse cada robot R1', R2', R3', R4'. Adicionalmente, la figura 7 muestra cuatro ubicaciones de la intersección CS1, CS2, CS3, CS4 que representan las secciones críticas de la intersección o de las trayectorias. Esta es el área donde la sincronización es crucial para evitar los interbloqueos. Dadas las coordenadas iniciales de los robots, el robot R1 determina que el robot R3 debe estar en una coordenada X que sea menor que la coordenada X de CS4 (es decir, $R3.x < CS4.x$) y en una coordenada Y que sea menor que la coordenada Y de R1 (es decir, $R3.y < R1.y$), para poder llevar a cabo su secuencia de eventos. R2 también determina que debe aplicarse $R3.x < CS4.x$ para poder llevar a cabo su secuencia de eventos. R3 determina que debe aplicarse $R1.x > CS1.x$ para poder llevar a cabo su secuencia de eventos. Finalmente, R4 determina que debe aplicarse $R2.y < CS2.y$ para poder llevar a cabo su secuencia de eventos. Tal como se mencionó anteriormente, los nodos pueden alcanzar consenso sobre el ordenamiento de eventos basándose en qué nodo está bloqueando el número más alto de nodos vecinos. En este caso, tanto R1 como R2 dependen de R3, por lo que dan su consentimiento a que R3 se desplace a la ubicación CS3, que es la ubicación más cercana que satisface ambas restricciones $R3.x < CS4.x$ y $R3.y < R1.y$. Habiendo R3 entrado en la fase de "ejecución" y desplazado a CS3, R1 solicita el permiso para desplazarse a CS2. R2 podría denegarlo potencialmente, sin embargo, dado que los nodos están usando un criterio que está dirigido a minimizar el número de nodos en el interior de la sección crítica al mismo tiempo y dado que R1 debe entrar en la sección crítica para que R3 pueda salir de ella, R2 proporciona su consentimiento y R1 se desplaza a CS2.

Con referencia a la figura 8, los robots/nodos están ahora en la fase de "ejecución" y tanto R3 como R1 se han desplazado al interior de la sección crítica. En este punto, debido a que la vecindad intentará minimizar el número de nodos que acceden simultáneamente a la sección crítica, cada vez que R1 y R3 vuelvan a entrar en la fase de "intención" para pedir a sus vecinos si pueden proseguir con la siguiente transición de estado, tanto R2 como R4 proporcionarán su consentimiento. Una vez que tanto R1 como R3 están fuera de la sección crítica, R2 y R4 llevarán a cabo los ciclos de intención/ejecución restantes siguiendo las restricciones definidas previamente. Tal como se mencionó anteriormente, también puede implementarse un sistema de intención prioritaria. Por ejemplo, en algunas realizaciones, un mensaje de intención prioritaria podría usarse por ambulancias, vehículos de policía, camiones de bomberos, etc., lo que les daría más importancia sobre los vehículos civiles para llevar a cabo una secuencia de eventos o tareas. Dicho de otro modo, una nueva acción por parte de la unidad de primera intervención o un vehículo de emergencia sustituiría las acciones o eventos solicitados por otros vehículos civiles.

En realizaciones, se calcula el tiempo lógico en secuencias basadas en eventos. Generalmente, cada evento tiene dos acciones instantáneas: una acción de inicio y una acción de fin. Por ejemplo, un frenado de coche puede modelarse como una acción de "frenado de inicio" y una acción de "frenado de fin" y la condición requerida para activar la acción de "frenado de fin" podría ser la cuando la velocidad del coche disminuye hasta cero.

Para otros eventos, por ejemplo, tener un evento en espera, puede usarse un reloj local (medición tradicional basada en el tiempo) para determinar la duración de un evento. Sin embargo, el uso de un reloj local basado en el tiempo no determina ni influye en las restricciones lógicas del reloj a través de los diferentes dispositivos en el sistema distribuido, o el sistema distribuido en su conjunto. En contraposición al uso de la programación basada en el tiempo absoluto, que puede ser difícil de sincronizar y puede suplantarse, esta realización usa la programación basada en eventos.

Otras aplicaciones para esta invención pueden incluir cámaras múltiples coordinadas/configuración de cámaras múltiples, dispositivos IoT o fábricas modulares.

Como ejemplo, en configuraciones de cámaras múltiples para películas y TV, en lugar de depender de información de sellos de tiempo de cada cámara para sincronizar los fotogramas de vídeo y/o los datos de audio en la edición, pueden convertirse activadores en la salida digital creada por la cámara en intenciones o eventos, por ejemplo, una variación en el perfil de audio, un cambio en el perfil de color, un cambio en la composición de la escena, luminancia de la escena, etc.

Estos eventos pueden usarse entonces para crear una secuencia lógica que puede usarse para sincronizar las líneas de tiempo de las cámaras múltiples y, en algunos casos, incluso determinar automáticamente la desviación antes de alimentar los datos al software de edición.

De manera similar, las fábricas están desarrollándose en el concepto de fábricas modulares. Normalmente, las fábricas tradicionales son muy eficientes en salidas fijas y ocasionalmente pueden reducir en rampa la producción cuando no hay cambios en el proceso. Sin embargo, estas fábricas tradicionales no pueden hacer frente bien a un aumento repentino de la demanda (por ejemplo, para satisfacer celebraciones comerciales tales como el día de la madre) ni pueden hacer frente bien a una ejecución limitada de un producto que se desvía del proceso predeterminado de la fábrica.

Las fábricas modulares proporcionan una solución a este problema al permitir que las fábricas construyan configuraciones modulares que pueden ampliarse a escala bajo demanda e introducir nuevas etapas o componentes cuando sea necesario.

Las realizaciones dadas a conocer en el presente documento también pueden usarse con fábricas modulares para permitirles trabajar muy fácilmente y en un verdadero paradigma de "enchufar y usar". Se definiría una intención de entrega y adquisición en el sistema y cada unidad de hardware de fábrica tendría la programada. Por tanto, cuando se introduce una nueva unidad o componente en el proceso, establecen una conexión controlada por secuencias a sus vecinos y pueden funcionar inmediatamente de manera conjunta sin ningún riesgo de error.

En las realizaciones y aspectos descritos en el presente documento, el hardware de comunicaciones usado para proporcionar comunicaciones entre las máquinas/sistemas/vehículos puede ser un módulo DSRC (comunicación dedicada de corto alcance) que puede proporcionar la ventaja de ser dúplex completo y puede proporcionar la ventaja de que, como la frecuencia a la que estos módulos operan normalmente está protegida normalmente por la normativa gubernamental de otros protocolos de comunicaciones tales como la interconexión en red inalámbrica de consumidores, hay menos probabilidad de interferencia o falta de ancho de banda debido a la congestión de tráfico en ese canal de comunicación. Opcionalmente, el uso de cualquier medio de comunicación por cable y/o inalámbrico puede usarse para transferir datos dentro de una red distribuida. En algunas realizaciones, los nodos pueden incluir uno o más sensores para determinar uno o más nodos dentro de una cierta proximidad. En algunas realizaciones, estos nodos adicionales, o externos, pueden considerarse actores externos que no dependen de o afectan a las secuencias de cualquier nodo dentro de una red distribuida o una vecindad de nodos. En alguna realización, los nodos pueden incluir sensores o transmisores para propagar una o más secuencias de acciones o estados a otros nodos dentro de la red distribuida (y vecindad de nodos), o a actores externos (nodos externos) dentro de una proximidad cercana (por ejemplo, una proximidad geográfica).

Cualquier característica del sistema tal como se describe en el presente documento también puede proporcionarse como una característica del método, y viceversa. Tal como se usa en el presente documento, los medios más las características de función pueden expresarse alternativamente en cuanto a su estructura correspondiente.

Cualquier característica en un aspecto puede aplicarse a otros aspectos, en cualquier combinación apropiada. En particular, los aspectos del método pueden aplicarse a aspectos del sistema, y viceversa. Además, cualquiera, algunas y/o todas las características en un aspecto pueden aplicarse a cualquiera, algunas y/o todas las características en cualquier otro aspecto, en cualquier combinación apropiada.

También debe apreciarse que combinaciones particulares de las diversas características descritas y definidas en cualquier aspecto pueden implementarse y/o suministrarse y/o usarse independientemente.

REIVINDICACIONES

- 5 1. Método de sincronización de estados basada en eventos para un sistema distribuido de nodos (C1-C3; R1-R4) físicos móviles, computando cada nodo (C1-C3; R1-R4) su propia secuencia de eventos, comprendiendo el método de computar la secuencia de eventos en cada nodo (C1-C3; R1-R4) del sistema distribuido las siguientes etapas:

establecer comunicación (130; 510a-510d) con una pluralidad de nodos vecinos de los nodos;

10 determinar (240; 600) cada nodo (C1-C3; R1-R4) una pluralidad de eventos interdependientes entre nodos (C1-C3; R1-R4) relevantes de la pluralidad de nodos (C1-C3; R1-R4) vecinos y concordancia con los eventos interdependientes, estando cada uno de los eventos interdependientes relacionado con el movimiento de nodos; y

15 comunicar cada nodo (C1-C3; R1-R4) su información de estado a sus nodos vecinos de tal manera que cada uno de sus nodos vecinos pueda monitorizar uno o más estados de los nodos relevantes de sus nodos vecinos durante la ejecución de los eventos;

20 en el que cada nodo (C1-C3; R1-R4) puede operarse para reaccionar al uno o más estados de los nodos relevantes de su pluralidad de nodos vecinos.
- 25 2. Método según la reivindicación 1, en el que la sincronización basada en eventos comprende coordinar eventos para la pluralidad de nodos (C1-C3; R1-R4) vecinos para adaptarse automáticamente a un cambio de estado para cualquier nodo vecino durante su secuencia de eventos.
- 30 3. Método según cualquier reivindicación anterior, en el que la etapa de establecer comunicación con la pluralidad de nodos vecinos comprende además construir un mapa en tiempo real de la pluralidad de nodos vecinos.
- 35 4. Método según cualquier reivindicación anterior, en el que cada nodo (C1-C3; R1-R4) del sistema distribuido o la pluralidad de nodos vecinos puede detectar uno o más nodos externos relevantes, en el que los nodos externos no forman parte del sistema distribuido de nodos (C1-C3; R1-R4) o la pluralidad de nodos vecinos, y actualizar su propia y/o la secuencia de eventos de uno o más nodos externos relevantes observando el comportamiento de dichos uno o más nodos externos relevantes.
- 40 5. Método según cualquier reivindicación anterior, en el que la determinación de la pluralidad de eventos interdependientes comprende crear un árbol de dependencias de eventos para cada uno de los nodos (C1-C3; R1-R4) vecinos.
- 45 6. Método según la reivindicación 5, en el que los árboles de dependencias de eventos se crean usando tiempo lógico.
- 50 7. Método según cualquier reivindicación anterior, en el que la etapa de comunicar cada nodo (C1-C3; R1-R4) su información de estado a los nodos vecinos comprende enviar una consulta de cambio de estado a los nodos vecinos, en el que, tras la recepción de la consulta de cambio de estado, los nodos vecinos determinan si la consulta de cambio de estado es conforme a su secuencia de eventos.
- 55 8. Método según la reivindicación 7, en el que la etapa de comunicar cada nodo (C1-C3; R1-R4) su información de estado a los nodos vecinos comprende además procesar localmente la información de estado recibida desde nodos vecinos para determinar una nueva secuencia de eventos para sí mismo y para los nodos vecinos para adaptarse al cambio de consulta de estado.
- 60 9. Método según la reivindicación 8, que comprende además enviar una solicitud para ejecutar las nuevas secuencias de eventos a nodos vecinos.
- 65 10. Método según cualquier reivindicación anterior, en el que cada nodo (C1-C3; R1-R4) puede operarse para reaccionar al uno o más estados que dependen de parámetros locales, en el que los parámetros locales comprenden uno o más de: ubicación, velocidad, dirección y proximidad.
11. Método según cualquier reivindicación anterior, en el que el método funciona sin un sistema de sellos de tiempo.
12. Método según cualquier reivindicación anterior, en el que cada nodo (C1-C3; R1-R4) puede operarse para evaluar cambios de estado de cada uno de los nodos vecinos y determinar una secuencia de eventos que evite riesgos, en el que los riesgos comprenden uno cualquiera o una combinación de riesgos físicos o lógicos.

13. Método según una cualquiera de las reivindicaciones anteriores, en el que los nodos vecinos de cada nodo pueden ser diferentes, en el que establecer comunicación comprende que cada nodo determine de manera dinámica sus nodos vecinos.
- 5
14. Sistema que comprende medios para llevar a cabo un método de sincronización de estados basada en eventos para un sistema distribuido de nodos (C1-C3; R1-R4) físicos móviles, computando cada nodo (C1-C3; R1-R4) su propia secuencia de eventos, comprendiendo el método de computar la secuencia de eventos en cada nodo del sistema distribuido las siguientes etapas:
- 10
- establecer comunicación con una pluralidad de nodos vecinos a los nodos (C1-C3; R1-R4);
- determinar cada nodo (C1-C3; R1-R4) una pluralidad de eventos interdependientes entre nodos relevantes de la pluralidad de nodos vecinos y concordancia con los eventos interdependientes, estando cada uno de los eventos interdependientes relacionado con el movimiento de los nodos; y
- 15
- comunicar cada nodo (C1-C3; R1-R4) su información de estado a sus nodos vecinos de tal manera que cada uno de sus nodos vecinos pueda monitorizar uno o más estados de los nodos relevantes de sus nodos vecinos durante la ejecución de los eventos;
- 20
- en el que cada nodo (C1-C3; R1-R4) puede operarse para reaccionar al uno o más estados de los nodos relevantes de su pluralidad de nodos vecinos.
15. Programa informático que comprende instrucciones que, cuando se ejecuta el programa por un ordenador, hacen que el ordenador lleve a cabo un método de sincronización de estados basada en eventos para un sistema distribuido de nodos (C1-C3; R1-R4) móviles físicos, computando cada nodo (C1-C3; R1-R4) su propia secuencia de eventos, comprendiendo el método de computar la secuencia de eventos en cada nodo (C1-C3; R1-R4) del sistema distribuido las siguientes etapas:
- 25
- establecer comunicación con una pluralidad de nodos vecinos a los nodos;
- 30
- determinar cada nodo (C1-C3; R1-R4) una pluralidad de eventos interdependientes entre nodos relevantes de la pluralidad de nodos vecinos y concordancia con los eventos interdependientes, estando cada uno de los eventos interdependientes relacionado con el movimiento de los nodos; y
- 35
- comunicar cada nodo su información de estado a sus nodos vecinos de tal manera que cada uno de sus nodos vecinos pueda monitorizar uno o más estados de los nodos relevantes de sus nodos vecinos durante la ejecución de los eventos;
- 40
- en el que cada nodo puede operarse para reaccionar al uno o más estados de los nodos relevantes de su pluralidad de nodos vecinos.

100

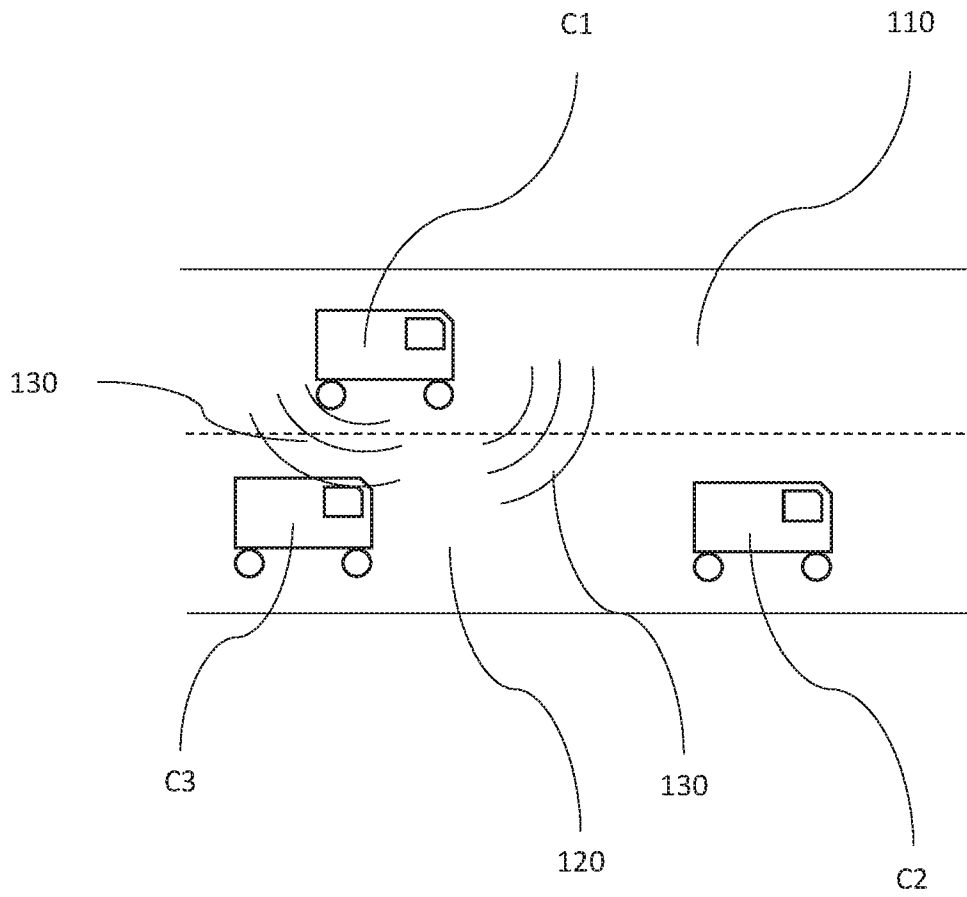


Fig. 1

200

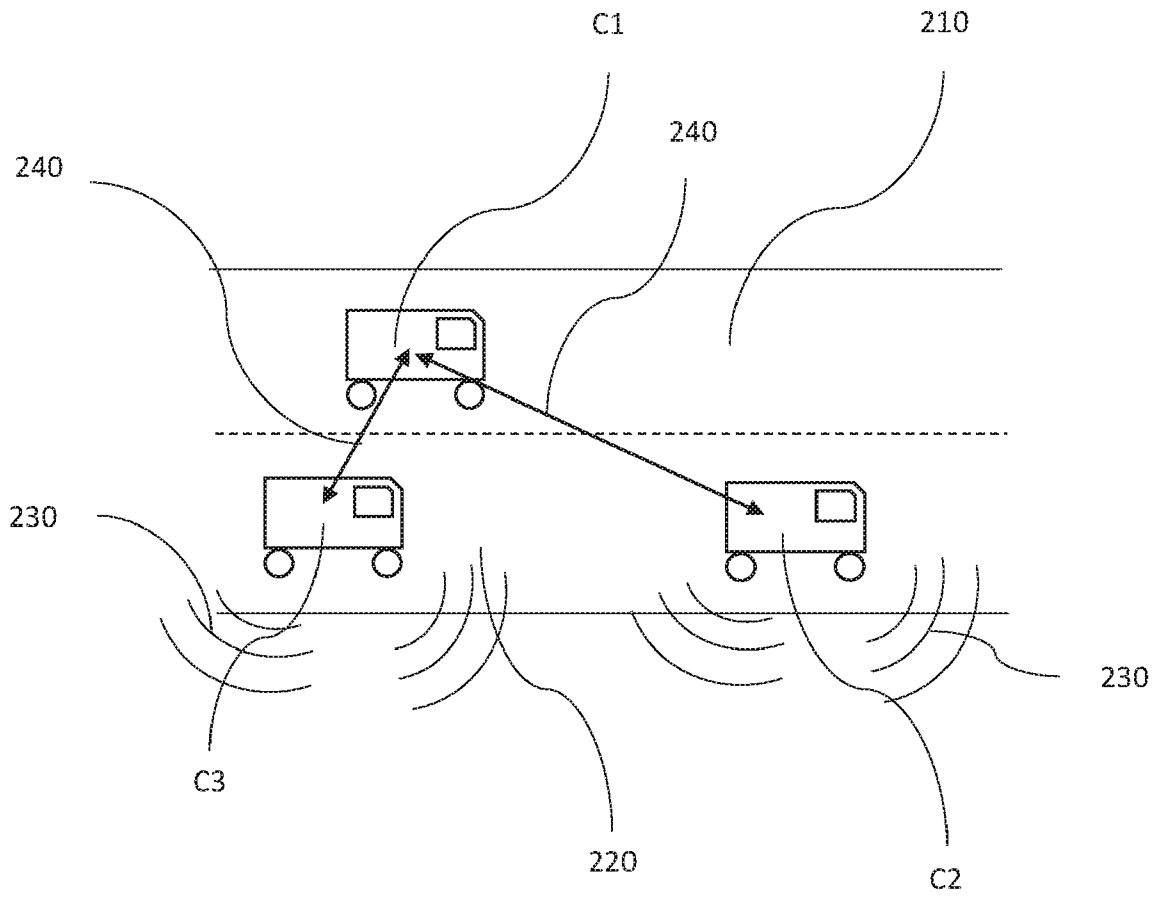


Fig. 2

300

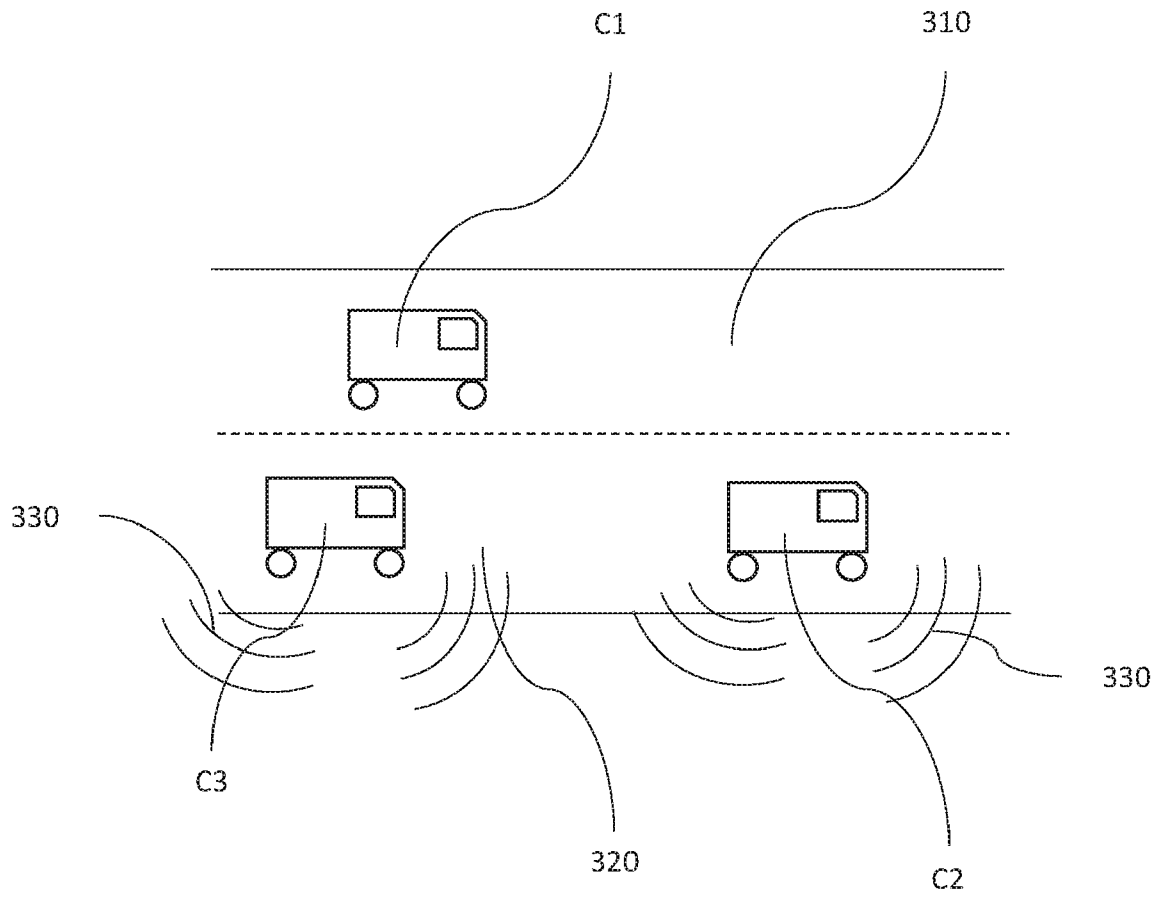


Fig. 3

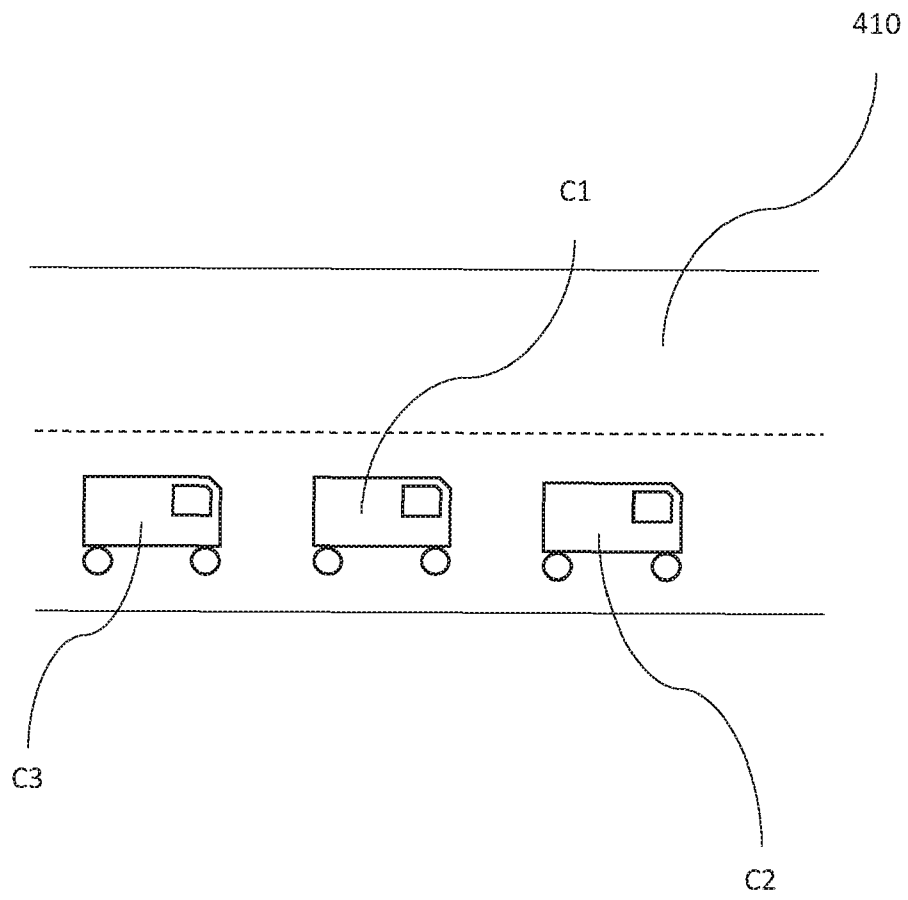


Fig. 4

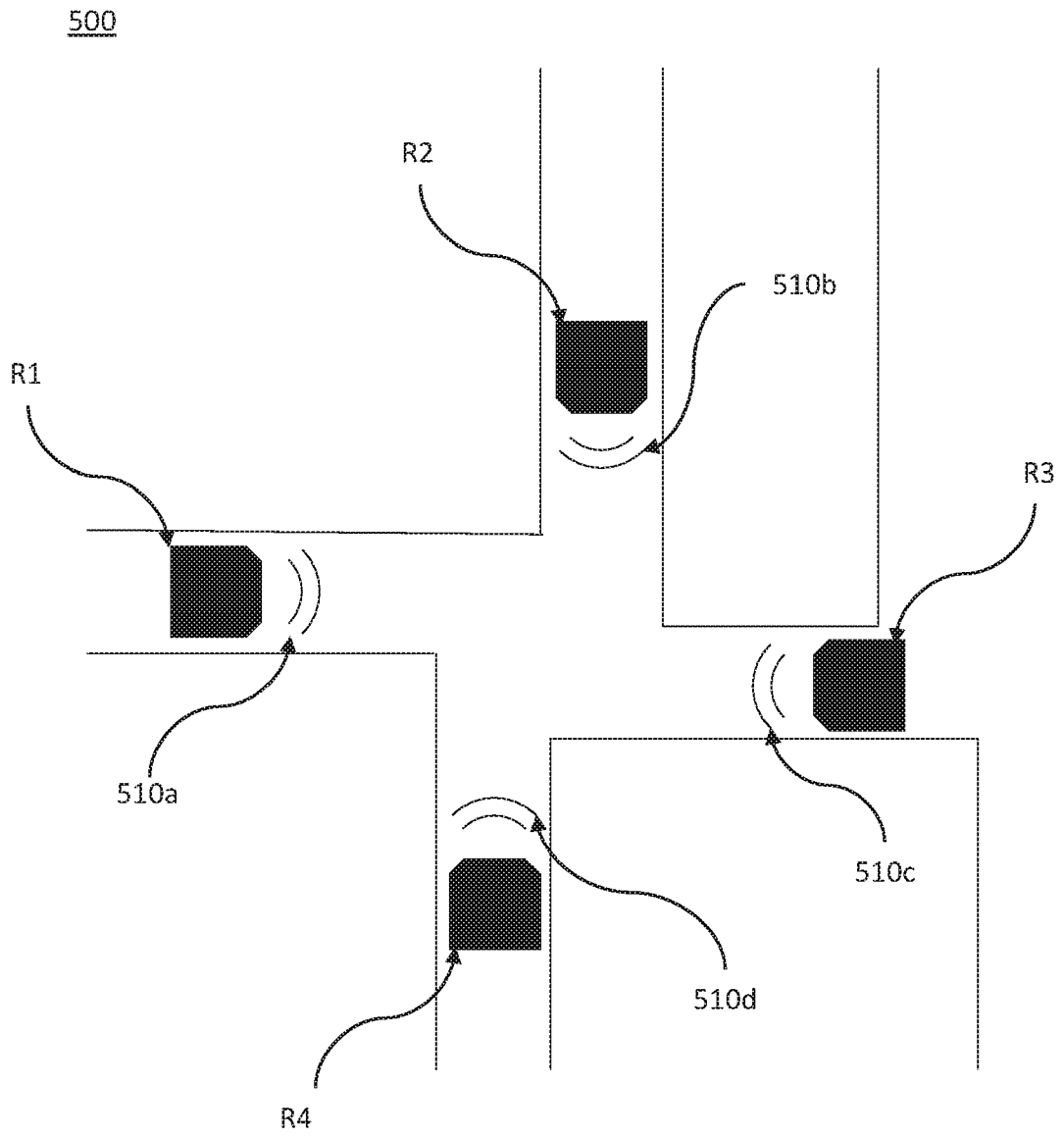


Fig. 5

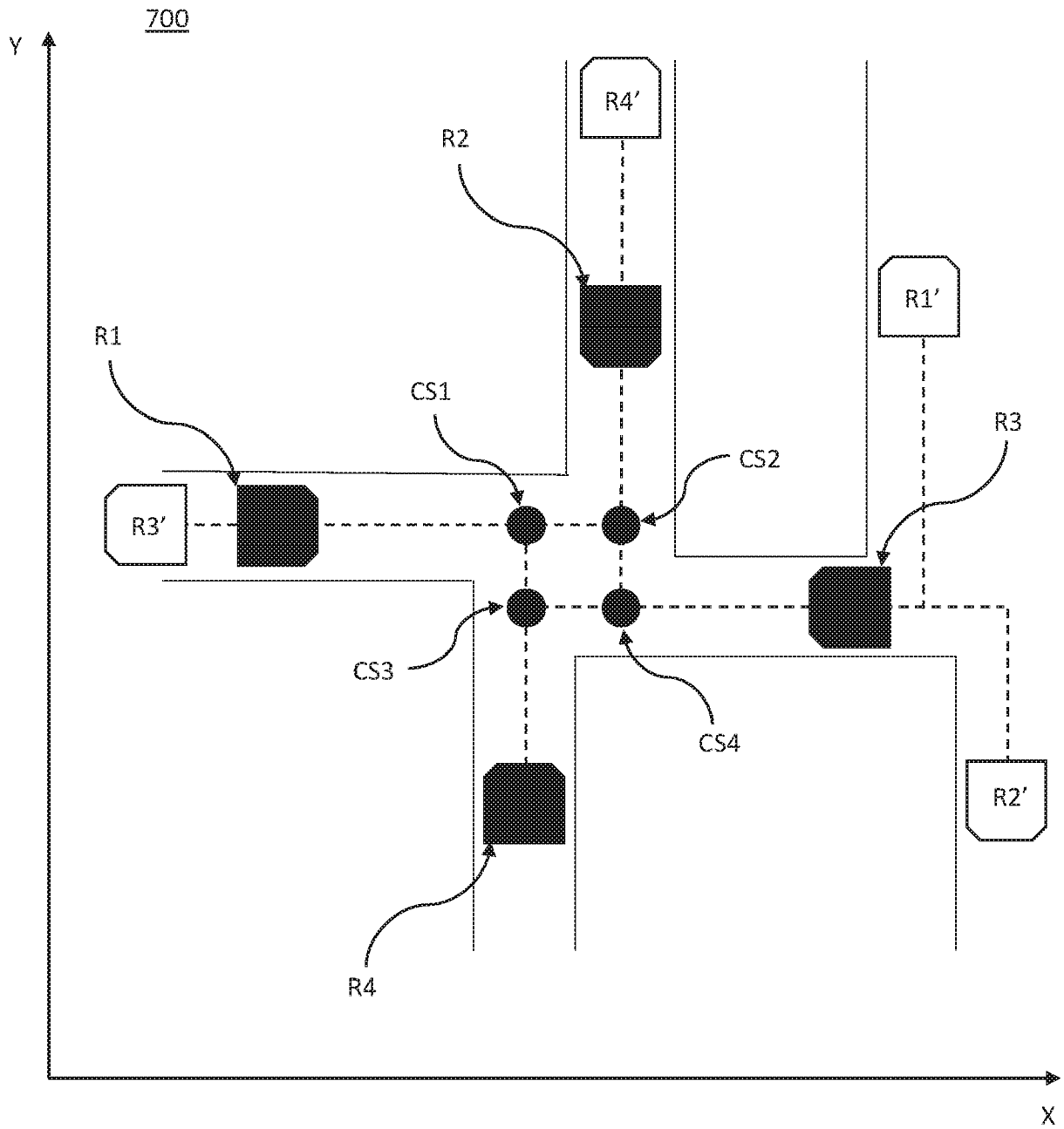


Fig. 7

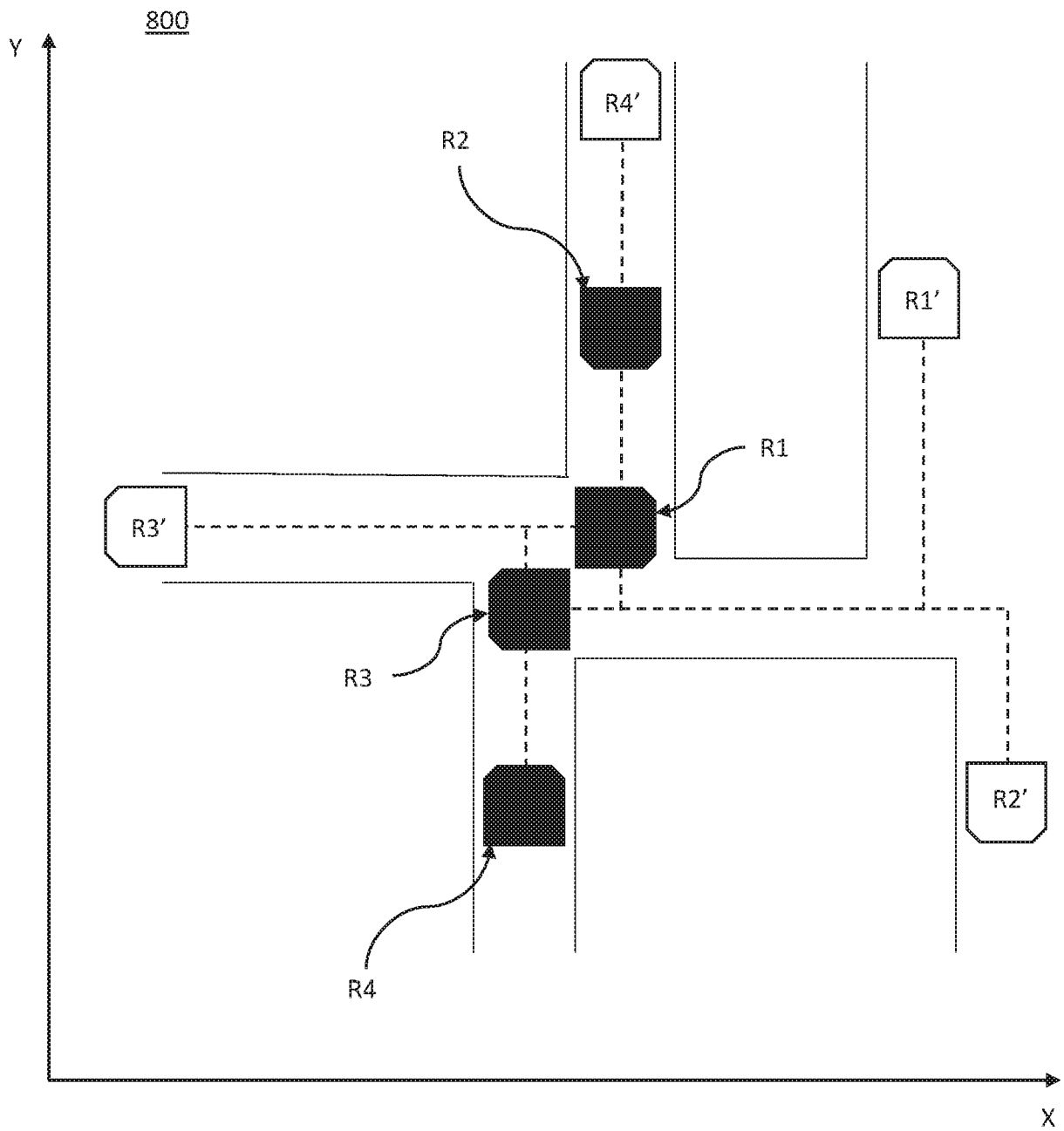


Fig. 8