Descentralización

¿Cómo funciona Bitcoin?

Qué es Bitcoin?

Bitcoin = Scroogecoin sin Scrooge

Como remover a Scrooge?

Dos componentes necesarios para descentralización:

- Mecanismos técnicos (blockchain, proof of work, mining)
- Incentivos económicos (block reward, transaction fee, community)

Dependen uno del otro cíclicamente:

- Mecanismos técnicos son seguros si hay mucha gente participando en el protocolo
- Gente participa en el protocolo si los incentivos económicos son buenos

Centralización vs. Descentralización

Un sistema descentralizado: Internet

Un sistema centralizado: Una red social (Facebook)

Semi - descentralizado: email

En realidad

No existe un sistema completamente descentralizado

Ni siquiera Bitcoin:

- Red p2p = completamente descentralizada
- Mining = casi completamente centralizado
- Cambio de las reglas = BitcoinCore desarrolladores

Descentralización

Algunas preguntas:

1. **Quien guarda el blockchain?**
2. **Quien decide cuales transacciones son válidas?**
3. **Quien crea nuevos Bitcoins?**
4. Quien decide como cambiar las reglas del protocolo?
5. Como Bitcoin logra tener valor en USD?

Consenso distribuido

Consenso distribuido es nuestro método para lograr descentralización

Consenso distribuido. Todos los participantes en el sistema tienen que estar de acuerdo sobre cuales transacciones son válidas. Solo en este caso las transacciones pueden aparecer en un bloque de nuestro blockchain.

Distribuido = no hay una entidad central decidiendo cuales transacciones son válidas. Todos participantes en el protocolo tienen que estar en acuerdo = **Consenso**)

Un problema común en CS

Bases de datos distribuidas:

- E.g. una red social tiene sus datos en millones de servidores
- Servidores = nodos en un sistema distribuido

- Muchos datos son duplicados
- Para hacer update a un dato, hay que hacer el update a todos los servidores
- Esto se logra a través consenso distribuido

- Aplicación clásica de consenso distribuido: un diccionario global

Consenso distribuido

Protocolo básico de consenso distribuido.

Tenemos una red de *n* nodos. Cada nodo tiene un valor de entrada. Algunos nodos son maliciosos. El protocolo logra lo siguiente:

1. Al terminar el protocolo, todos los nodos honestos están en acuerdo sobre el valor de input.
2. El valor fue generado por un nodo honesto.

Consenso distribuido en Bitcoin

Que significa esto para Bitcoin?

- Bitcoin es una red p2p de nodos
- Nodos reciben transacciones
- Los nodos tienen que estar de acuerdo sobre cuales transacciones son válidas
- Al ejecutar el protocolo, estas transacciones se agregan al libro contable global

Los nodos quieren lograr consenso sobre cuales transacciones son válidas en este momento.

Este consenso reemplaza a Scrooge.

Consenso distribuido en Bitcoin

Alice quiere pagar a Bob (dos pasos):

1. Alice crea la transacción (Inputs, Outputs, firmas)
2. Alice transmite la transacción a la red p2p de Bitcoin

Importante: No es necesario que Alice o Bob sean nodos en la red.

Alice no manda la transacción a Bob. La manda solo a la red. No importa si Bob no sabe sobre la transacción. Si la red la acepta la plata le pertenece a él.

Alice paga a Bob

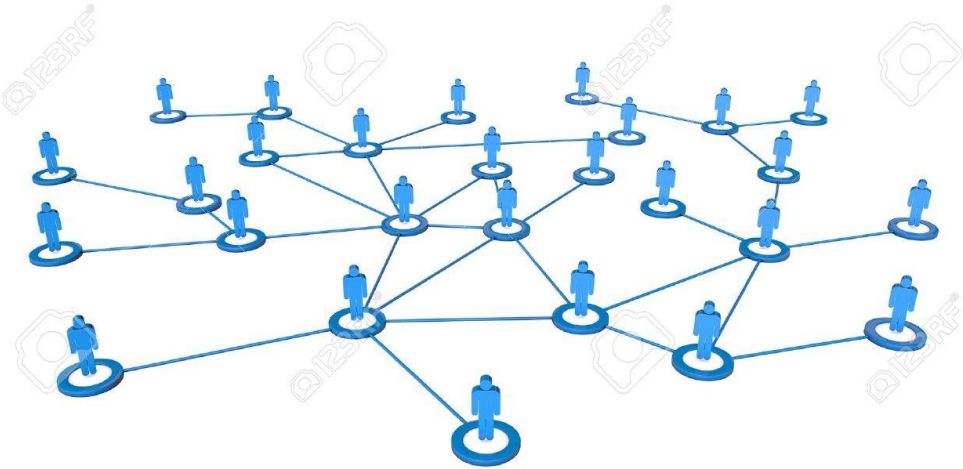
Step 1

transID: 74		type: PayCoins
coins consumed		
num	consumed coinID	
0	coinID 73(1)	
coins created		
num	value	recipient
0	3.2	0xf4…
Signature by $\mathbf{LS}_{\mathbf{Alice}}$		

Alice paga a Bob

Step 2

transID:…



Qué pasa en Bitcoin?

Como vamos a guardar nuestro ledger?

Lo mismo como en Scroogecoin. En un blockchain.

En cada paso, los nodos quieren ponerse de acuerdo sobre el siguiente bloque.

Importante: Todos los nodos guardan el blockchain. Todo el blockchain.

No hay una entidad central guardando el blockchain para nosotros!!!

Funcionamiento de Bitcoin

1. En cada momento los nodos de Bitcoin tienen un blockchain de transacciones válidas (sobre cuales ya tenemos un consenso -- casi)
2. Nodos reciben nuevas transacciones y las agregan a su pool de transacciones por agregar al blockchain
3. **Cada 10 minutos cada nodo propone el siguiente bloque (basado en transacciones que tiene en su pool)**
4. **Nodos ejecutan protocolo de consenso distribuido (input = el bloque)**
5. Si el consenso es exitoso vamos a seleccionar un bloque válido

Funcionamiento de Bitcoin

Observaciones:

- Hay nodos maliciosos
- Si solo un nodo propone el bloque elegido, este bloque es todavía válido
- Si una transacción válida no fue incluida en este bloque no hay problemas (va a estar incluida en uno de los siguientes bloques)

Es fácil lograr esto?

Problemas de implementación:

- La red p2p no está perfecta (comunicación incompleta)
- Nodos entran y salen de la red
- Los nodos no tienen una identidad fija (no se pueden identificar)
- No hay una noción de tiempo global válido para todos los nodos

Noción de tiempo global

Noción de tiempo global

Creada: 10:00

transA

Noción de tiempo global

Creada: 10:00

transA

Creada: 10:05

transB

Noción de tiempo global

Creada: 10:00

transA

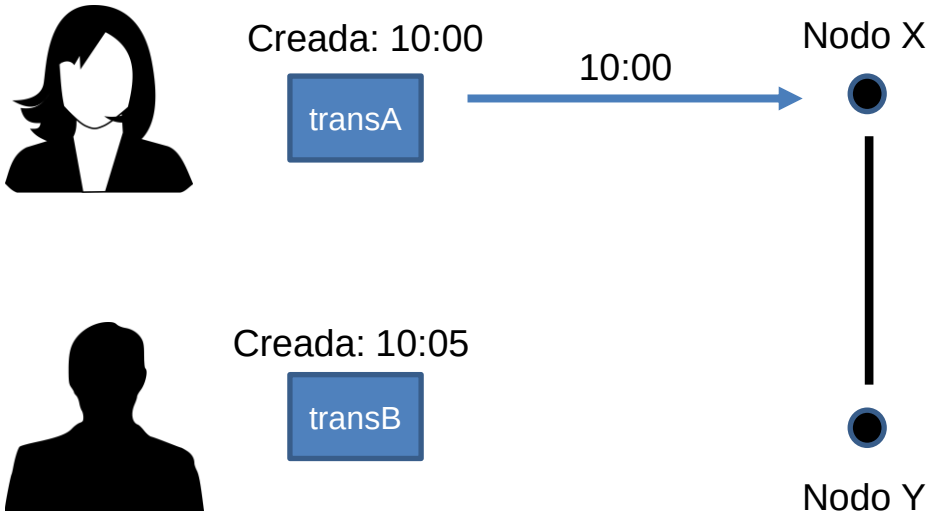
Creada: 10:05

transB

Nodo X

Nodo Y

Noción de tiempo global



Noción de tiempo global

Creada: 10:00

transA

10:00

Nodo X

transA

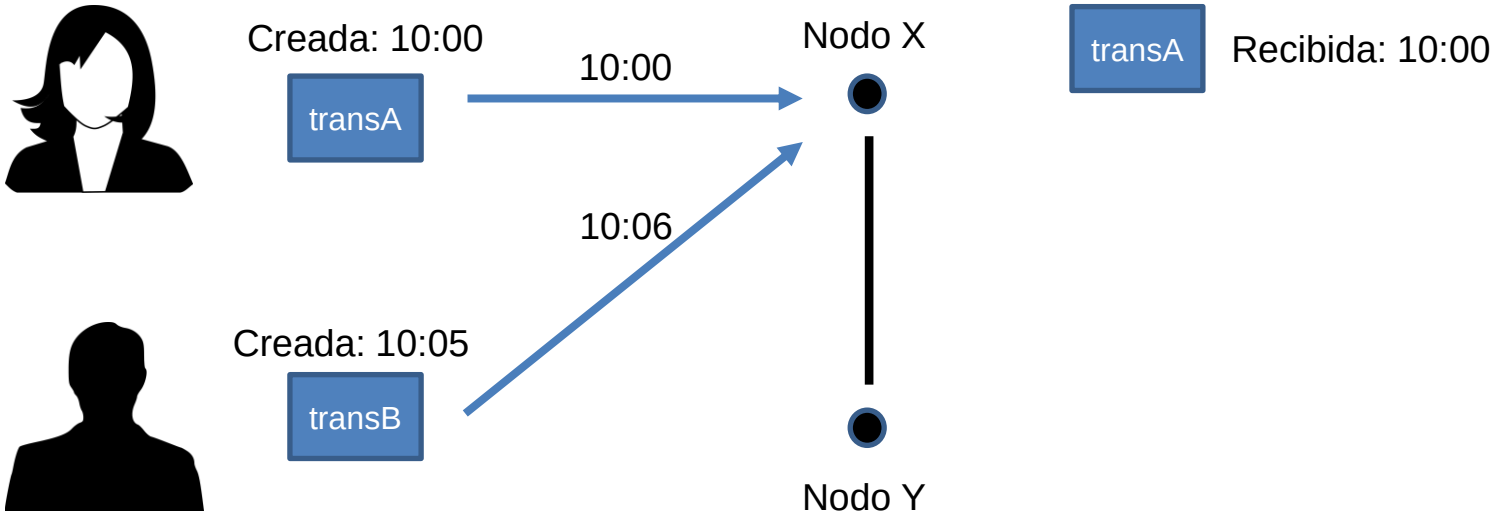
Recibida: 10:00

Creada: 10:05

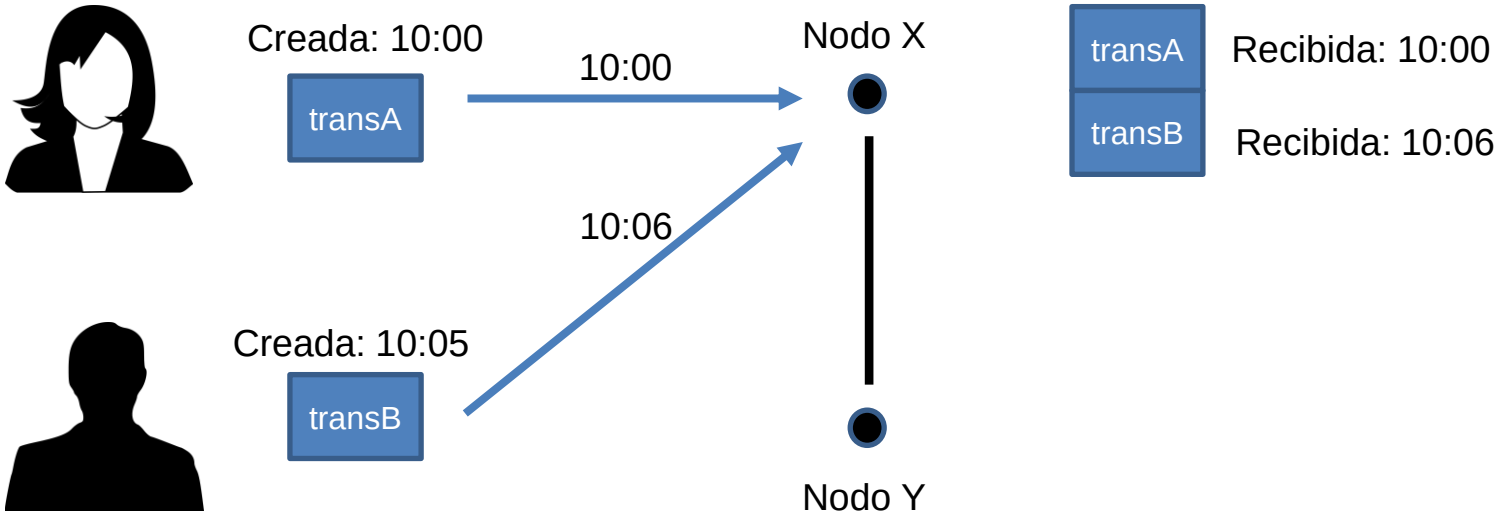
transB

Nodo Y

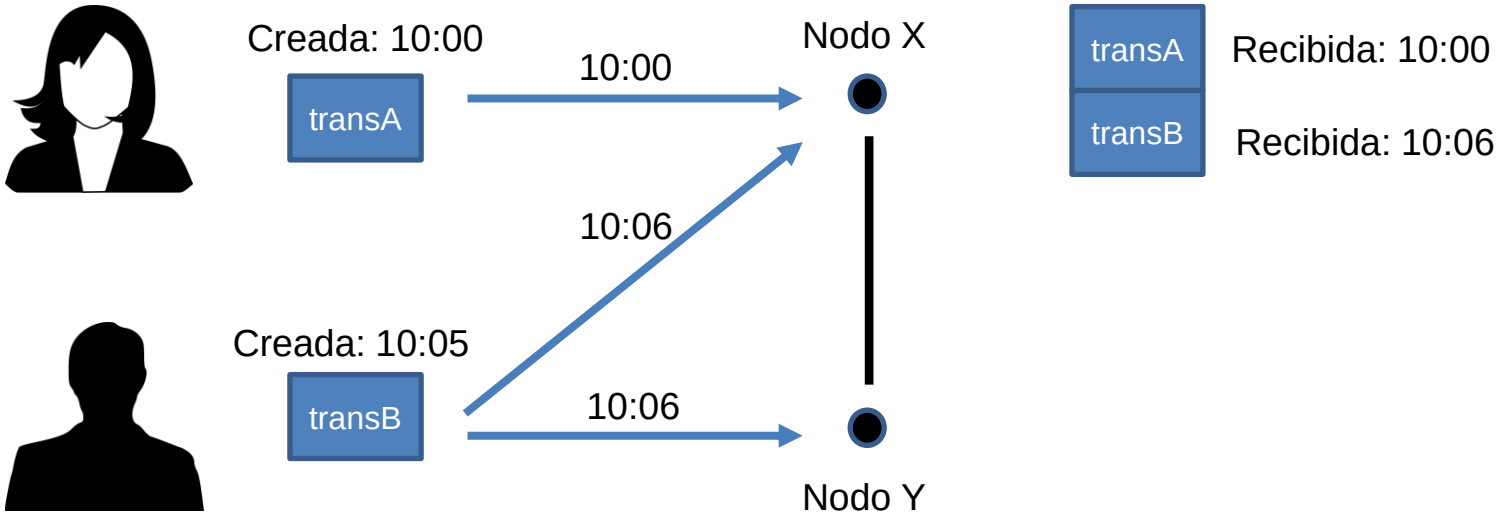
Noción de tiempo global



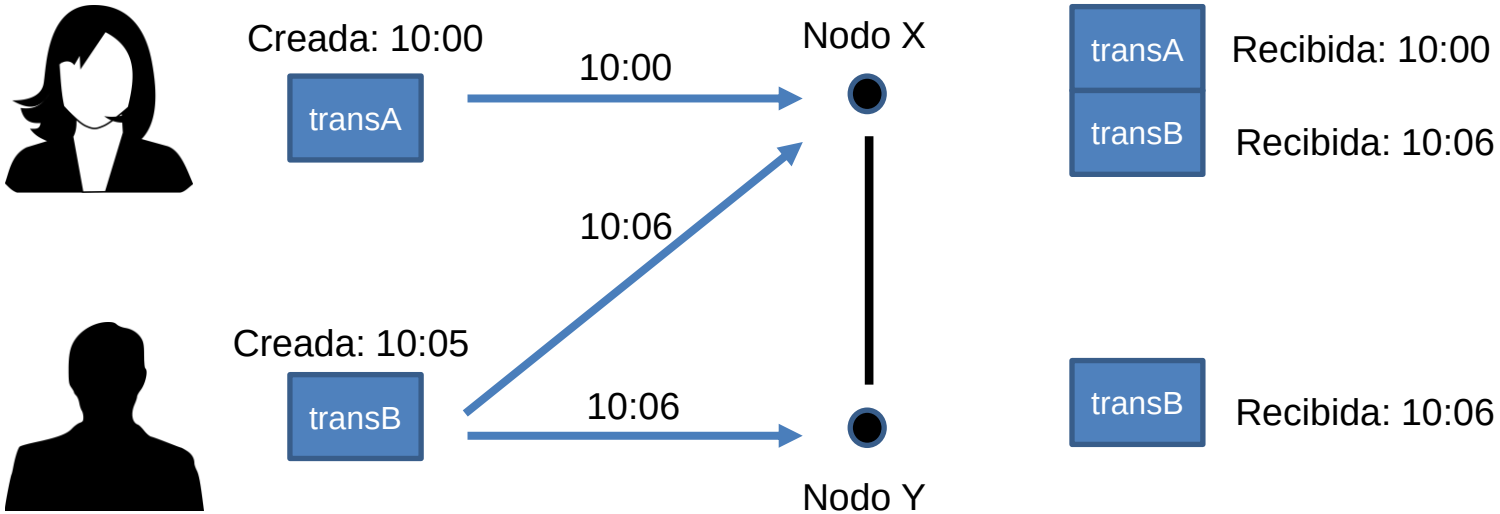
Noción de tiempo global



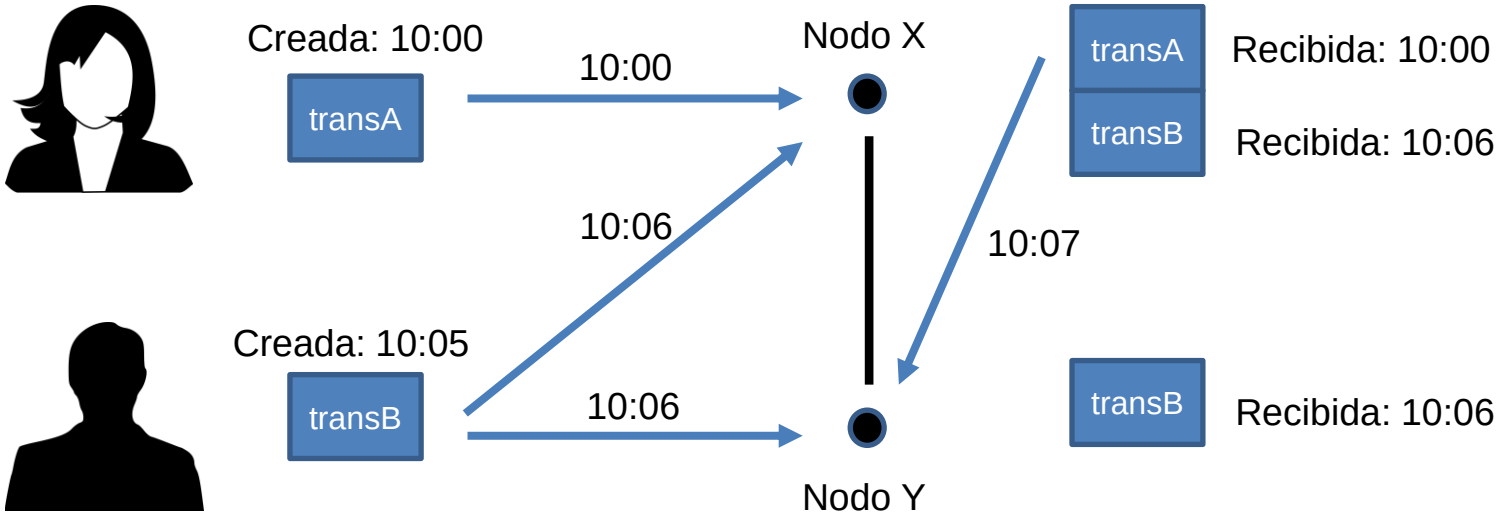
Noción de tiempo global



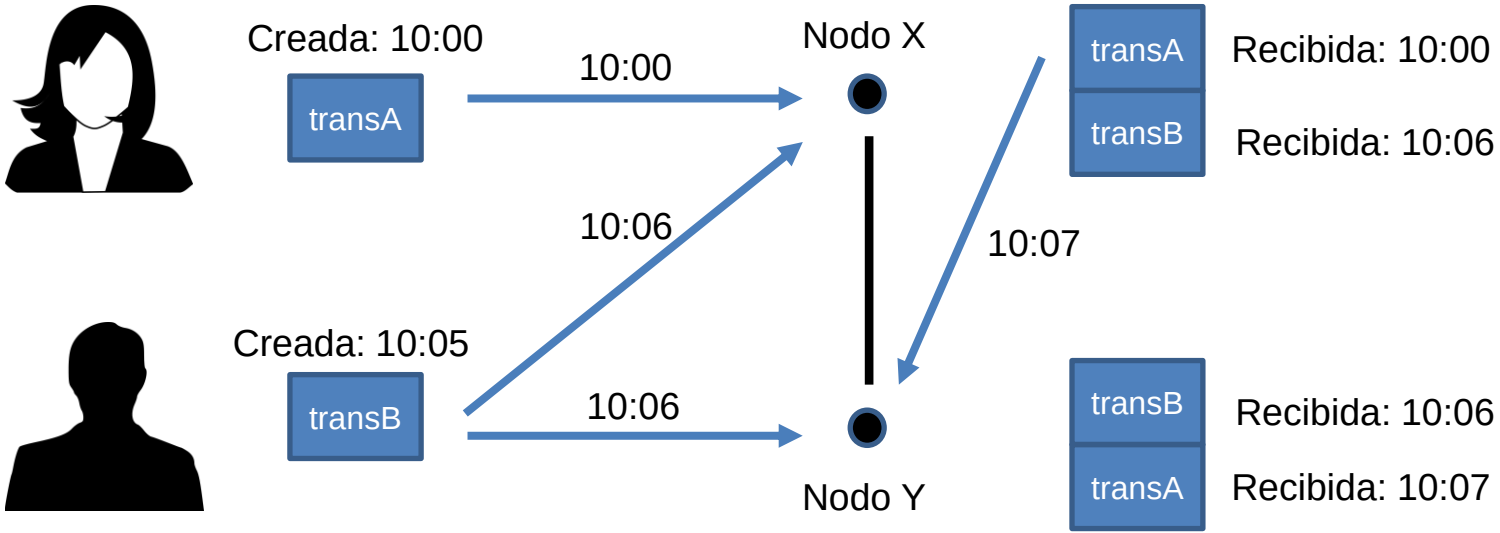
Noción de tiempo global



Noción de tiempo global



Noción de tiempo global



Nodos no tienen IDs

Por diseño, los nodos no tiene IDs

Sybil attack = Crear 1000 nodos maliciosos para atacar la red

Como no hay Ids, no podemos castigar o remover nodos malos

Tampoco podemos estimar cuantos nodos malos hay en la red

Como vamos a lograr el consenso?

En teoría

El consenso no existe:

- Generales Bizantinos (con un tercio nodos malos)
- Fischer-Lynch-Paterson (con un nodo malicioso)

Paxos:

- Siempre consistente, pero puede entrar a una calle sin salida

En la práctica

El consenso de Bitcoin funciona casi perfecto!!!

Qué pasa aquí?

Los resultados de imposibilidad son para bases de datos:

- Un modelo muy especifico

Diferencia en el mundo de Bitcoin:

- Incentivos económicos (consenso solo para criptomonedas)
- Aleatoriedad (consenso evoluciona con el tiempo)

La teoría para Bitcoin todavía no está desarrollada

Trabajo en desarrollo

Shameless plug

Cryptocurrency Mining Games with Economic Discount and Decreasing Rewards

Marcelo Arenas
PUC Chile & IMFD Chile, Santigo, Chile
marenas@ing.puc.cl

Juan Reutter
PUC Chile & IMFD Chile, Santigo, Chile
jreutter@ing.puc.cl

Etienne Toussaint
University of Edinburgh, Edinburgh, UK
etienne.toussaint@ed.ac.uk

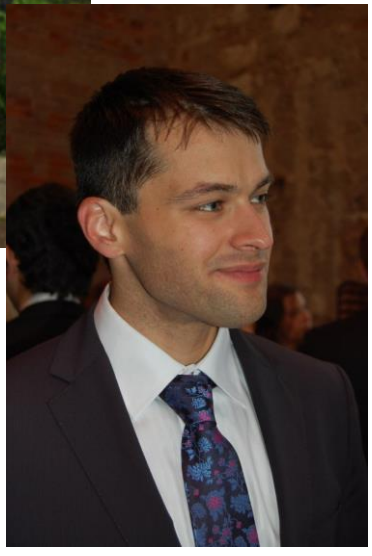
Martín Ugarte
PUC Chile & IMFD Chile, Santigo, Chile
martin@martinugarte.com

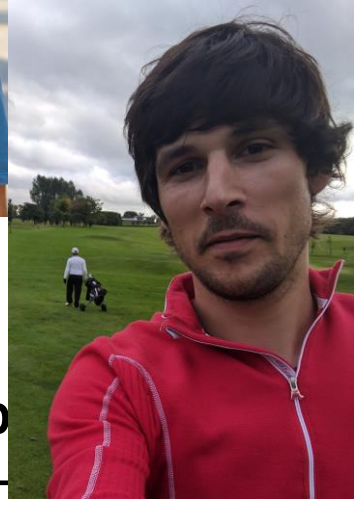
Francisco Vial
ProtonMail & IMFD Chile, Santiago, Chile
fvial@pm.me

Domagoj Vrgoč
PUC Chile & IMFD Chile, Santigo, Chile
dvrgoc@ing.puc.cl

STACS 2020

Trabajo en desarrollo





Cryptocurrency Mining Games with
and Decreasing Rewards

enas
FD Chile, Santigo, Chile
.cl

fvial@p

Doma
PUC C
dvrgoc

STACS 2020

Consenso implícito

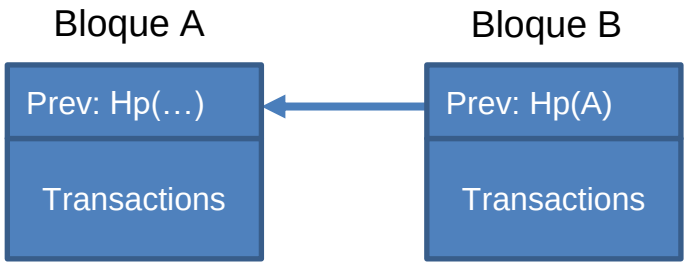
Asumamos que se puede seleccionar un nodo de manera aleatoria evitando el ataque de sybil (i.e. podemos detectar nodos replicados).

<u>El protocolo:</u>

1. Nodos escuchan/reciben nuevas transacciones
2. Cada nodo agrupa transacciones en un bloque
3. Cada 10 minutos se elige **aleatoriamente** un nodo para proponer su bloque
4. Otros nodos van a aceptar el bloque solo si está válido
5. Aceptación se expresa extendiendo el blockchain desde este bloque

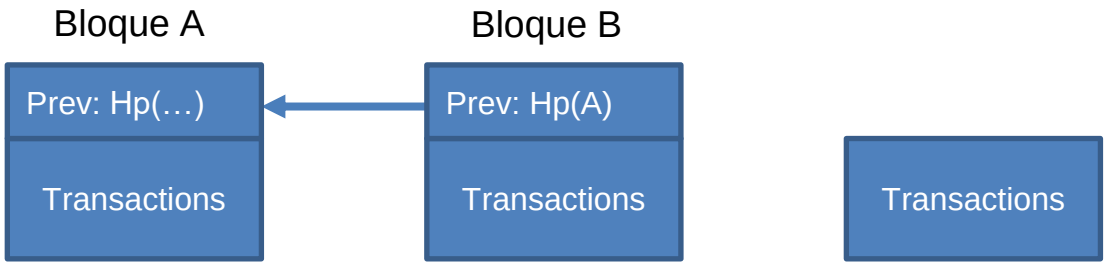
Consenso implícito

Aceptación de Bloque X



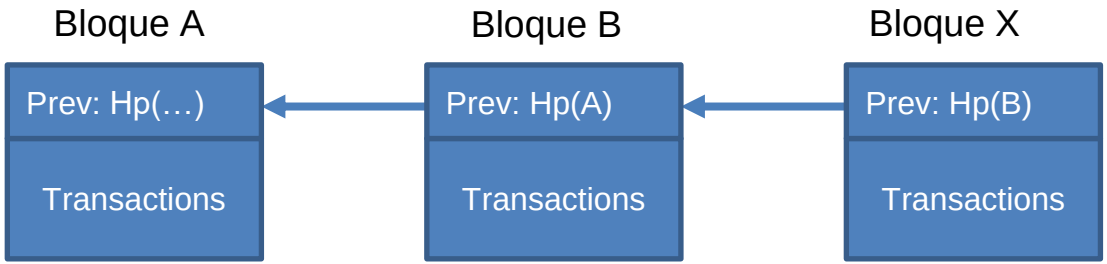
Consenso implícito

Aceptación de Bloque X



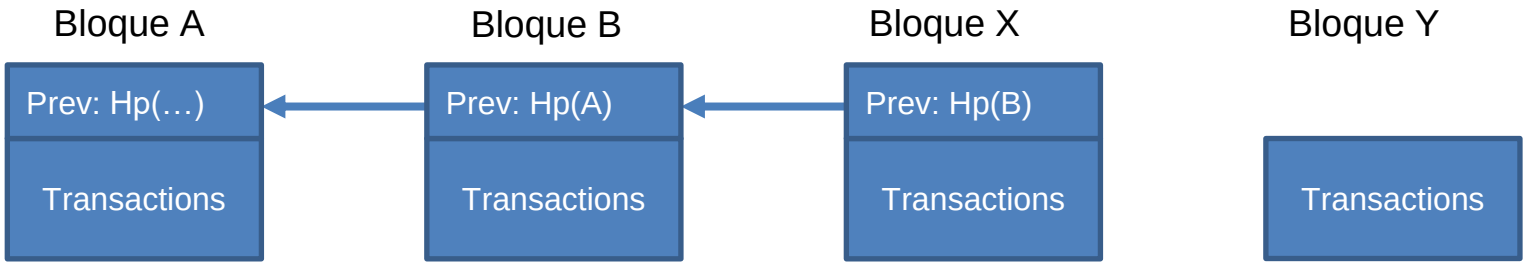
Consenso implícito

Aceptación de Bloque X



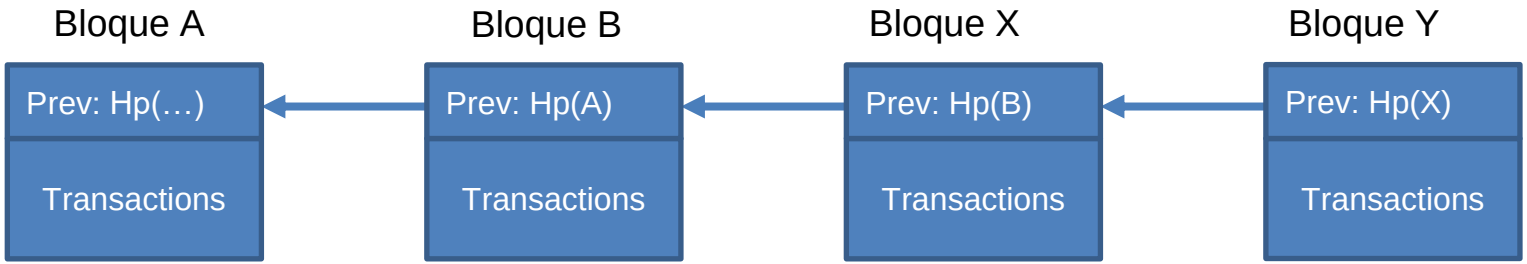
Consenso implícito

Aceptación de Bloque X



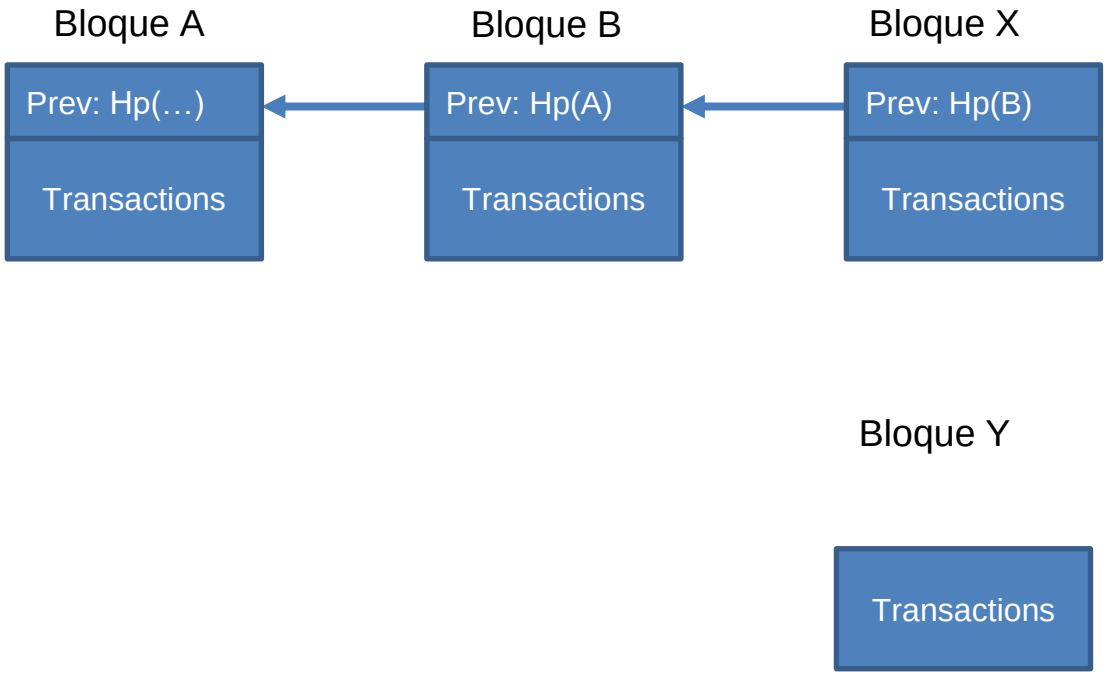
Consenso implícito

Aceptación de Bloque X



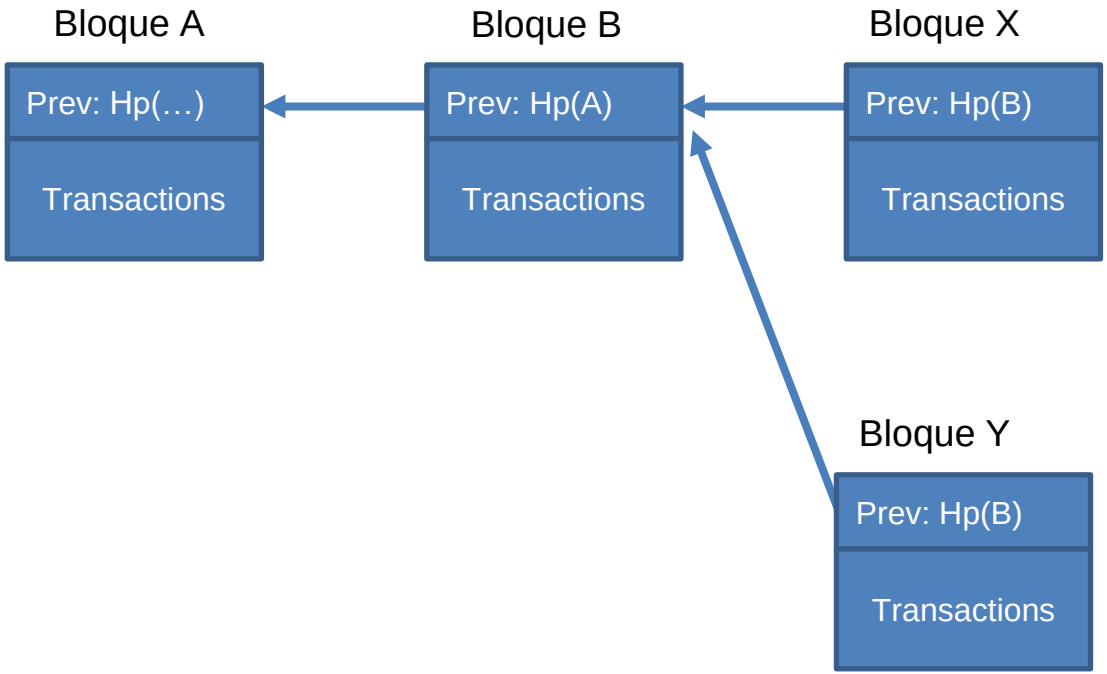
Consenso implícito

Rechazo de Bloque X



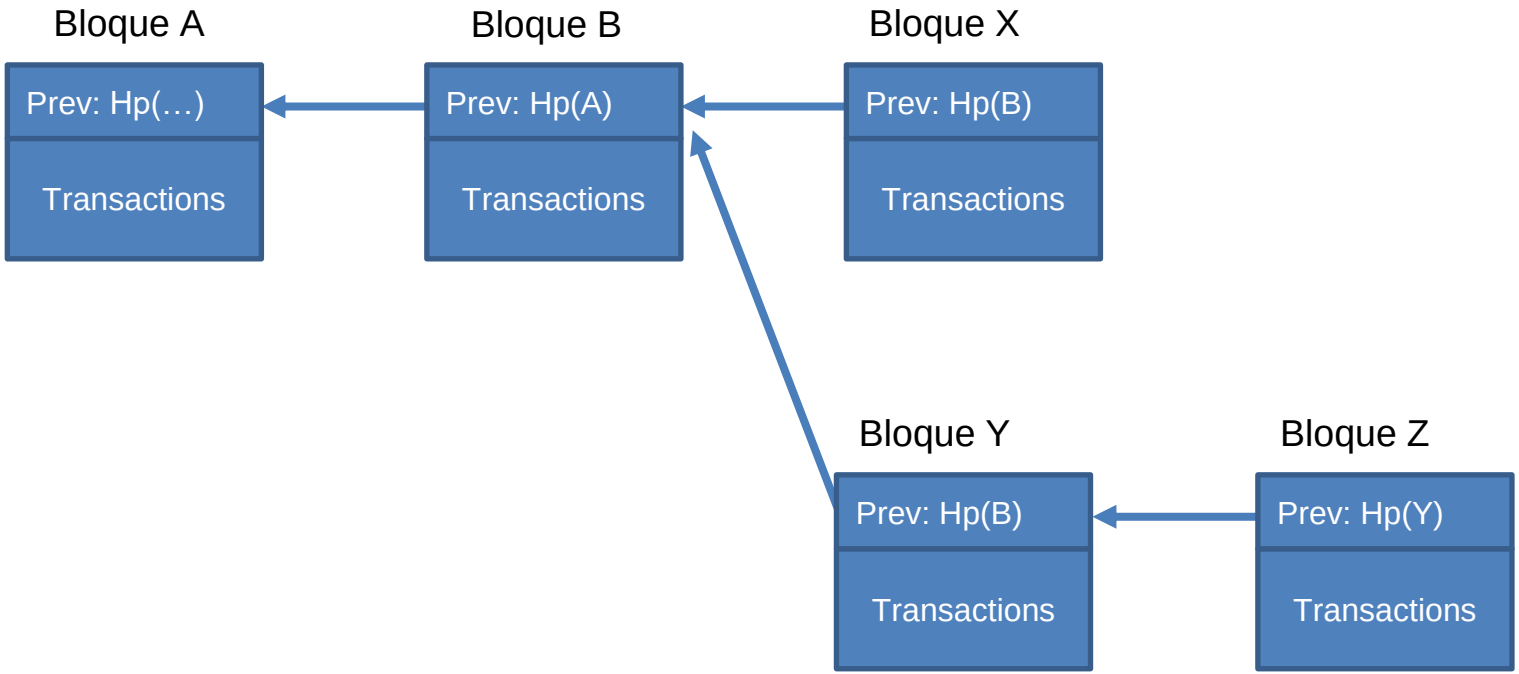
Consenso implícito

Rechazo de Bloque X



Consenso implícito

Rechazo de Bloque X



Consenso implícito

Nodos aceptan/rechazan el bloque X de manera implícita:

- No lo declaran de manera publica
- Si lo aceptan extienden el blockchain desde el Bloque X
- Si lo rechazan lo extienden desde otro bloque anterior a Bloque X

Recordar como implementamos blockchain usando dicionarios!

Propiedades de consenso implícito

Se pueden robar los Bitcoins en consenso implícito?

Propiedades de consenso implícito

Se pueden robar los Bitcoins en consenso implícito?

- Nah
- Si le toca a Alice proponer el próximo bloque, debería fabricar una firma
- Fácil para detectar (firmas no verifican)

- En conclusión: ni siquiera si un nodo malo propone el bloque, no puede mandarse toda la plata a si mismo (otros nodos van a rechazar el bloque)

Propiedades de consenso implícito

Puede funcionar denial-of-service attack?

- El nodo elegido quiere prohibir a un usuario publicar su transacción

Propiedades de consenso implícito

Puede funcionar denial-of-service attack?

- El nodo elegido quiere prohibir a un usuario publicar su transacción

- Funciona solo en la ronda donde el nodo malo propone el bloque
- Si la transacción es válida, un nodo bueno la va a incluir en el próximo bloque

- No hay un ataque aquí

Propiedades de consenso implícito

Puede funcionar double spend attack?

- Bob es un vendedor de software online
- Bob acepta Bitcoin (una vez pagado, Bob permite descarga de software)

- La pregunta es, que significa el pago con Bitcoin?

Propiedades de consenso implícito

Como Alice puede ejecutar un ataque de doble gasto?

- Alice forma una transacción pagando el monto a Bob
- Alice transmite la transacción a la red
- Un nodo honesto incluye esta transacción en el siguiente bloque
- Bob permite a Alice descargar el software

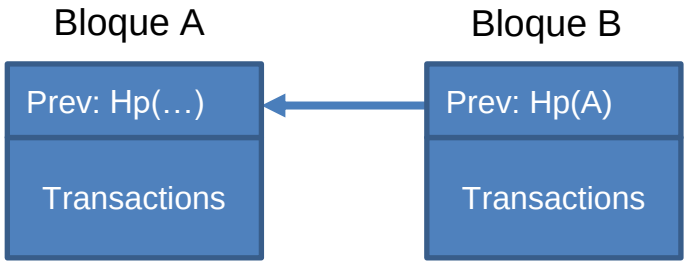
Propiedades de consenso implícito

Como Alice puede ejecutar un ataque de doble gasto?

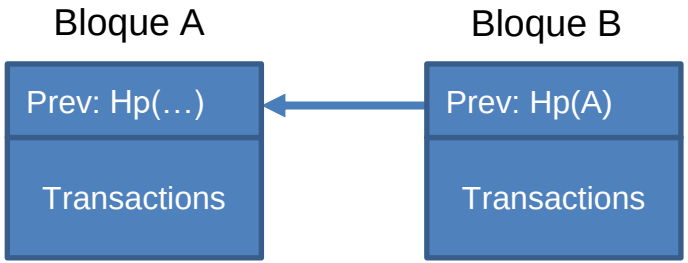
- Alice forma una transacción pagando el monto a Bob
- Alice transmite la transacción a la red
- Un nodo honesto incluye esta transacción en el siguiente bloque
- Bob permite a Alice descargar el software

- Alice propone el próximo bloque
- Qué puede hacer Alice?

Doble gasto

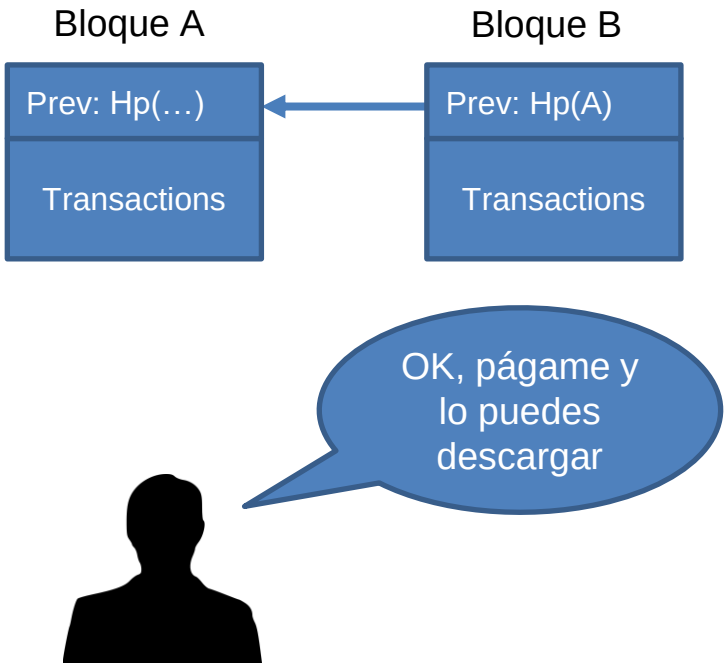

Bloque A
Bloque B
Prev: Hp(…)
Prev: Hp(A)
Transactions
Transactions

Doble gasto


Bloque A
Bloque B
Prev: Hp(…)
Prev: Hp(A)
Transactions
Transactions

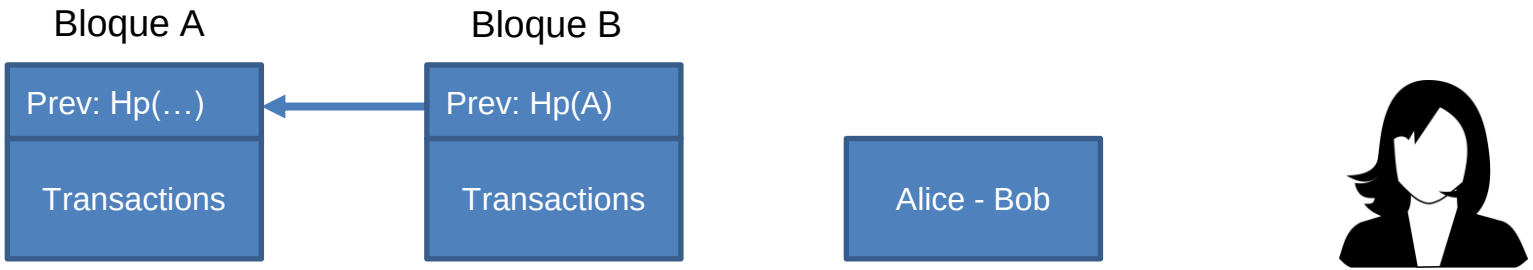

Quiero comprar tu software

Doble gasto

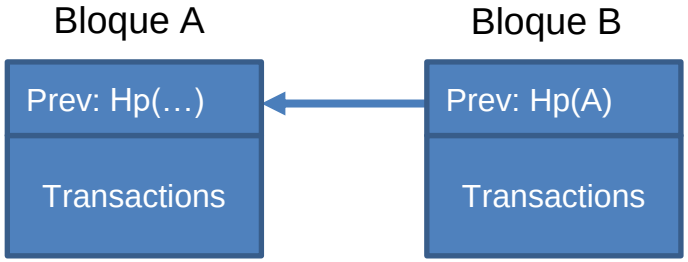

Bloque A
Bloque B
Prev: Hp(…)
Prev: Hp(A)
Transactions
Transactions
OK, págame y lo puedes descargar


Quiero comprar tu software

Doble gasto

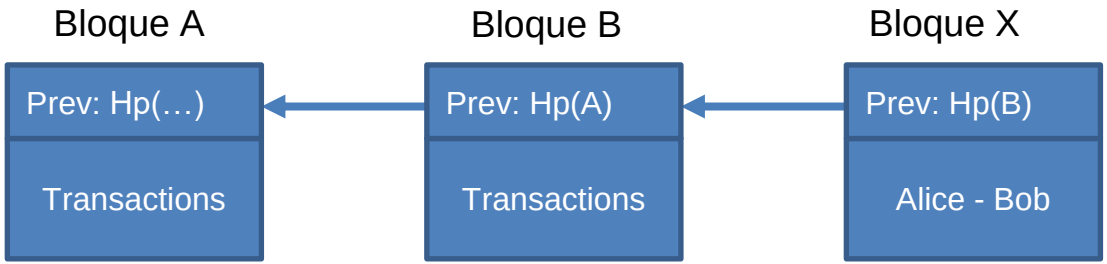

Bloque A
Bloque B
Prev: Hp(…)
Transactions
Prev: Hp(A)
Transactions
Alice - Bob

Doble gasto

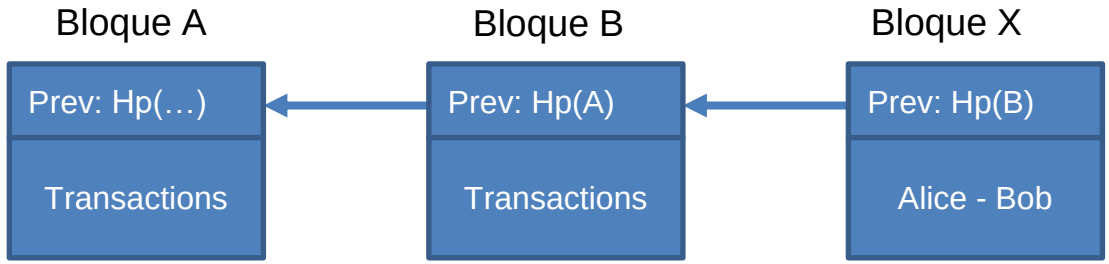


Alice - Bob

Doble gasto

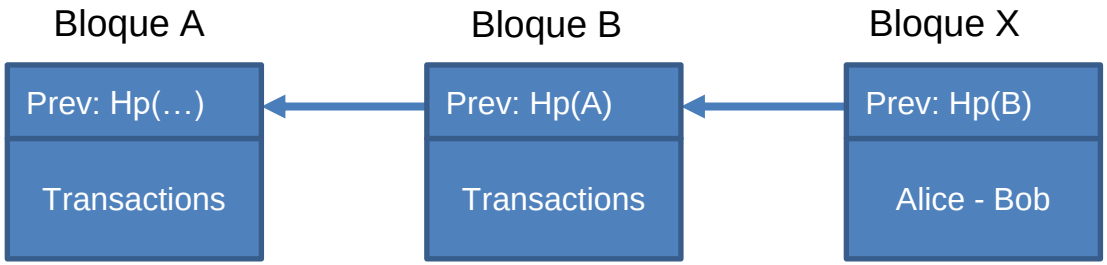

Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob

Doble gasto

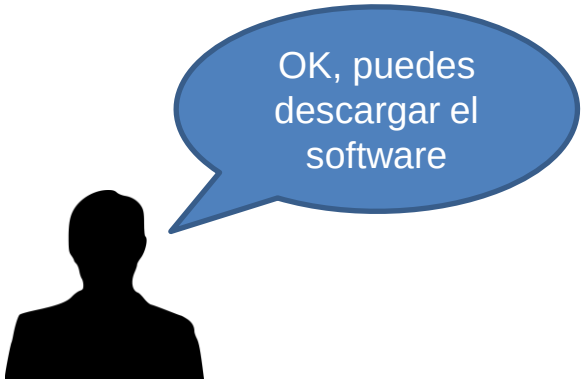

Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob


El pago está
en el Bloque X

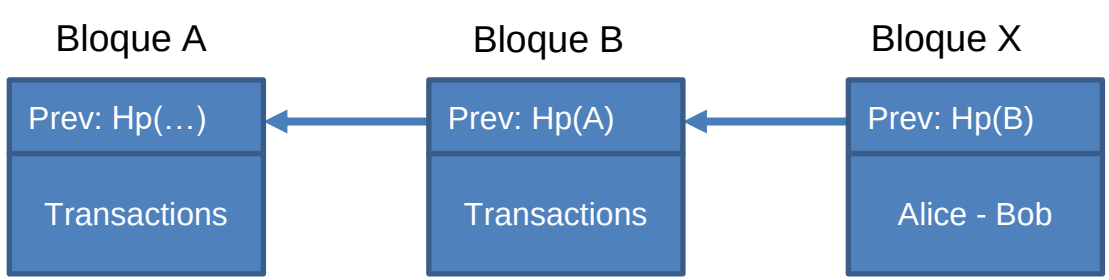
Doble gasto

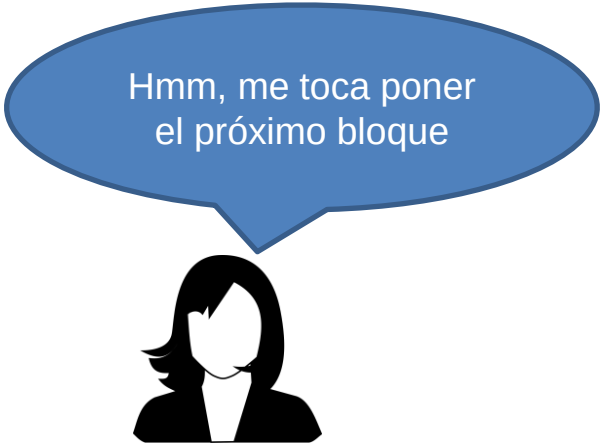

Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob


El pago está en el Bloque X

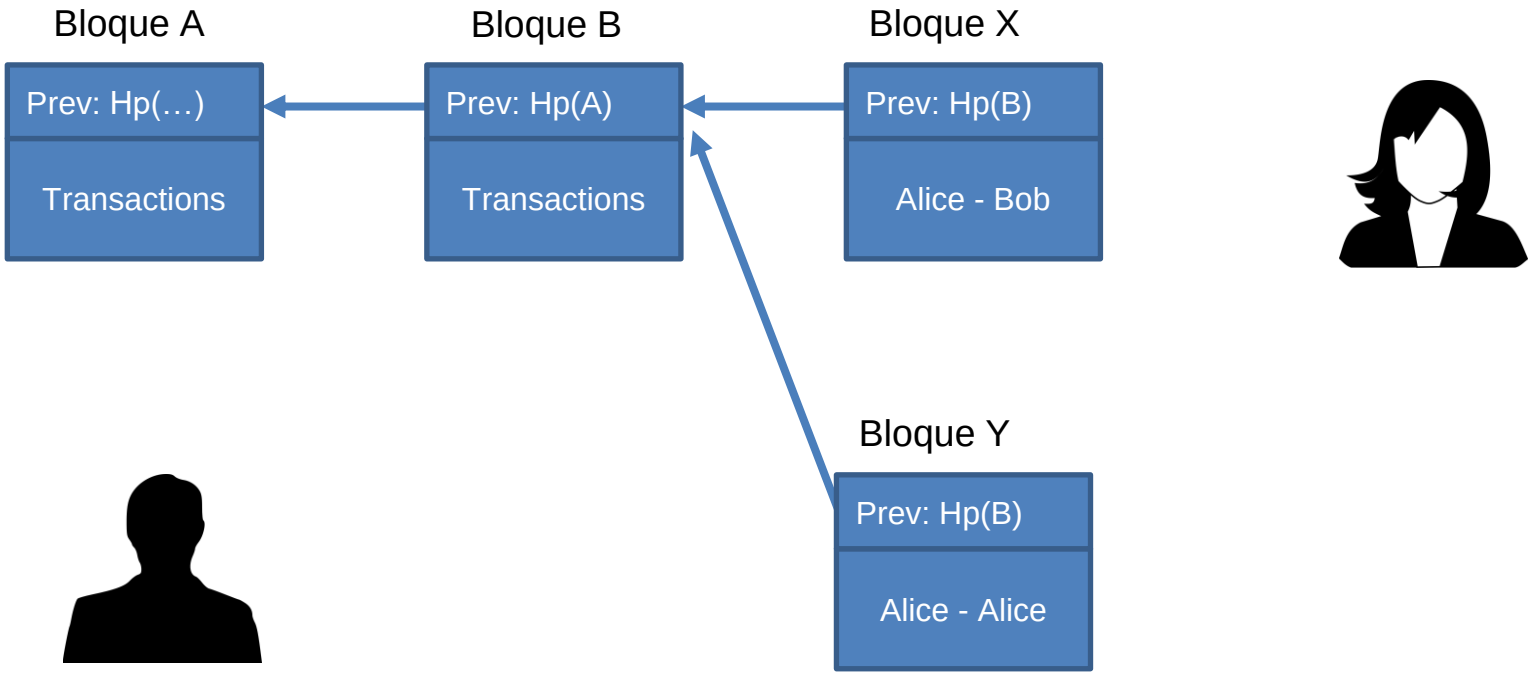

OK, puedes descargar el software

Doble gasto

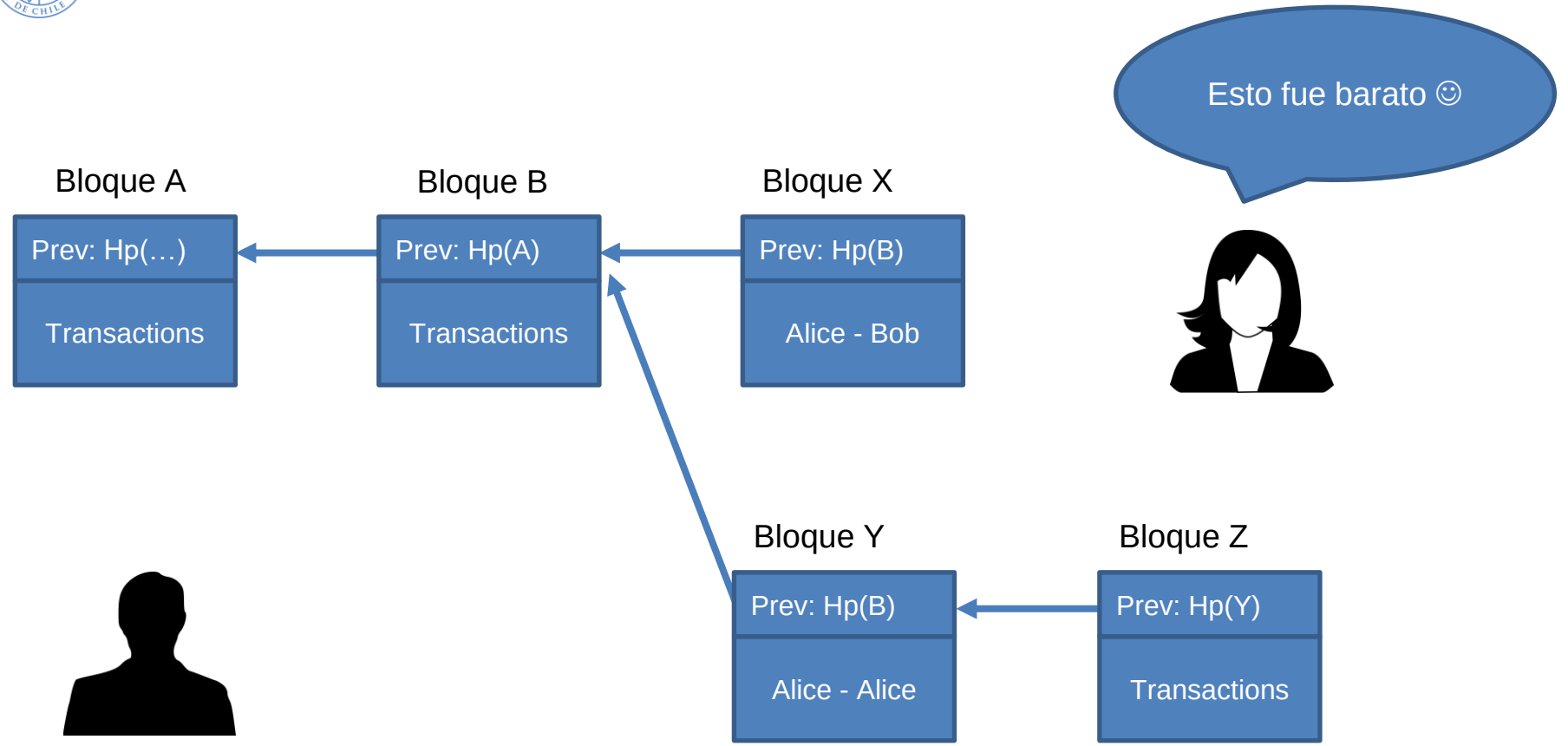

Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob


Hmm, me toca poner
el próximo bloque

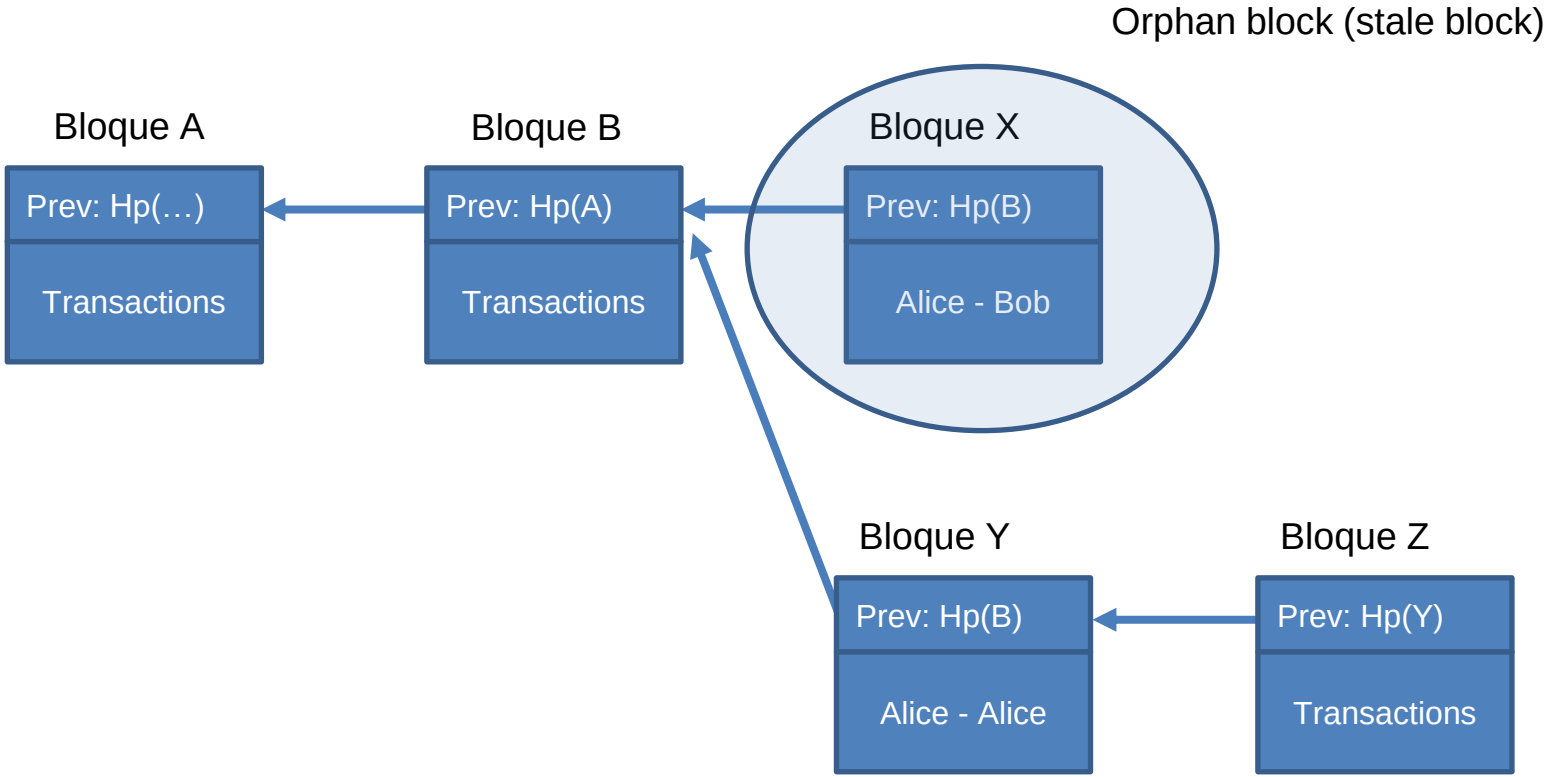
Doble gasto


Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob
Bloque Y
Prev: Hp(B)
Alice - Alice

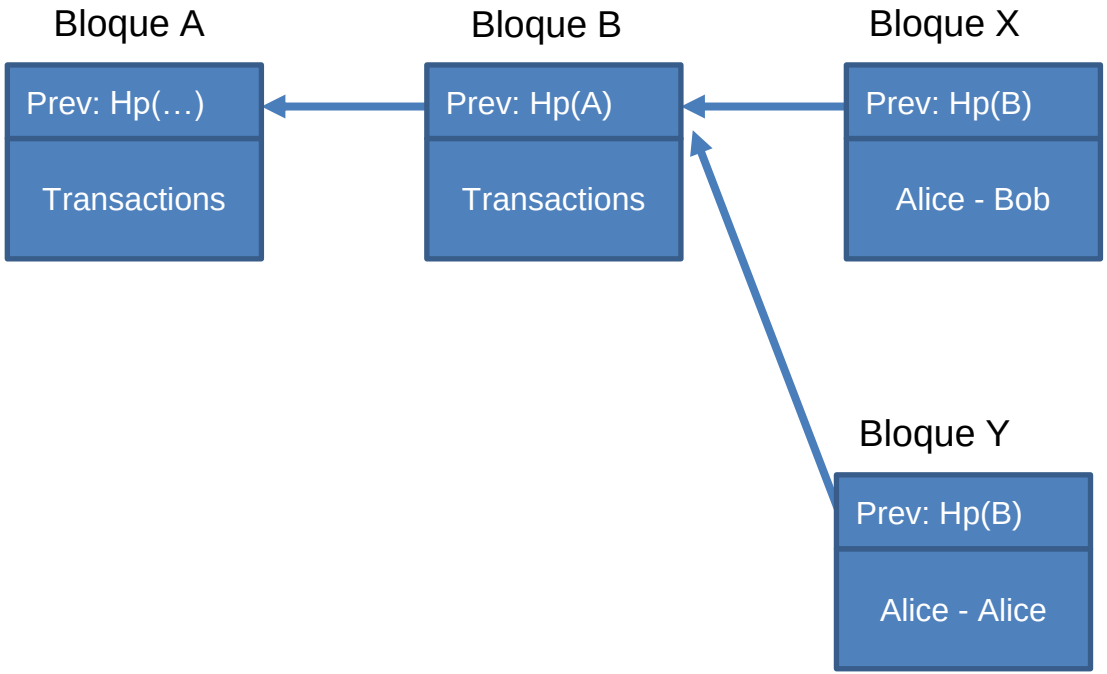
Doble gasto


Esto fue barato ☺
Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob
Bloque Y
Prev: Hp(B)
Alice - Alice
Bloque Z
Prev: Hp(Y)
Transactions

Doble gasto


Orphan block (stale block)
Bloque A
Prev: Hp(…)
Transactions
Bloque B
Prev: Hp(A)
Transactions
Bloque X
Prev: Hp(B)
Alice - Bob
Bloque Y
Prev: Hp(B)
Alice - Alice
Bloque Z
Prev: Hp(Y)
Transactions

Doble gasto



Importante:

- Moralmente X es correcto
- Criptograficamente da lo mismo

- Nodos no conocen la historia
- Para ellos X y Y son válidos
- Pueden extender cualquiera

Propiedades de consenso implícito

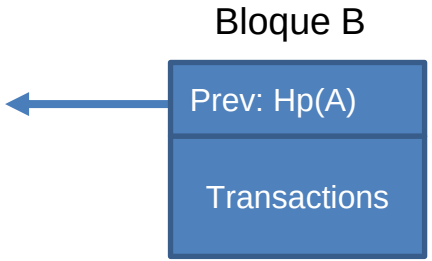
Como Alice puede ejecutar un ataque de doble gasto?

- Alice forma una transacción pagando el monto a Bob
- Alice transmite la transacción a la red
- Un nodo honesto incluye esta transacción en el siguiente bloque

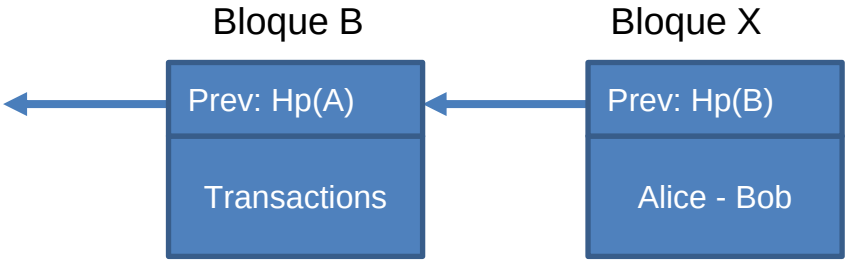
Bob tiene varias opciones:

- Zero confirmations (doble gasto de bajo nivel)
- Una confirmación (doble gasto ya explicado)
- *k* confirmaciónes

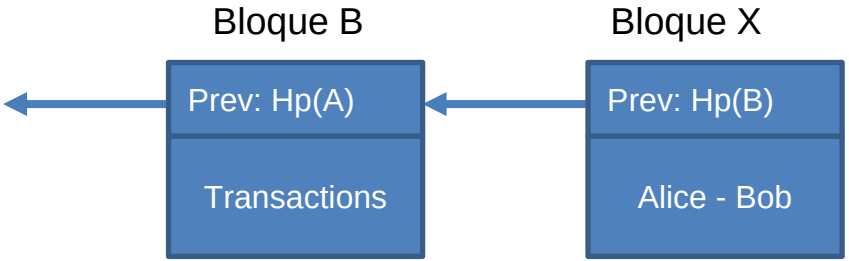
Múltiples confirmaciones



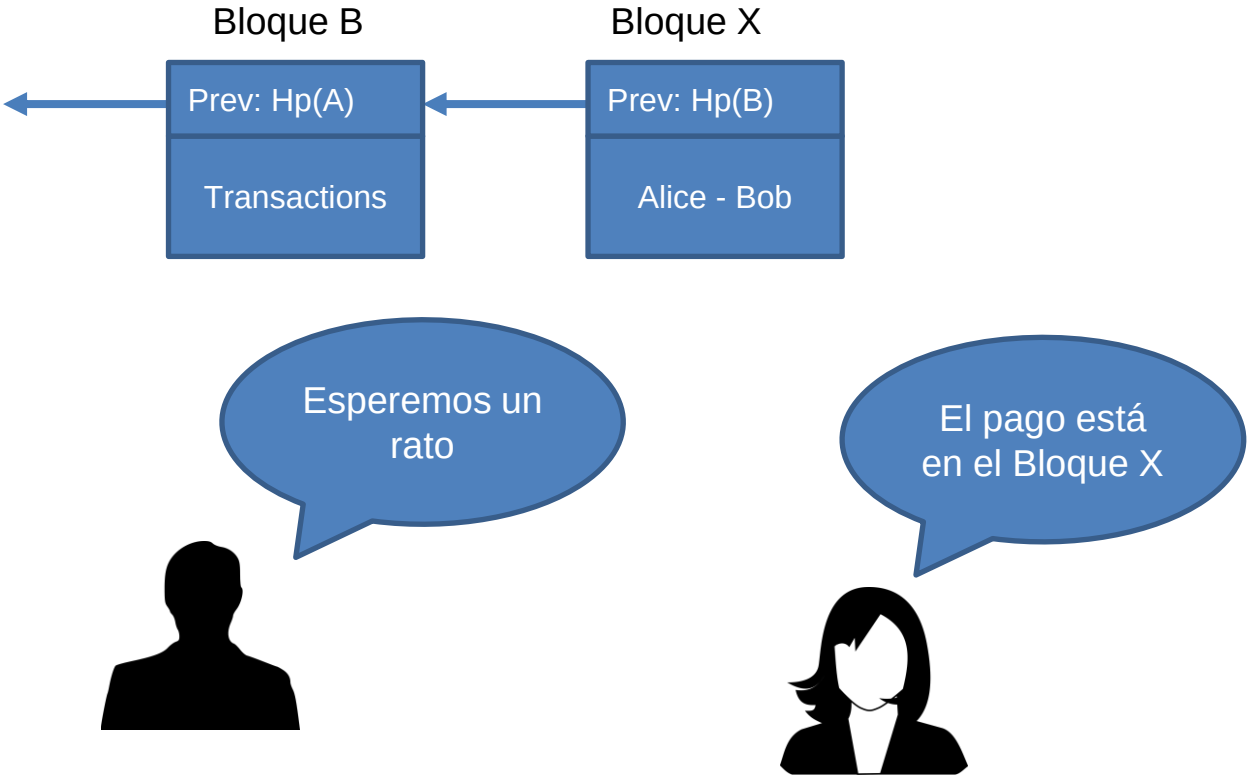
Múltiples confirmaciones



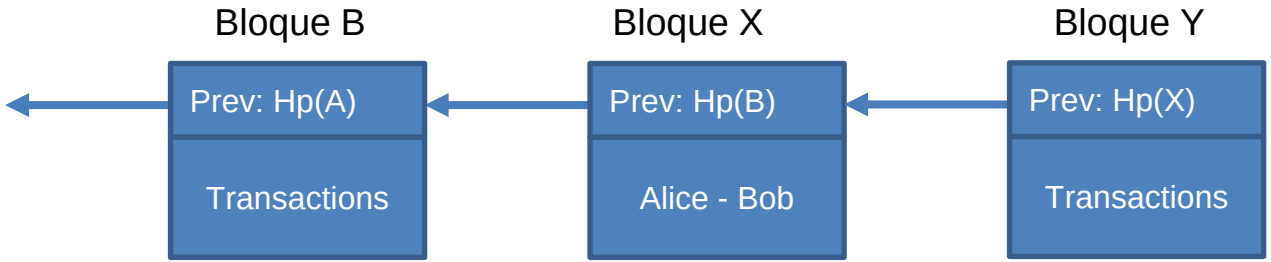
Múltiples confirmaciones



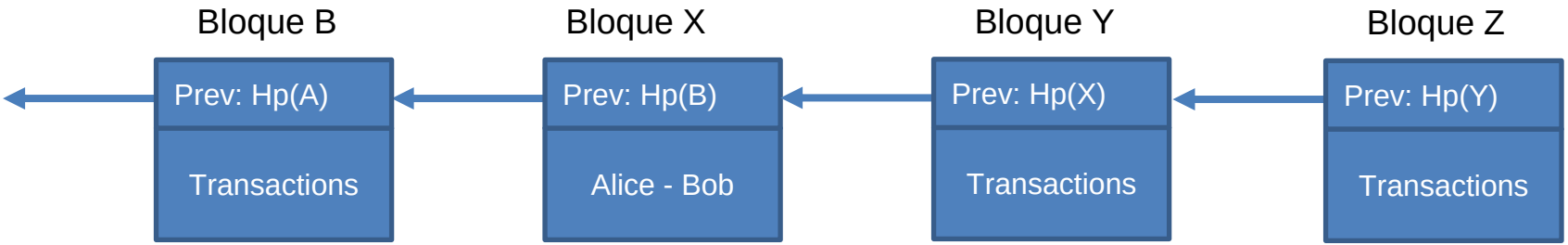
Múltiples confirmaciones

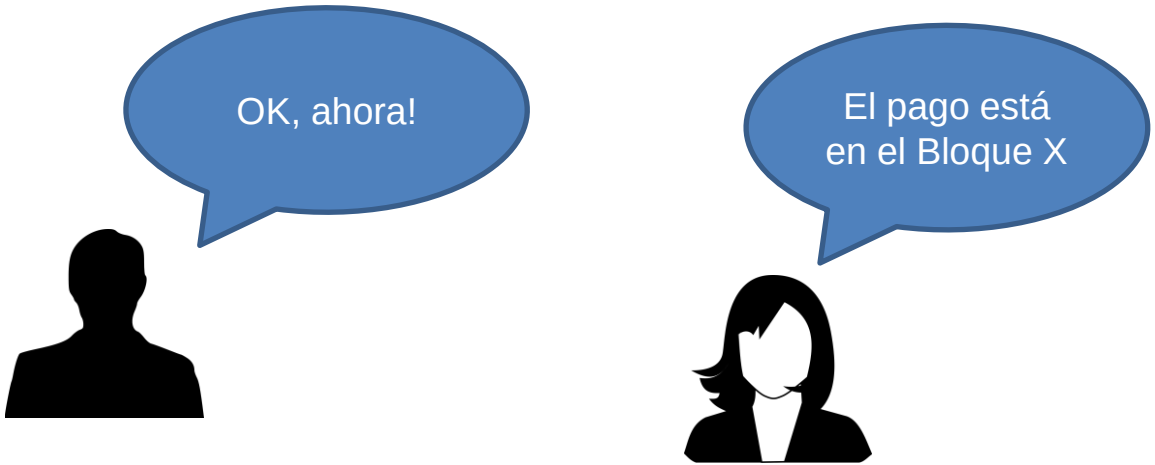


Múltiples confirmaciones



Múltiples confirmaciones





Propiedades de consenso implícito

Bob tiene varias opciones:

- Zero confirmaciones (doble gasto de bajo nivel)
- Una confirmación (doble gasto ya explicado)
- *k* confirmaciones (6 para Bitcoin = cca 1 hora)

Importante: 6 confirmaciones no garantiza que la transacción va a quedarse en el blockchain más largo. Pero con cada nueva confirmación la probabilidad de perder la plata disminuye exponencialmente. 6 es una buena heurística.

Propiedades de consenso implícito

Protección contra transacciones invalidas:

- Completamente criptográfico (firma está mala)
- Impuesto por el consenso (nodos buenos no incluyen esta transacción)

Protección contra el doble gasto:

- Completamente por el consenso
- La criptografía está buena en las dos ramas, una va a ganar

Qué nos falta?

Consenso implícito asume:

- Podemos elegir a un nodo aleatoriamente
- Al menos 50% de los nodos son honestos

Para remover estas suposiciones Bitcoin usa incentivos

Qué nos falta?

Consenso implícito asume:

- Podemos elegir a un nodo aleatoriamente
- Al menos 50% de los nodos son honestos

Para remover estas suposiciones Bitcoin usa incentivos

Cómo castigar a los nodos maliciosos?

Qué nos falta?

Consenso implícito asume:

- Podemos elegir a un nodo aleatoriamente
- Al menos 50% de los nodos son honestos

Para remover estas suposiciones Bitcoin usa incentivos

~~**Cómo castigar a los nodos maliciosos?**~~

Qué nos falta?

Consenso implícito asume:

- Podemos elegir a un nodo aleatoriamente
- Al menos 50% de los nodos son honestos

Para remover estas suposiciones Bitcoin usa incentivos

~~**Cómo castigar a los nodos maliciosos?**~~

Cómo premiar a los nodos buenos?

Incentivos en Bitcoin

Cómo premiar a los nodos buenos?

- **Con Bitcoins**

Dos tipos de incentivos:

1. **Block reward**
2. **Transaction fees**

Block rewards

Cada bloque en Bitcoin incluye una transacción especial:

- "Coinbase transaction" == CreateCoins de Scroogecoin
- Crea nuevos Bitcoins y los manda a la dirección del nodo que puso el bloque
- Premio para la creación de un bloque

Block rewards

Cada bloque en Bitcoin incluye una transacción especial:

- "Coinbase transaction" == CreateCoins de Scroogecoin
- Crea nuevos Bitcoins y los manda a la dirección del nodo que puso el bloque
- Premio para la creación de un bloque

El nodo puede hacer trampa?

Block rewards

Cada bloque en Bitcoin incluye una transacción especial:

- "Coinbase transaction" == CreateCoins de Scroogecoin
- Crea nuevos Bitcoins y los manda a la dirección del nodo que puso el bloque
- Premio para la creación de un bloque

El nodo puede hacer trampa?

- Incluir el bloque malo e igual cobrar el premio?

Block rewards

Cada bloque en Bitcoin incluye una transacción especial:

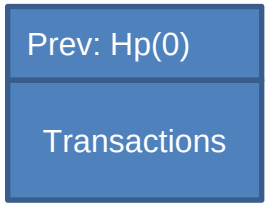
- "Coinbase transaction" == CreateCoins de Scroogecoin
- Crea nuevos Bitcoins y los manda a la dirección del nodo que puso el bloque
- Premio para la creación de un bloque

El nodo puede hacer trampa?

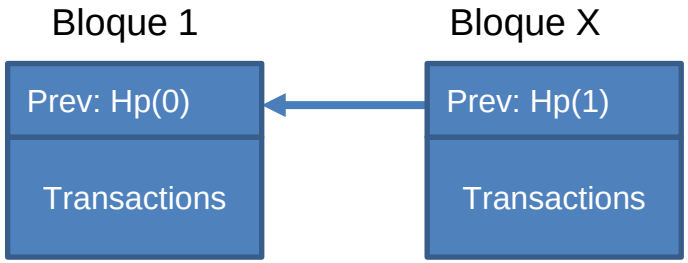
- Incluir el bloque malo e igual cobrar el premio?
- No, Coinbase es válido solo si tiene 100 confirmaciones
- Si el bloque está malo, el resto de la red no va a extender la cadena desde aquí
- Incentivo para extender el blockchain más largo (y válido)

Block reward

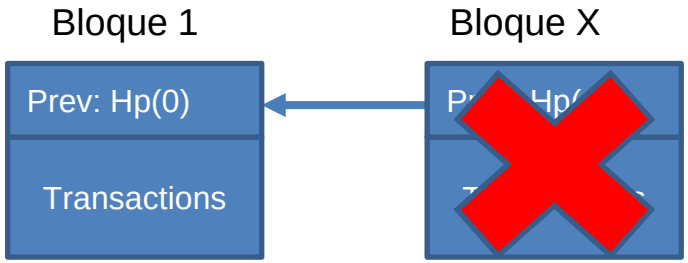
Bloque 1



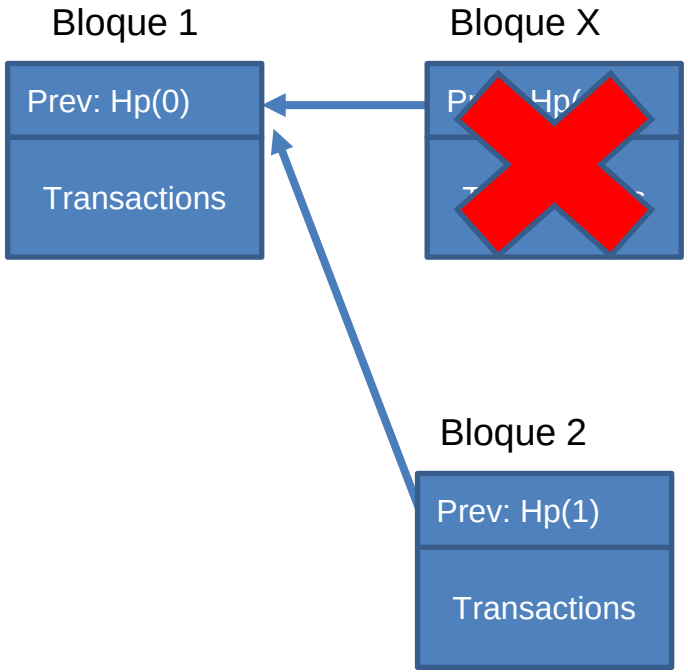
Block reward



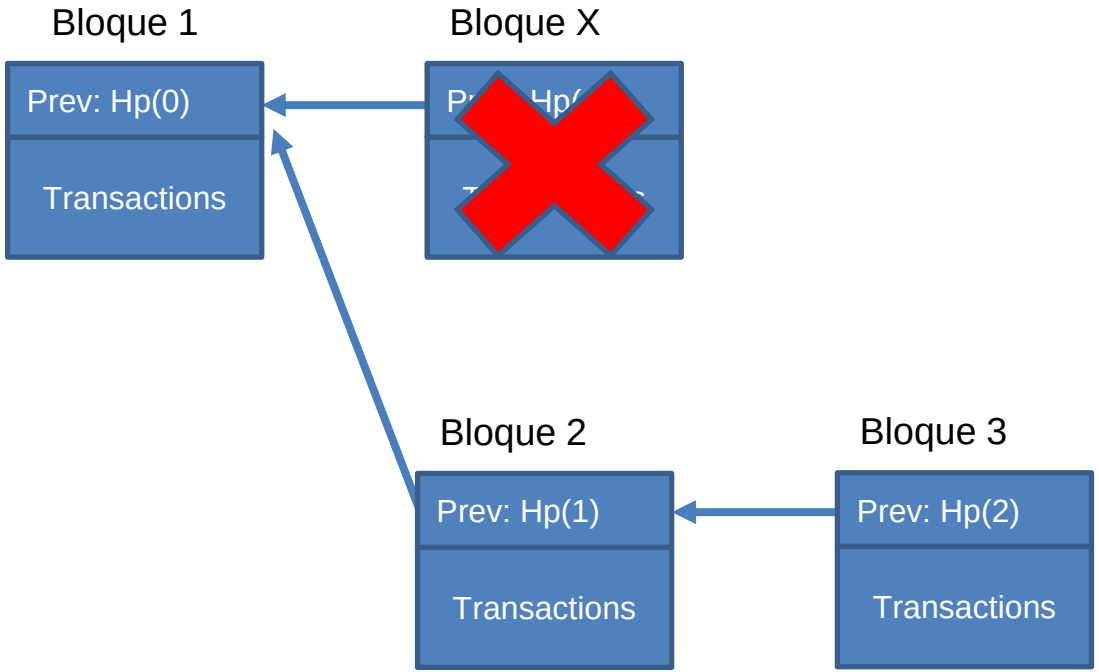
Block reward



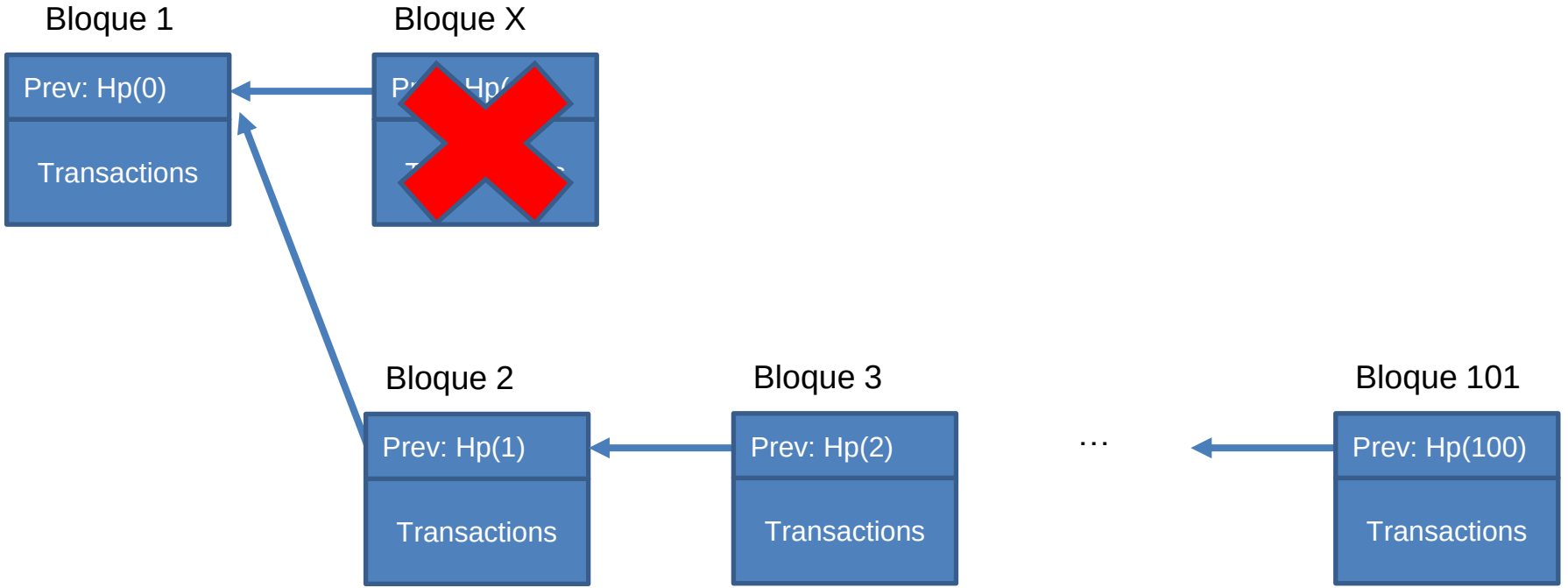
Block reward



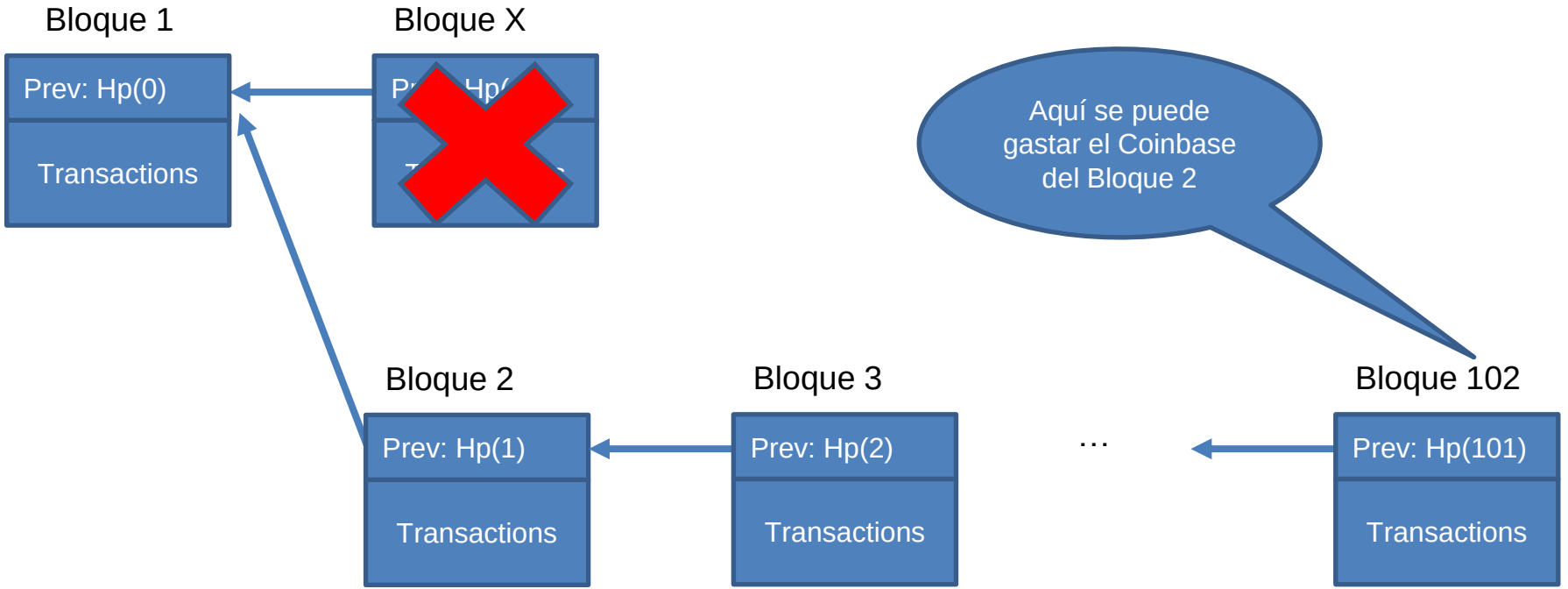
Block reward



Block reward



Block reward



Block rewards

La idea:

- Incentivar tener una rama
- El blockchain más largo (y válido) es el rey

Block rewards

Cantidad del reward está fija:

- En este momento 12.5 Bitcoins
- Se divide en 2 cada 210.000 bloques (cca. 4 años)
- Empezó con 50BTC, bajó dos veces: a 25 y 12.5 Bitcoins

Coinbase transaction:

- La única manera de crear un Bitcoin nuevo
- Esto es la emisión de Bitcoin como una moneda
- Proceso descentralizado (la red decide emitir nuevos Bitcoins en cada bloque)

Block rewards

Cantidad de Bitcoin es finita:

- 1 Bitcoin contiene 10^8 Satoshi
- La denominación más chica de BTC = 0.00000001 BTC
- El premio disminuye 50% cada 4 años

- En total va a haber cca. 21 millón de Bitcoins
- El Block reward se acaba en el año 2140

Block rewards

Por qué está bueno que hay una cantidad finita de Bitcoins?

Block rewards

Por qué está bueno que hay una cantidad finita de Bitcoins?

- Control de inflación
- Un recurso escaso (si todos pueden emitir nuevos Bitcoins no va a tener valor)

Block reward se acaba en 2140!!!

Transaction fees

Suma de valores de Inputs puede ser < suma de valores de outputs

- La diferencia se llama el "**transaction fee**"
- Se paga a la dirección especificada en el Coinbase transaction

- La idea es que esto sea una propina al nodo que propone el bloque
- Para asegurar la calidad de servicio

- Todos los transaction fees de todas las transacciones en un bloque van al minero

Transaction fees

Puede causar problemas:

- La transacción con más propina es más atractiva para incluir

Transaction fees

Puede causar problemas:

- La transacción con más propina es más atractiva para incluir
- Basic double spend: A→B fee: 0.01BTC; A→A fee 0.1BTC

Bloque tiene capacidad de 1MB:

- Tamaño de la transacción vs. fee
- Numero de transacciones vs. fee
- Una transacción con fee gigantesco
- Como formar el próximo bloque para maximizar el fee?

Pendiente

1. Cómo elegir un nodo para proponer el próximo bloque?
2. Cómo asegurarse que todos los nodos hacen algo útil?
3. Cómo prevenir ataque de sibilas?

Mining via proof of work

Consenso implícito

Asumamos que se puede seleccionar un nodo de manera aleatoria evitando el ataque de sybils (i.e. podemos detectar nodos replicados).

<u>El protocolo:</u>

1. Nodos escuchan/reciben nuevas transacciones
2. Cada nodo agrupa transacciones en un bloque
3. Cada 10 minutos se elige **aleatoriamente** un nodo para proponer su bloque
4. Otros nodos van a aceptar el bloque solo si está válido
5. Aceptación se expresa extendiendo el blockchain desde este bloque

Proof of work

Selección de un nodo aleatorio vamos a simular:

- Usando un recurso que no se puede monopolizar (fácilmente)
- Vamos a seleccionar al nodo usando su proporción de este recurso

Recurso usado en Bitcoin:

- Poder computacional (numero de hashes por segundo)
- Los nodos van a ser seleccionados basado en la cantidad de pega que hacen

Proof of work

Selección de un nodo aleatorio vamos a simular:

- Usando un recurso que no se puede monopolizar (fácilmente)
- Vamos a seleccionar al nodo usando su proporción de este recurso

Recurso usado en Bitcoin:

- Poder computacional (numero de hashes por segundo)
- Los nodos van a ser seleccionados basado en la cantidad de pega que hacen

Resuelve ataque de sibilas

Proof of work

Vamos a usar hash puzles:

- Todos los nodos comparten una dificultad *target*
- Los nodos buscan a un nonce t.q.

$$H(nonce||prev\_hash||tx\_1||tx\_2||...||tx\_n) < target$$

El primer nodo que encuentra el *nonce* publica el siguiente bloque!!!

Proof of work

Con hash puzles:

- No es necesario elegir un nodo aleatorio
- Aleatoriedad se simula usando hash puzles
- La competencia no depende de una entidad central
- Nadie decide el nodo, es una competencia justa (dado que hash es casi aleatorio)

Nodos competiendo para encontrar el nonce se llaman **mineros**

El proceso de solucionar el hash puzzle = **Bitcoin mining**

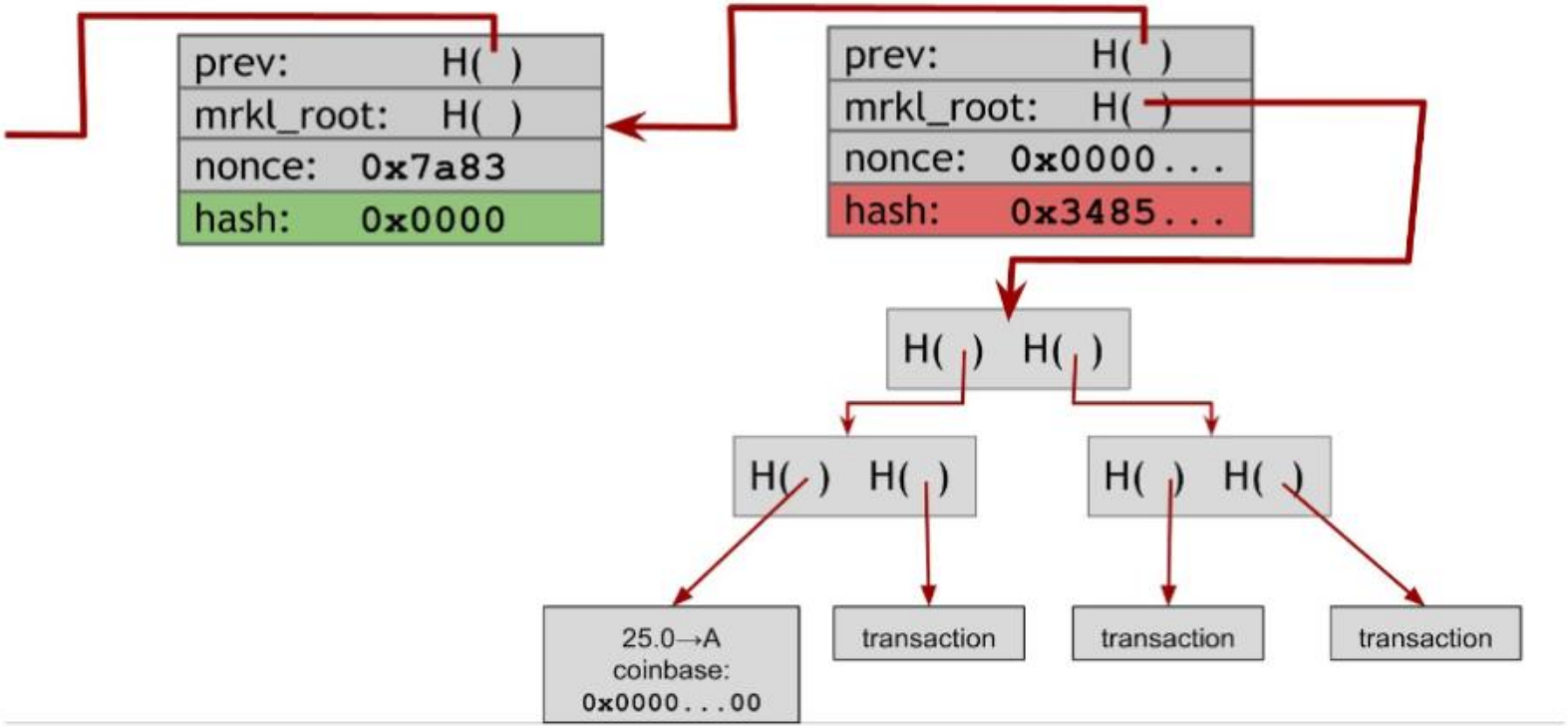
Proof of work

Qué se hashea realmente en Bitcoin?

- **Block header**

Field	Purpose	Updated when...	Size (Bytes)
Version	Block version number	You upgrade the software and it specifies a new version	4
hashPrevBlock	256-bit hash of the previous block header	A new block comes in	32
hashMerkleRoot	256-bit hash based on all of the transactions in the block	A transaction is accepted	32
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC	Every few seconds	4
Bits	Current target in compact format	The difficulty is adjusted	4
Nonce	32-bit number (starts at 0)	A hash is tried (increments)	4

Como funciona esto?



Fuente de la imagen: Bitcoin and Cryptocurrency Technologies, Naranyanana et. al. (2016)

Proof of work

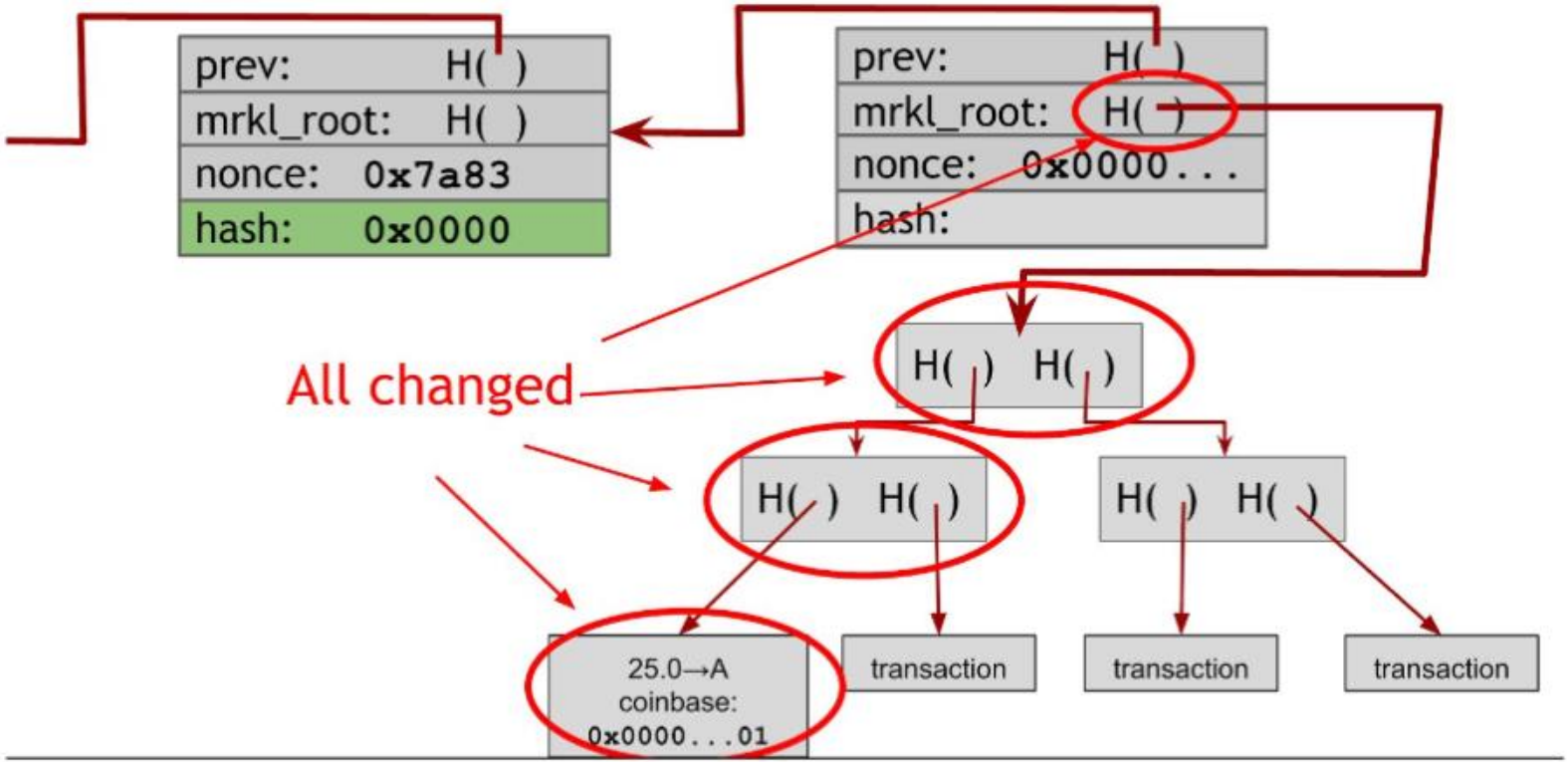
Qué se hashea realmente en Bitcoin?

- **Nonce tiene 32 bits!!!**
- **Target (actual) requiere en promedio 2^{75} bits!!!**
- **Campo extra nonce en el CoinBase!!!**

Difficulty in block header is a bit different!!!

Como funciona esto?

Updating the extra nonce



Fuente de la imagen: Bitcoin and Cryptocurrency Technologies, Naranyanana et. al. (2016)

Proof of work

La dificultad se ajusta automáticamente:

- Acuérdense: un bloque cada 10 minutos
- Si hay más mineros, bloques se encuentran más rápido
- Hay un reajuste de dificultad cada 2016 bloques

$$nextTarget = \frac{currentTarget \cdot 2016 \cdot 10\,min}{time\ to\ mine\ last\ 2016\ blocks}$$

Proof of work

Por qué 10 minutos?

- No hay un razón muy bueno
- Todos están de acuerdo que debería ser un tiempo agotado
- Y que no puede bajar arbitrariamente

Proof of work

Facil para verificar:

- Encontrar el *nonce* es dificil
- Verificar que el *nonce* está válido toma un computo de hash (fácil)

- Una vez publicado, validez de un bloque es fácil de verificar

Consenso en Bitcoin

Red p2p de pares. Red imperfecta, nodos entran y se van. Mensajes no se propagan de manera inmediata (delay). Nodos anónimos.

<u>El protocolo:</u>

1. Nodos escuchan/reciben nuevas transacciones
2. Cada nodo agrupa transacciones en un bloque
3. **Nodos buscan el nonce que resuelve el hash puzle (difícil)**
4. **Nodo que encuentra el nonce publica el siguiente bloque**
5. Otros nodos van a aceptar el bloque solo si está válido **(fácil)**
6. Aceptación se expresa extendiendo el blockchain desde este bloque

Consenso en Bitcoin

Seguridad:

- Más de 50% de los nodos son honestos

(El valor fue generado por un nodo honesto)

Mining

Porcentaje de hash power = porcentaje de bloques minados

- Si Alice tiene 0.1% de hash power, va a encontrar 1 bloque en 1000
- En el largo plazo

- Si Alice tiene 1000 veces más hash power que Bob
- Alice = 99.9%, Bob = 0.01%
- Bob encuentra un bloque en 10000, Alice el resto
- No es la verdad que Alice ganará siempre!!!!

Mining

Mineros:

- Ganan dinero con Bitcoin
- Pero tienen costos de hardware
- Tienen costos de electricidad

Uno espera un equilibrio:

- Si mineros no ganan plata, se van a retirar (el resto va a ganar más)
- Si es rentable minar más gente va a minar

Consenso es el Rey

Alice tiene 10 BTC:

- Significa que la red de Bitcoin está de acuerdo qué las direcciones controladas por Alice tienen 10 BTC

- El consenso puede cambiar en el futuro, pero es muy poco probable si las transacciones ya están confirmadas muchas veces

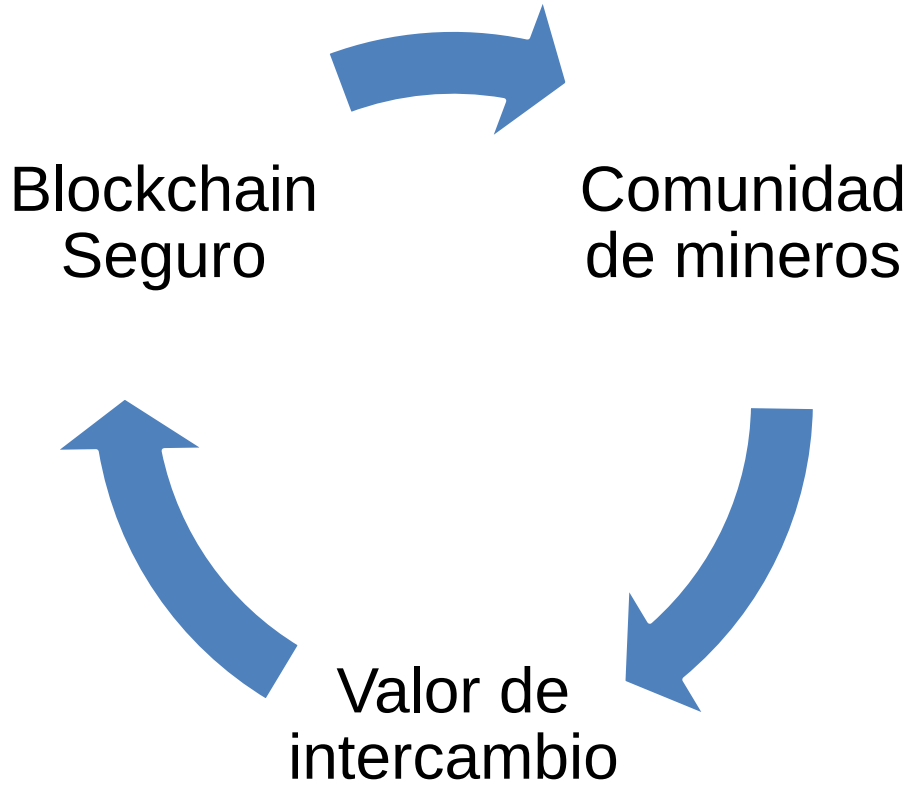
Bootstrapping

Qué necesita una criptomoneda para ser exitosa:

- Un blockchain seguro
- Una comunidad de mineros sana
- Valor en USD

Cuándo existen estas cosas?

Bootstrapping



Bootstrapping

Como partió Bitcoin:

- Con solo Satoshi
- Blockchain tenía una persona minando
- Cualquier persona con maquinas poderosas podía atacar a la red
- No había nada de valor en USD

Bootstrapping

Razones por éxito de Bitcoin?

- Media attention
- Más gente interesada en minar
- Blockchain más seguro
- Más valor
- …
- Más gente interesada
- Más seguridad
- Más valor
- …

51% attack

Qué puede lograr un ataque de 51%?

1. Robar los Bitcoins? (no, otros nodos no lo aceptan)

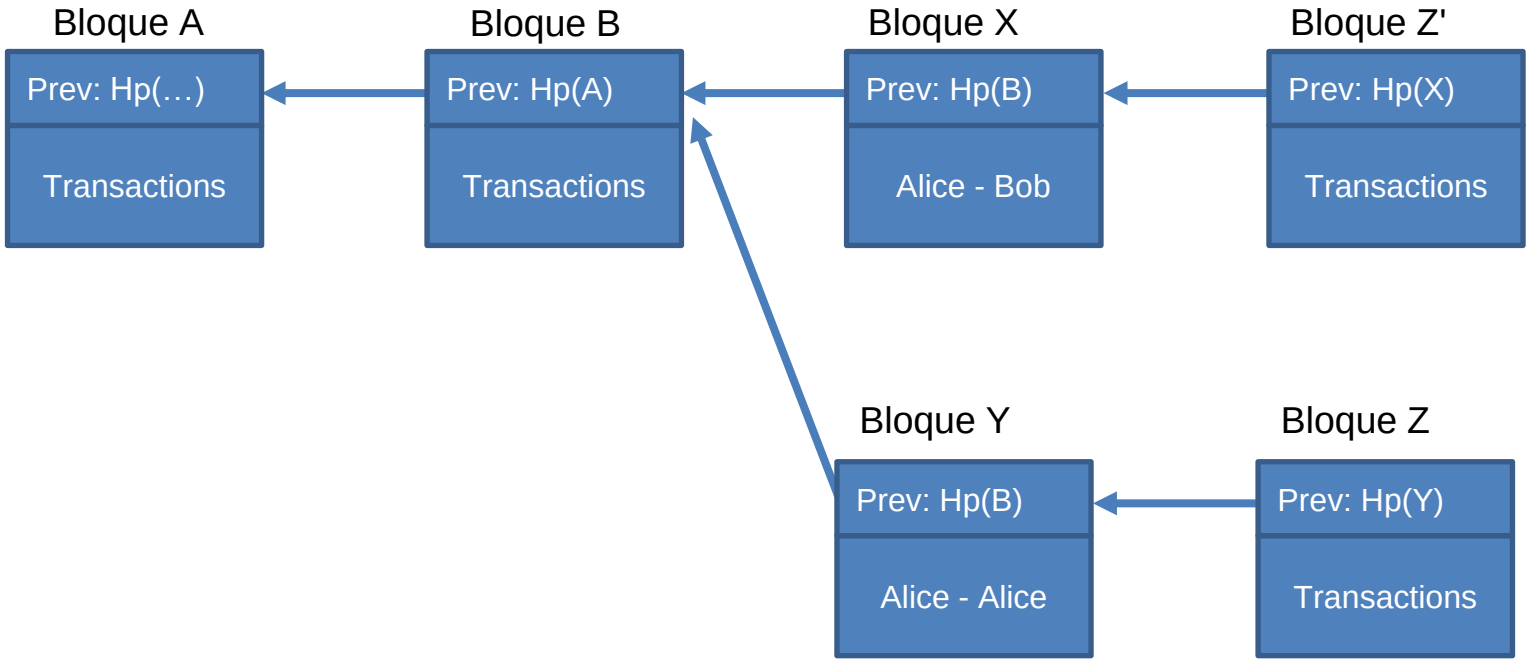
2. Blacklisting? (si, pero es fácil darse cuenta de esto si es enforzado)

3. Cambiar el premio? (no, otros nodos no lo aceptan)

4. Destruir la confianza en el sistema? (si)

Forks

Dos (o más ramas en el blockchain)



Forks

Un cambio de reglas:

- Todos los nodos deberían cambiar a su software
- No ocurre naturalmente

- **Hard forks vs. Soft forks**

Forks

Hard fork:

- Un cambio que no es forward compatible
- Se introducen reglas que antes no eran validas

- E.g. hay que firmar el doble hash de la transacción H(H(tx)), y no solo el H(tx)

- Nodos que hicieron el upgrade están bien, pero los antiguos no
- El blockchain se divide

- Ejemplo: Bitcoin Cash

Forks

Soft fork:

- Un cambio que es forward compatible
- Introduce reglas más estrictas

- Nodos con software antiguo van a aceptar todos los nuevos bloques
- Nodos con software nuevo van a rechazar algunos bloques antiguos
- Mineros antiguos pueden darse cuenta que hay que hacer el upgrade

- Ejemplo: (casi) cualquier BIP; e.g. P2SH

Referencias

Narayanan et al., Capitulo 2