Libro púrpura: Protocolo de Incentivos para Desarrolladores

Título original: ***Mauve Paper: Developer Incentive Protocol***

Nebulas Research

Octubre de 2018
Versión: 1.0.0

Traducción: abril de 2019
Versión: 1.0.0

Tabla de contenidos

1 Introducción

Generalmente, cuando un desarrollador de software crea aplicaciones dentro de las plataformas más conocidas (como Windows[1], Linux[2], macOS[3], iOS[4], Android[5] etc.) lo hace buscando un beneficio económico en el sentido tradicional de la industria del software: mediante salarios pagados por las empresas para las cuales trabajan en relación de dependencia, mediante ingresos por licencias de aplicación, o bien mediante anuncios que se muestran dentro de sus aplicaciones. A su vez, las empresas que contratan a esos desarrolladores reciben ingresos —por la venta del software desarrollado por ellos—, que no comparten con sus trabajadores.

Veamos un ejemplo basado en el desarrollo bajo la plataforma macOS: un diseñador UI/UX que necesita hacer uso de *Sketch* debe pagar no sólo por los complementos de esa herramienta, sino que además debe pagarle a Apple[6] por el dispositivo necesario para correr la herramienta. Así, Apple se beneficia de ese *cliente cautivo* pero no comparte esos ingresos con los desarrolladores de la herramienta *Sketch*. Algo similar ocurre con la popular herramienta AutoCAD[7], por la cual es necesario pagar a Microsoft[8] o a Apple el sistema operativo necesario para poder utilizarla. En estos casos, el factor clave para que los usuarios elijan una plataforma es si la plataforma soporta las aplicaciones que los usuarios necesitan. Es decir, las aplicaciones de alta calidad son fundamentales para el desarrollo de una *plataforma de aplicaciones*.

Basándonos en las consideraciones anteriores, podemos decir que —hasta cierto punto— las plataformas de aplicaciones no toman en cuenta las necesidades de los desarrolladores, dañando así sus intereses.

Este escenario se replica también en la industria blockchain, en donde los desarrolladores de aplicaciones descentralizadas (DApps) están siendo ignorados por las plataformas para las cuales desarrollan. En 2014, la comunidad de Ethereum introdujo el concepto de *contratos inteligentes*, que extienden la capacidad del *blockchain* de una simple herramienta de registro para criptodivisas p2p a una verdadera plataforma de aplicaciones descentralizadas. Sin embargo, a semejanza de la industria central-

[1] https://www.microsoft.com/es-es/windows
[2] https://es.wikipedia.org/wiki/Linux
[3] https://es.wikipedia.org/wiki/MacOS
[4] https://es.wikipedia.org/wiki/IOS
[5] https://es.wikipedia.org/wiki/Android
[6] https://en.wikipedia.org/wiki/Apple_Inc.
[7] https://en.wikipedia.org/wiki/AutoCAD
[8] https://en.wikipedia.org/wiki/Microsoft

izada del desarrollo de software, no hubo mejoras significativas en las ganancias para los desarrolladores —entre otras cosas, porque no tienen forma de beneficiarse de la apreciación en el valor de las plataformas blockchain para las cuales desarrollan.

De forma general, las recompensas emitidas por cada nuevo bloque en un *blockchain* representan un incremento de valor para ese sistema, y la distribución de tales recompensas determina el camino de los incentivos en esos sistemas descentralizados. En nuestra opinión, el incremento del valor de un sistema *blockchain* proviene esencialmente del valor implícito de los datos de sus usuarios, algo que debería distribuirse entre todas las partes intervinientes, incluyendo a los desarrolladores de aplicaciones descentralizadas. Aun así, lo que se ve en la práctica, al menos en la mayoría de los blockchains basados en PoW[9] —representados por Bitcoin y sus clones— es que las recompensas por nuevos bloques se distribuyen únicamente entre mineros; en los blockchains basados en el algoritmo PoS[10], las recompensas se asignan mayormente a quienes son propietarios de grandes sumas de la criptodivisa asociada. Así, los intereses de los desarrolladores de aplicaciones descentralizadas se ven afectados.

Conceptualmente, una DApp es un conjunto de contratos inteligentes con una serie de funcionalidades específicas; un contrato inteligente es, a su vez, un protocolo computacional dirigido a facilitar, verificar o forzar digitalmente la negociación o la realización de un contrato. Estos contratos inteligentes permiten la realización de transacciones fiables sin necesidad de la intervención de terceros[11].

Desde el punto de vista de la arquitectura de estos sistemas, la mayoría de las DApps utilizan usualmente contratos inteligentes en el *backend*, mientras hacen uso de tecnologías más tradicionales para el *frontend*. Las DApps pueden presentarse como aplicaciones tradicionales de PC, como aplicaciones móviles o bien como aplicaciones web.

Creemos que la relación entre las plataformas de aplicaciones descentralizadas, sus desarrolladores y sus usuarios es sinérgica y simbiótica. En primer lugar, la emergencia de aplicaciones descentralizadas permitió la expansión de la comunidad de desarrolladores blockchain; existe un crecimiento sostenido en la cantidad de desarrolladores que intentan escribir DApps que cumplan distintos requerimientos y que les permitan beneficiarse económicamente de ellas. En segundo lugar, esos desarrolladores proveen riqueza a las DApps, expandiendo así los escenarios posibles de uso para los blockchains, y a su vez atraen más usuarios hacia esas plataformas. Finalmente, los usuarios de DApps son quienes guían la optimización y actualización

[9] Prueba de Trabajo, o *Proof of Work* en inglés
[10] Prueba de Participación, o *Proof of Stake* en inglés
[11] Cf. `https://es.wikipedia.org/wiki/Contrato_inteligente`

permanentes de estas plataformas, incrementando en paralelo la movilidad de los tokens en las plataformas, y logrando en conjunto el crecimiento y el desarrollo del sistema blockchain.

Es necesario notar que los desarrolladores descriptos aquí sólo refieren a quienes escriben aplicaciones para plataformas descentralizadas, no necesariamente desarrolladores para la plataforma Nebulas ni tampoco necesariamente desarrolladores dentro del sistema blockchain (por esa razón hacemos mención de *desarrolladores de DApps* en forma genérica). Así, en este documento usaremos el término *desarrolladores* como sinónimo para *desarrolladores de DApps* con el fin de evitar esa ambigüedad. También es importante notar que un desarrollador DApp puede ser también un *stake holder* que obtiene beneficios económicos por esa vía; sin embargo, debido a que no todos los desarrolladores tienen asegurado ese derecho ni esa posesión, no es posible afirmar que el *stake holding* sea un medio legítimo para el incremento del valor de un blockchain; los intereses del desarrollador pueden ser ignorados o vulnerados sin perjuicio de este hecho.

Es inapropiada la distribución de ganancias por el incremento de valor de una plataforma entre sus desarrolladores de aplicaciones; por un lado, los ingresos pertenecen a corporaciones centralizadas, y los desarrolladores de aplicaciones para esas plataformas no tienen forma de conocer los detalles de los ingresos, ni tampoco participar en la distribución de esas ganancias. Por otro lado, es difícil cuantificar la contribución de cada desarrollador al crecimiento de una plataforma dada, por lo que la equidad de un mecanismo de distribución tal sería cuanto menos difícil de garantizar.

Afortunadamente, esa situación se puede cambiar en la industria del blockchain, ya que cada llamada a un contrato inteligente por parte de los usuarios queda registrada de forma pública en el blockchain. Así, *es posible otorgar recompensas o incentivos a cada desarrollador de DApps simplemente mediante la cuantificación de cada contribución de una DApp dada*.

Un mecanismo ideal para la distribución de incentivos debe satisfacer algunas propiedades básicas:

- Equidad: el protocolo debe mantener la objetividad al recompensar a los desarrolladores; esto es, cada DApp debe tratarse equitativamente y sus usos deben ser evaluados de forma transparente y verificable. Aun así existe cierto margen para manipulaciones.

- Efectividad: las recompensas deben reflejar las preferencias de los usuarios; esto es, aquellas DApps que reciben más recompensas deben coincidir con las que son de uso más frecuente por parte de los usuarios, mientras las DApps con

recompensas bajas o nulas deben coincidir con aquellas que los usuarios evitan utilizar.

En este documento proponemos la implementación del Protocolo de Incentivos para Desarrolladores (*Developer Incentive Protocol*, o DIP), que apunta a otorgar recompensas e incentivos a los desarrolladores, habilitándolos a beneficiarse del desarrollo de nuestra plataforma de aplicaciones descentralizadas. Naturalmente, no existe un protocolo de incentivos para desarrolladores ideal debido a que la evaluación de los usuarios sobre las DApps son subjetivas y multidimensionadas. Así, el protocolo DIP presentado en este documento tiene todavía margen para su mejora. No obstante ello, el balance que deja este libro púrpura es innovativo, en el sentido de que la premisa es garantizar los intereses de los desarrolladores de DApps en términos de resistencia a la manipulación.

El diseño de DIP está basado en el sistema pre-existente *Nebulas Rank* (NR)[1] y se beneficia de algunas cualidades del mismo. Intuitivamente, la evaluación de las DApps se reduce a un proceso de votación en el sistema DIP: cada llamada a una DApp por parte de un usuario se trata como un voto, y la capacidad de voto de cada usuario es una función de su número NR[12]. Los desarrolladores obtendrán las recompensas por parte del sistema, eventualmente, de acuerdo a los resultados de la votación.

Más allá del análisis teórico del modelo del Protocolo de Incentivos para Desarrolladores, analizaremos también las medidas contra las posibles manipulaciones e ilustraremos la implementación del protocolo mencionado con ejemplos tales como el ajuste y la actualización de DIP.

> Nota: este documento se enfoca en la discusión del Protocolo de Incentivos para Desarrolladores; a raíz de ello, la información aquí presente es más avanzada y actualizada que la disponible en el *Libro Blanco Técnico* [2], cuya versión 1.02 se lanzó en abril de 2018. Comparado con la demostración conceptual llevada a cabo entonces, y luego de un año de verificaciones en la práctica, tenemos la confianza y la experiencia para crear algoritmos más rigurosos y soluciones más claras para este protocolo.

2 Antecedentes

El protocolo tratado en este libro púrpura se basa en un gran número de trabajos relacionados, y además completa o actualiza la información previamente disponible.

[12]Número adimensional que indica la valuación dada a ese usuario por el algoritmo Nebulas Rank

En esta sección trataremos esas obras relacionadas, que jugaron un rol significativo como referencia y guía para la confección de este libro púrpura.

2.1 Incentivos para el desarrollo de aplicaciones descentralizadas

Por lo que sabemos, en la actualidad ninguna plataforma descentralizada basada en blockchains ofrece mecanismos de incentivos a largo plazo para desarrolladores de DApps. En su papel de representantes de la blockchain 2.0, Ethereum supuso un gran avance al implementar contratos inteligentes Turing-completos: en su red aparecieron distintas aplicaciones, incluyendo juegos, apuestas, sistemas de *crowd sourcing*, préstamos, créditos, entre otros. En particular, el juego de *tokens* coleccionables CryptoKitties —a fines de 2017— y Fomo3D —en 2018— atrajeron la mayor atención.

En la actualidad, y tal como sucede con los ejemplos descriptos en el párrafo anterior, la mayoría de los desarrolladores de DApps obtuvieron sus utilidades únicamente a través del cobro de comisiones a sus usuarios, sin que les resultara posible beneficiarse del incremento del valor de Ethereum o de las recompensas por nuevos bloques.

Con esta falta de incentivos para los desarrolladores, el escenario de las aplicaciones descentralizadas también se vio afectado. Por ejemplo, implícitamente, puede existir una carencia total de DApps gratuitas debido a la dificultad de obtener ingresos por medio de ellas. Como resultado de esto, la cantidad, la calidad y la diversidad de DApps se ve afectada. En contraste, la implementación de un mecanismo efectivo para incentivar a los desarrolladores logra el propósito de atraerlos al ecosistema del desarrollo de DApps, lo que promueve la prosperidad y el desarrollo del ecosistema blockchain.

Hasta cierto punto, muchos sistemas blockchain emergentes comprenden la necesidad de implantar mecanismos de incentivos para construir sus ecosistemas. Por ejemplo, en el programa *Nebulas Incentive*, se desarrollaron más de 6781 DApps, y también se logró que un gran número de equipos de desarrollo puedan obtener beneficios de esas DApps de forma directa.

En paralelo, otros blockchains públicos lanzaron también programas de incentivos a corto plazo basados en una administración centralizada. Tales programas apuntan principalmente a publicitarse entre la comunidad, aunque con evaluaciones oficiales tomando un rol central, sin sustentabilidad a largo plazo.

2.2 Nebulas Rank

Nebulas Rank (NR)[1] cuantifica la contribución de cada cuenta al rendimiento económico total, y provee características para impedir la manipulación en las mediciones. En particular, Nebulas Rank introduce la **función Wilbur**, que tiene las siguientes propiedades:

Propiedad 1. *Para dos variables positivas dadas x_1,x_2, la suma de sus funciones es menor que la función de su suma.*

$$f(x_1 + x_2) > f(x_1) + f(x_2) \quad x_1 > 0, x_2 > 0 \tag{1}$$

Propiedad 2. *Para dos variables positivas dadas x_1,x_2, cuando estas tienden a infinito, la suma de sus funciones tiende a la función de su suma.*

$$\lim_{x_1 \to \infty, x_2 \to \infty} f(x_1 + x_2) = f(x_1) + f(x_2) \quad x_1 > 0, x_2 > 0 \tag{2}$$

Como base de NR, estas dos propiedades ofrecen también resistencia contra la manipulación.

2.3 Mecanismos de voto

Como se mencionó antes, el proceso por el cual los usuarios eligen y utilizan las DApps disponibles se puede ver como una forma de votar por ellas en el protocolo DIP. El mecanismo de los incentivos es similar al algoritmo de valuación (*ranking*). Con respecto a estos dos últimos, existen numerosos trabajos relacionados en distintos campos; a continuación brindaremos una muestra de ellos.

Uno de los estudios más famosos es el teorema de Arrow, que sostiene que, cuando los votantes tienen tres o más alternativas, no es posible crear un sistema de votación que refleje las preferencias de los votantes en una *preferencia global* mientras se satisfacen tres condiciones: independencia de alternativas irrelevantes (que la valuación relativa de dos candidatos no se vea afectada por la de un tercer candidato), ausencia de *dictadores* (que ningún candidato o individuo tenga el poder de cambiar las preferencias del grupo), y eficiencia de Pareto[13] (el resultado de la valuación satisface los

[13]Dada una asignación inicial de bienes en un grupo de individuos, una redistribución de esos bienes que mejora la situación de un individuo, sin hacer que empeore la situación de los demás, se denomina eficiencia de Pareto

intereses de la mayoría). El estudio implica que ningún algoritmo de valuación puede cubrir todas las posibilidades. Así, el Protocolo de Incentivos para Desarrolladores que se discute en este libro púrpura se enfocará en los atributos más importantes y conocidos.

En la vida real existe un sinnúmero de escenarios que requieren algoritmos de valuación. Un ejemplo claro para esta afirmación es la reputación dada por los compradores a los vendedores de Amazon, Taobao y otros sitios similares. Los vendedores con mayor reputación obtendrán mejor exposición en la plataforma, y gracias a ello obtendrán más visibilidad y mayor tasa CTR[14]. En particular, existen problemas similares en estas plataformas de comercio electrónico, como el ataque Sybil[15].

Por el momento, esas plataformas centralizadas dependen mayormente de la capacidad de aprendizaje de sus sistemas para distinguir entre transacciones normales y falsas[3, 4, 5]. De todas maneras, la práctica demostró que tales métodos no son los ideales. [6] señala que incluso la identificación mediante inteligencia artificial no puede distinguir entre ambas cuentas con efectividad. [7] expone un algoritmo que elimina el incentivo para tales manipulaciones basadas en el diseño del mecanismo. Aun cuando ese modelo difiere del nuestro, es posible utilizarlo como una referencia significativa.

[8] introduce un algoritmo de valuación para publicaciones en redes sociales que combina los votos de los usuarios y la declinación del tiempo.

[9] introduce un algoritmo de valuación para publicaciones en la red social Reddit, que implica una situación en la que los usuarios pueden emitir votos negativos.

[10] introduce el algoritmo de valuación de comentarios de la red social Reddit, que toma en cuenta el intervalo de confianza.

IMDB [11] presenta la idea de un Promedio Bayesiano de Modelos (BMA en inglés) para su sistema de valuación de filmes, que puede reducir la brecha entre las diferentes películas debido al número de votantes.

Gracias a las propiedades anti-manipulativas de NR, el Protocolo de Incentivos para Desarrolladores propuesto en este libro púrpura puede distinguir entre usuarios normales y usuarios falsos de una forma más clara. Así, el énfasis de este documento está en transferir el valor NR de los usuarios a los puntos de valuación de las DApps a través de comportamientos interactivos.

[14] *Click Through Rates*, o tasa de clics.

[15] Ataque que consiste en crear transacciones falsas para obtener revisiones de cinco estrellas.

3 Modelo de Incentivos para Desarrolladores

El Protocolo de Incentivos para Desarrolladores, (*Developer Incentive Protocol*, o DIP), incluye dos procesos: la valuación de las DApps y la distribución de los incentivos.

Por un lado, la construcción de un sistema de valuación robusto puede brindarle a los desarrolladores una plataforma conveniente y efectiva para promover aplicaciones y brindarles a sus usuarios un sistema de recomendaciones fiable; tal como sucede en la plataforma App Store, las buenas aplicaciones se nmuestran primero en los listados y, debido a ello, reciben más atención de los usuarios. Por otro lado, los usuarios reciben una mejor experiencia cuando encuentran aplicaciones de alta calidad directamente en ese listado. Más aun, la valuación de las aplicaciones puede ser utilizada en la búsqueda por palabras clave.

Tal como ocurre en los motores de búsqueda, y en las plataformas de *e-commerce*, el listado de aplicaciones ordenado por palabras clave y de acuerdo a su valuación en los resultados de la búsqueda contribuye a la satisfacción de los usuarios.

Como vimos en la sección 2, el propósito del Protocolo de Incentivos para Desarrolladores es el de brindar incentivos a todos los desarrolladores de aplicaciones de calidad, aumentando así los incentivos de esos desarrolladores para diseñar buenas DApps y promoviendo, en paralelo, el desarrollo del ecosistema.

De esto se desprende que el segundo proceso en el modelado del Protocolo de Incentivos para Desarrolladores es el diseño de un mecanismo equitativo de incentivos, en concordancia con la valuación de cada DApp.

3.1 Representación del modelo

En esta sección presentamos la notación y símbolos necesarios en el modelo DIP.

- $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ representa el conjunto de usuarios que participan en el sistema de valuación durante un periodo dado. Utilizamos votantes para denotar a esos usuarios. Nótese que un usuario está definido como votante únicamente si realiza una llamada a algunas DApps EOA (Cuenta de Propiedad Externa, o *External Owned Account* en inglés) dentro de un periodo dado. Defínase

  $$\mathcal{A}^* = \{a_1, a_2, \ldots, a_m, a_{m+1}, \ldots, a_{m^*}\}$$

  como el conjunto de todos los usuarios en la comunidad durante el mismo periodo de tiempo; esto es, $m^* - m$ usuarios que no invocan ninguna DApp.

- $\mathcal{D} = \{d_1, \ldots, d_n\}$ representa el conjunto de DApps durante un periodo de tiempo dado.

- $e_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$ representa la cantidad de veces que el votante a_i invoca la DApp d_j. Debido al carácter público y descentralizado del blockchain, el modelo DIP difiere del de otros sistemas de valuación en mercados de aplicaciones centralizadas. Generalmente hablando, DIP valúa las DApps de acuerdo al comportamiento de las invocaciones realizadas por los usuarios en un entorno descentralizado. Los detalles del análisis se brindan en la siguiente sección de este documento.

- $\Gamma_i, i = 1, 2, \ldots, m$ representa la capacidad de voto de un votante a_i durante un periodo de tiempo dado. [1] demuestra que el valor NR de un usuario es una medida efectiva del valor de su cuenta. Así, en DIP, NR se utiliza también como un criterio significativo para decidir la capacidad de voto de los votantes.

- $\Gamma_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$ representa el valor contributivo de un votante a_i a una DApp d_j, que puede considerarse como el número de votos que a_i está dispuesto a otorgar a d_j.

- $S_j, j = 1, 2, \ldots, n$ representa el puntaje de la valuación de una DApp d_j, que está determinado por el valor recibido por parte de todos sus votantes. Intuitivamente, el puntaje de valuación determina la posición de las DApps en la lista de valuación.

- M representa el total del fondo de incentivos para desarrolladores, que tiene su origen en las recompensas por nuevos bloques. La recompensa real se reducirá adecuadamente de acuerdo con la tasa de participación de toda la comunidad durante el período de tiempo dado.

- $U_j, j = 1, 2, \ldots, n$ representa el incentivo final de la DApp d_j, que se determina por el total disponible en el fondo de incentivos, y por el puntaje de valuación de las DApps.

Para sumarizar, las interacciones entre votantes y DApps se puede representar mediante un gráfico bipartito en la figura 1.

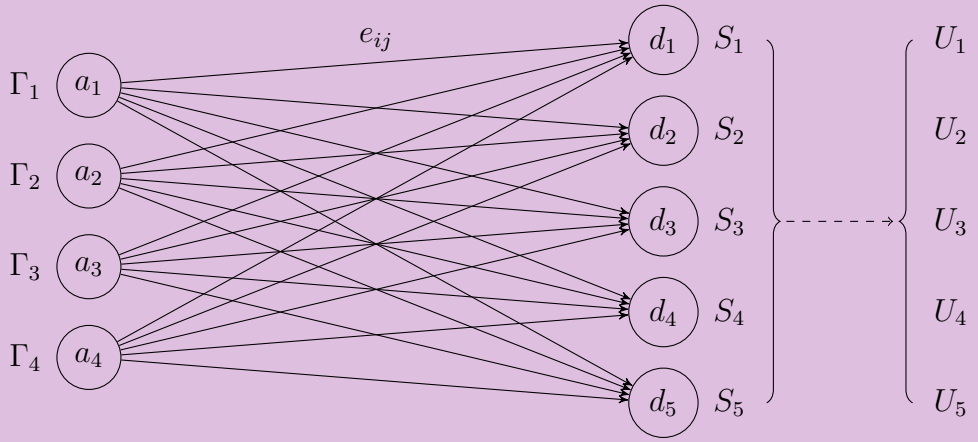

Figure 1: Interacciones entre votantes y DApps

3.2 Acción de voto

En cualquier App Store centralizada[16], el sistema puede almacenar información específica tal como la cantidad de veces que una App fue descargada, lo cual es un factor clave para determinar su valuación. No obstante, en el mundo del blockchain, la forma en la que los usuarios utilizan las DApps es mediante la invocación de las direcciones de contratos inteligentes; esto se puede cuantificar mediante la cantidad de veces que un usuario a_i invoca la DApp d_i, algo denotado por e_{ij}. Comparado a la información tradicional acerca de cantidad de descargas, la metodología que usa DIP, de tomar la información de las llamadas a DApps como parámetro de valuación, tiene las siguientes ventajas:

- El número de llamadas se almacena en el blockchain, que es muy resistente al fraude, es más abierto y transparente, en comparación con otros métodos centralizados tales como el registro del número de descargas.
- El número de llamadas es un parámetro más detallado que la cantidad de descargas, ya que el número de descargas sólo registra el comportamiento de un usuario por única vez, mientras que el registro de cantidad de llamadas a una DApp señala, además, la fidelidad de ese usuario con la aplicación. En consecuencia, el número de llamadas es un parámetro más razonable para reflejar el comportamiento real de los usuarios.

En realidad existe más información disponible cuando los usuarios invocan una llamada a las DApps. Por ejemplo, la cantidad de *gas* que se ha utilizado, y las trans-

[16] `https://en.wikipedia.org/wiki/App_store`

ferencias de tokens involucradas en una transacción. Sin embargo, DIP no toma estos datos en consideración por dos motivos.

Primero, la cantidad de *gas* consumido depende de la cantidad de instrucciones ejecutadas en el contrato inteligente cada vez que un usuario invoca una DApp, algo que no guarda relación con la calidad intrínseca de la DApp en cuestión. Más aun, en el sistema actual de Nebulas, la cantidad promedio de *gas* consumido durante cada invocación está en el orden de 10^{-8} NAS, algo insignificante.

Segundo, la razón por la que no se toman en cuenta las transferencias de tokens es la falta de un método efectivo contra la manipulación. Intuitivamente sabemos que la voluntad de los usuarios de pagar tokens adicionales cada vez que llaman a una DApp es algo que mejora la valuación de esta. Sin embargo, en la práctica, cuando un usuario decide pagar tokens para utilizar una DApp, el destino final de esos tokens puede ser cualquiera de estos tres casos:

1. Los tokens quedan finalmente en posesión del desarrollador de la DApp. En este caso, es probable que el usuario pague voluntariamente por el uso de la DApp. Como su desarrollador recibe beneficios por parte del usuario, no es relevante incrementar aun más la valuación de su DApp.

2. La misma naturaleza de la DApp requiere una transferencia de tokens —por ejemplo DApps de apuestas— lo que lleva a una gran cantidad de tokens transferidos entre el usuario y la DApp, algo que es normal y aceptable. Sin embargo, la valuación de estas DApps no se debería incrementar por ello, debido al hecho de que el propósito de esas transferencias es la de obtener ganancias, algo que no refleja la calidad de la DApp.

3. El desarrollador de la DApp afirma que todos los tokens recibidos por la DApp serán devueltos al usuario. Esto constituye en sí mismo un acto de manipulación, que se agravaría si la DApp recibe recompensas y valuación por ello.

En la práctica, sin analizar el código fuente de los contratos inteligentes, no es posible determinar en cuál de estos casos cae cada transferencia de tokens entre usuarios y DApps y en todo caso en ninguno de ellos se amerita otorgar incentivos, de modo que el algoritmo del DIP no tomará en cuenta la transferencia de tokens.

En el modelo DIP, un usuario $a_i \in \mathcal{A}$ se ve esencialmente como una dirección Nebulas. Tal como se refiere en [1], un único usuario puede controlar múltiples direcciones. Debido a que no hay costo alguno en la creación de una dirección Nebulas, el usuario podría crear numerosas direcciones con el fin de votar, algo que se conoce como el ataque Sybil. Similarmente, un desarrollador podría dividir sus DApp en

distintas direcciones —esto es, dividir su DApp en distintas DApps de baja calidad, y obtener recompensas por todas las DApps asociadas a esas direcciones. Mientras tanto, un desarrollador podría pagarle a uno o más usuarios para votar por su DApp.

Hemos analizado todas las manipulaciones descriptas más arriba al momento de diseñar DIP, y les hemos dado solución. Los detalles de los mecanismos antifraude de DIP se encuentran en la sección 5.

3.3 Intervalo de muestreo

En la sección 3.1, hemos mostrado que NR es un criterio significativo para determinar la capacidad de voto de cada usuario. Sin perjuicio de ello y de acuerdo a la definición en [1], el periodo de muestreo de los datos para el DIP es mucho mayor que el necesario para NR, lo que implica que durante el proceso de registro del comportamiento de las invocaciones de los usuarios, el valor NR podría fluctuar, incluso significativamente.

Un enfoque simplista es sincronizar el período de muestreo de los datos NR y los datos DIP. En la práctica, sin embargo, se ha visto que la utilización de periodos cortos de tiempo (un día, por ejemplo), son insuficientes para la mayoría de los usuarios, ya que muchos de ellos realizan invocaciones en periodos de tiempo mayores, de modo que tiene poco sentido valuar las DApps de este modo cuando el comportamiento de los usuarios es disperso, y no hay garantía de que se satisfagan las condiciones enumeradas en la sección 5.

De este modo, nuestra estrategia es la de extender apropiadamente el periodo de muestreo para recoger suficiente información sobre el comportamiento de las invocaciones, y encuadrar la variación del valor NR de los usuarios de forma simultánea. La variación del valor NR de una dirección se muestra en la figura 2. Allí dividimos el proceso completo de DIP en distintos periodos de tiempo. De acuerdo a los datos acerca de las variaciones de NR, se elige un entero t tal que la variación de NR dentro de t días menos que el umbral τ es apto para la mayoría de los usuarios. Tomamos t días como un periodo de muestreo, reuniendo el promedio de valores NR de los votantes y los datos de invocación durante ese periodo para computar el puntaje de valuación de las DApps y las recompensas finales para sus desarrolladores. Luego tomamos los datos medios de todos los períodos de tiempo durante el proceso de valuación como resultados finales.

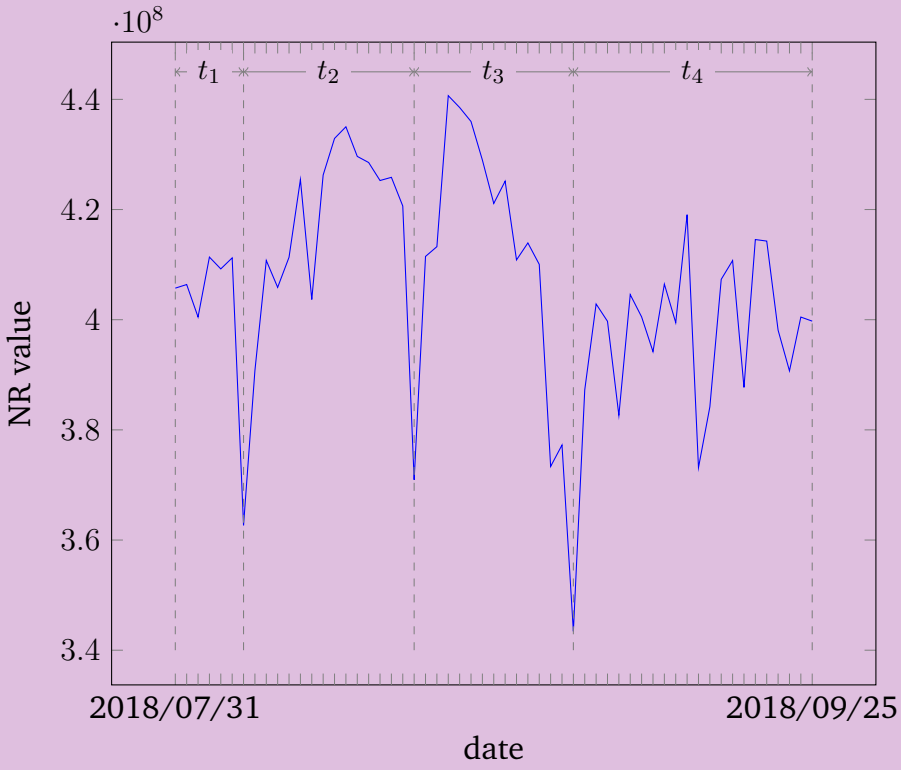

Figure 2: Gráfico de la variación de NR de una dirección en la red Nebulas. La dirección mainnet es `n1Ugq21nif8BQ8uw81SwXHK6DHqeTEmPRhj`.

4 Protocolo de Incentivos para Desarrolladores

Basándonos en el modelo de la sección anterior, en esta presentaremos el Protocolo de Incentivos para Desarrolladores (DIP, por sus siglas en inglés, *Developer Incentive Protocol*). DIP se compone de dos procedimientos: la valuación de las DApps y la incentivación de los desarrolladores. Específicamente, desde el comportamiento de invocación por parte de los votantes hasta la recepción de los incentivos por parte de los desarrolladores, DIP incluye cuatro transformaciones: tiempos de invocación $\xrightarrow{1}$ capacidad de voto $\xrightarrow{2}$ valor de contribución $\xrightarrow{3}$ puntaje de valuación $\xrightarrow{4}$ incentivo final.

4.1 Capacidad de voto y Valor de Contribución

Para todo votante a_i, utilizamos Γ_i para denotar su capacidad de voto, que se puede expresar como el número total de votos que posee el votante a_i. [1] demuestra que el valor NR de un votante es una medida efectiva del valor de su cuenta. Así, en DIP, el valor NR es usado también como un criterio significativo para decidir la capacidad de voto de los votantes. Para un votante a_i, su capacidad de voto se puede representar

como una función de su valor NR:

$$\Gamma_i = f(\mathcal{C}(a_i)) \tag{3}$$

donde $\mathcal{C}(a_i)$ representa el valor NR del votante a_i.

Normalmente, querríamos que f sea una función creciente, esto es, votantes con mayor valor NR tienen mayor capacidad de voto. Aquí brindamos una función que satisface esa condición:

$$f(\mathcal{C}(a_i)) = \mathcal{C}^2(a_i). \tag{4}$$

En otras palabras,

$$\Gamma_i = \mathcal{C}^2(a_i). \tag{5}$$

De ello surge que esta función provee propiedades interesantes, como ser fortaleza contra ataques Sybil. Véase la sección 5.

Luego es necesario discutir el mecanismo de distribución de la capacidad de voto. De acuerdo a la sección 5, Γ_{ij} representa el valor de contribución de un votante a_i a una DApp d_j. Lo definimos de esta manera:

$$\Gamma_{ij} = \frac{e_{ij}}{e_{i0} + \sum_{j=1}^{n} e_{ij}} \Gamma_i. \tag{6}$$

La fórmula 6 puede ser entendida como la proporción entre la cantidad de invocaciones a d_j y el total de invocaciones. Aquí e_{i0} representa la cantidad de invocaciones que no corresponden a ninguna DApp. Un votante puede ajustar arbitrariamente los valores de e_{i0} y e_{ij}.

Con la introducción de e_{i0}, se desprende que

$$\sum_{j=1}^{n} \Gamma_{ij} \leq \Gamma_i.$$

Es importante notar que el propósito de la fórmula6 es el de permitirles a los votantes distribuir arbitrariamente sus valores NR para votar (mediante la elección arbitraria de valores de contribución). En la práctica es posible que algunas DApps aumenten por la fuerza el número de invocaciones (al forzar a los usuarios a enviar el doble de ellas), pero como tanto el número de invocaciones como los valores NR son de do-

minio público, un votante puede todavía lograr la distribución valores de contribución que desea simplemente ajustando sus tiempos de invocación.

La razón detrás de la introducción de e_{i0} es que, para el resto de los IR (*Individual Rationals*) de los votantes —los intereses de los votantes no serán vulnerados— no forzamos a los votantes a emitir todos sus votos. Los votantes pueden ejercer selectivamente parte de sus derechos de voto o bien abstenerse por completo ajustando apropiadamente el valor de e_{i0}.[17]

4.2 Puntaje de valuación

Dados todos los valores de contribución de los votantes a las DApps, es posible computar el puntaje de valuación de las DApps: dados todos los valores de contribución $\Gamma_{ij}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$, definimos el puntaje de valuación de una DApp d_j como una función de múltiples variables sobre los valores de contribución de todos los votantes:

$$S_j = g(\Gamma_{1j}, \Gamma_{2j}, \ldots, \Gamma_{mj}) \tag{7}$$

Similarmente, la siguiente función satisface esas condiciones:

$$g(\Gamma_{1j}, \Gamma_{2j}, \ldots, \Gamma_{mj}) = \sum_{i=1}^{m} \sqrt{\Gamma_{ij}} \tag{8}$$

Esto es, la valuación de una DApp en el Store equivale a la suma de las raíces cuadradas de los valores de contribución de todos sus votantes. No es difícil ver que, para un usuario a_i, cuando sólo vota por una DApp (asumiendo que no deshecha ningún voto), su valor de contribución total es $\sqrt{\Gamma_i}$. Cuando emite sus votos a distintas DApps, su valor de contribución total se incrementa debido a la propiedad $\sqrt{a+b} < \sqrt{a} + \sqrt{b}$ de la función raíz cuadrada. En otras palabras, el votante se pone en contacto con más DApps, algo alentado por nuestro sistema. En la sección 5, probaremos las propiedades detalladas para nuestra construcción. Un método similar está referenciado en [12].

Dados los puntajes de valuación $S_j, j = 1, 2, \ldots, m$, las valuaciones de las DApps están determinadas de forma acorde. Por ejemplo, en el cliente Nebulas nano[18], las

[17] e_{i0} se puede implementar estableciendo oficialmente un contrato inteligente vacío, que no ejecuta ninguna instrucción. Los votantes, entonces, pueden invocar ese contrato inteligente las veces que lo necesiten.

[18] `https://nano.nebulas.io/`

DApp de alta valuación son listadas en posiciones prominentes, lo que sirve para atraer más atención sobre ellas.

4.3 Incentivo final

DIP le ofrece a los votantes una lista confiable de valuación de DApps.[19] Para los desarrolladores, necesitamos distribuir las recompensas de acuerdo a sus puntos de valuación.

Dados todos los puntajes de valuación de las DApps, definimos el incentivo para el desarrollador de una DApp d_j por medio de:

$$U_j = \frac{S_j^2}{\sum_{k=1}^{n} S_k^2} \cdot \lambda M \tag{9}$$

donde M es el total del fondo de incentivos para desarrolladores, que proviene de las recompensas otorgadas por los nuevos bloques. λ se define como el factor de participación; esto es, deseamos que el total de incentivos se incremente junto al total de los valores NR de los votantes. La definición específica es como se muestra a continuación:

$$\lambda = \min\{\frac{\Gamma_p}{\alpha\Gamma_s} \cdot \min\{\frac{\beta\Gamma_p^2}{\sigma^2(\Gamma_p)}, 1\}, 1\}, \tag{10}$$

donde

$$\Gamma_p = \sum_{i=1}^{m}(\Gamma_i - \Gamma_{i0}), \quad \Gamma_{i0} = \frac{e_{i0}\Gamma_i}{e_{i0} + \sum_{j=1}^{n} e_{ij}},$$

(la capacidad total efectiva de voto de todos los votantes)

$$\Gamma_s = \sum_{i=1}^{m^*} \Gamma_i,$$

la capacidad total de voto (la raíz cuadrada del valor NR) de todos los usuarios en la comunidad, σ es la desviación estándar (raíz cuadrada de la varianza) de la capacidad total efectiva de voto de todos los votantes:

$$\sigma^2(\Gamma_p) = \sum_{i=1}^{m}(\Gamma_i - \frac{1}{m}\Gamma_p)^2$$

[19] Asumimos que los votantes únicamente sienten interés por la valuación de las DApps de su preferencia, al tiempo que las recompensas que sus desarrolladores reciben no son de su incumbencia.

de la cual su valor máximo es $\frac{(m-1)^2}{m^2}\Gamma_p^2$, y $\alpha, \beta < 1$ son parámetros ajustables.

El propósito de la introducción del factor de participación λ es el de esperar que la capacidad total de voto de los votantes supere un umbral (α veces la capacidad total de voto de los usuarios de la comunidad) y para limitar la varianza en la capacidad de voto de los votantes, evitando el escenario de varios votantes de alto valor NR mezclados con muchas cuentas falsas de bajo valor NR. Los dos términos se pueden complementar entre sí, es decir, cuando la tasa de participación es alta, podemos ignorar los efectos de la divergencia.

5 Análisis de propiedades

Habiendo introducido el Protocolo de Incentivos para Desarrolladores, en esta sección analizaremos las manipulaciones que pueden ocurrir en la práctica, y las propiedades de DIP contra ellas. Desde el punto de vista de los votantes y los desarrolladores, respectivamente, las manipulaciones incluyen compra de votos, división maliciosa de DApps, ataques Sybil y otros.

5.1 Compra de votantes

Esta manipulación sucede cuando un desarrollador atrae todos los votos de un grupo de votantes hacia su DApp por medio de sobornos u otras prácticas similares, algo que ocurre con frecuencia en la vida real. Aquí suponemos que todos los votantes tienen su interés únicamente en sí mismos; asumimos que los votantes normales sólo sienten interés por la valuación de las DApps que son de su interés más que los incentivos que recibe tal o cual desarrollador. En otras palabras, un votante normal buscará maximizar la ponderación de la valuación de todas las DApps que son de su agrado. Nuestro algoritmo de valuación cuadrática garantiza la siguiente propiedad:

Propiedad 3. *En el modelo DIP, un votante normal con interés propio generalmente usará sus votos en múltiples DApps.*

Es posible ilustrar esta propiedad mediante el siguiente modelo: suponiendo que las ponderaciones con los que un votante a_i valúa todas las DApps son $b_{i1}, b_{i2}, \ldots, b_{in}$ respectivamente (puede ser considerado como la preferencia del votante por todas las DApps). Tomando la forma de8, el valor de contribución del usuario satisface

$$\frac{b_{i1}}{\sqrt{\Gamma_{i1}}} = \frac{b_{i2}}{\sqrt{\Gamma_{i2}}} = \cdots = \frac{b_{in}}{\sqrt{\Gamma_{in}}}.$$

Dicho de otro modo, los valores de contribución del votante a_i coinciden con sus verdaderas preferencias por esas DApps. La prueba detallada se encuentra en la sección A.1.

Los modelos tradicionales de voto generalmente computan el puntaje de valuación linealmente; esto es,

$$g(\Gamma_{1j}, \Gamma_{2j}, \ldots, \Gamma_{mj}) = \sum_{i=1}^{m} \Gamma_{ij}.$$

En este modelo, un votante racional sólo emitirá votos por su DApp favorita. En comparación, la fórmula 8 puede promover interacciones entre los votantes y las DApps, debido a la propiedad de la función raíz cuadrada. En otras palabras, los votantes que votan por múltiples DApps maximizan la utilización de su capacidad de voto. Un análisis similar se puede encontrar en [12]. Para sumarizar, un votante podría votar por distintas DApps y mantener al mismo tiempo la prioridad sobre sus DApps favoritas, tal como se observa en la ecuación más arriba.

En la práctica, a veces el modelo de voto lineal tradicional limitará la cantidad máxima de votos por votante para una DApp, para forzar la dispersión de sus votos, mientras nuestro algoritmo logra el mismo objetivo por medio de incentivos esenciales, con una expresión matemática más elegante y simple.

Corolario 1. *El valor de contribución total de un votante comprado es mucho menor que el valor total de contribución de un usuario normal.*

Un votante comprado a_i puede como mucho ofrecerle a su DApp un valor de contribución $\sqrt{\Gamma_i}$. Un votante normal que no ha sido comprado, asumiendo que planea votar por K DApps,[20] cuando la ponderación de sus valuaciones hacia esas DApps están uniformemente distribuidos, el incremento total del puntaje de valuación sobre todas las DApps causado por el votante está por encima de $O(\sqrt{K\Gamma_i})$, esto es, la eficiencia de un votante normal es K veces la eficiencia de un votante comprado. En consecuencia, el costo de la manipulación al comprar votantes se incrementa.

5.2 División maliciosa

En cuanto a los desarrolladores, otra posible manipulación consiste en dividir sus DApps con el fin de obtener mayores incentivos. Intuitivamente, la división de sus DApps les permite incrementar el número de participantes propios en el mecanismo

[20] K refleja el número de DApps de las cuales el valor de contribución del votante puede discriminar de otras DApps —que es usualmente mayor a 1— siempre y cuando la ponderación de la valuación del votante de las K DApps no esté extremadamente distribuido (es decir, el votante sólo vota por una DApp en particular y los votos para otras DApps tiende a 0).

de incentivos, e incrementar así el total de recompensas recibidas. No obstante, nuestro modelo garantiza que esto no suceda. Asumimos que todos los desarrolladores tienen interés en recibir incentivos así como una utilidad potencial causada por la mejora de su posicionamiento en el sistema de valuación.

Específicamente, tal como sucede con el algoritmo que calcula los incentivos finales, la convexidad de la fórmula 9 garantiza la siguiente propiedad:

Propiedad 4. *Si todos los votantes son normales, dividir una DApp en varias DApps no incrementará el total de recompensas para el desarrollador.*

Se asume que un votante normal pertenece a uno de estos casos: i). el votante distribuye los votos destinados a la DApp original en las DApps resultantes de la división. Tal caso ocurre cuando una aplicación tiene diferentes direcciones de contratos inteligentes para su invocación. ii). Suponiendo que el votante valúa la DApp original con una ponderación a, y valúa con ponderaciones b y c dos DApps divididas, respectivamente, entonces $c > a + b$. Tal caso se puede ilustrar con el hecho de que, luego de la división, las calidades de las DApps' se reducen fuertemente debido a la falta de vinculación entre ellas. Así, la suma de las calidades de las DApps dividida es menor que la calidad de la DApp original.

En ambos casos, el incentivo final para el desarrollador no se incrementa. Se brinda una prueba detallada de esto en la sección A.2.

Mas aun, un desarrollador podría simultáneamente comprar votantes y dividir su DApp: primero divide su DApp en K DApps y luego permite que sus votantes comprados distribuyan sus votos uniformemente entre todas las DApps resultantes de la división, maximizando así el uso de los votantes comprados. Preparamos el siguiente corolario contra este caso:

Corolario 2. *Incluso con la introducción de votos comprados, el desarrollador no puede incrementar sus incentivos por medio de la división de sus DApps.*

Se brinda una prueba detallada en la sección A.3.

Es de notar que la valuación de las DApps se decrementará si los desarrolladores las dividen, por lo que las potenciales utilidades también se verían disminuidas. En suma, nuestro algoritmo esencialmente previene estos casos de divisiones maliciosas.

Sin duda que para un desarrollador que escribe distintas DApps, como no hay relaciones de simetría o división entre sus DApps, la utilidad que recibe no se verá afectada.

5.3 Ataque *Sybil*

Al generalizar el ataque Sybil nos referimos a un ataque que subvierte el sistema de reputación al crear un gran número de identidades seudónimas, utilizando esas identidades espurias para ganar una influencia desproporcionadamente grande [13]. En el libro amarillo de Nebulas [1], las propiedades de Nebulas Rank contra tales manipulaciones está probada. Así, en el algoritmo de valuación DIP, los votantes tampoco serán capaces de incrementar sus valores NR mediante la creación de cuentas espurias, esto es,

$$\mathcal{C}(c) > \mathcal{C}(a) + \mathcal{C}(b)$$

donde c es la cuenta original, a, b son cuentas títere. De acuerdo a la fórmula 6 sus capacidades de voto satisfacen la siguiente restricción:

$$\sqrt{\Gamma_{a+b}} > \sqrt{\Gamma_a} + \sqrt{\Gamma_b} \tag{11}$$

Suponiendo que el propósito de un votante al lanzar un ataque Sybil es el de incrementar el puntaje de la valuación de una DApp específica y, simultáneamente, los incentivos de su desarrollador, de acuerdo a la restricción mencionada arriba, tenemos la siguiente propiedad:

Propiedad 5. *Para todo votante, la ejecución de un ataque Sybil no incrementará el puntaje de valuación de la DApp a la cual apunta a votar, ni tampoco el incentivo otorgado al desarrollador de la DApp mencionada.*

Así, la propiedad contra el ataque Sybil queda garantizada.

6 Implementación del Protocolo de Incentivos para Desarrolladores

La implementación completa del Protocolo de Incentivos para Desarrolladores está fuera del alcance de este libro púrpura; discutiremos únicamente los aspectos clave que deben ser manejados durante ese evento.

6.1 Cómo distribuir los incentivos

Para la distribución de los incentivos, se establecerá una cuenta especial, D. En el ínterin, una parte de las recompensas por nuevos bloques se transferirá a la cuenta D

de acuerdo a una proporción fija.

Es importante asegurarse de que cada desarrollador reciba sus incentivos de forma puntual.[21]. Con el fin de poder enviar las recompensas al blockchain, se requiere la clave privada de la cuenta asignada a tal fin, para poder firmar digitalmente la transacción necesaria. Por ello, para garantizar la seguridad, la cuenta D requiere un tratamiento especial.

En primer lugar, se implementará un tipo especial de transacción en el sistema, denotado como transacción `dip`, que contiene la información necesaria sobre monto total del incentivo para un desarrollador dado, y la *altura* del blockchain. En segundo lugar, el sistema rechazará todas las transacciones que no sean `dip` iniciadas por D, para asegurar que ninguna cuenta pueda extraer tokens de D. Finalmente, los nodos de verificación del blockchain verificarán las transacciones `dip`. Particularmente, esos nodos deberán correr DIP en forma local y verificar si la información de las transacciones `dip` coincide con sus resultados locales.

Por medio de los métodos descriptos en el párrafo anterior, no sólo se asegurará la distribución normal de los incentivos a los desarrolladores, sino que además se garantizará la seguridad de la cuenta D destinada a enviarlos.

6.2 Actualización de Protocolo de Incentivos para Desarrolladores

Como sabemos, Protocolo de Incentivos para Desarrolladores guarda una estrecha relación con el ecosistema. Al ser su factor de variación, DIP debe recibir actualizaciones, particularmente en sus parámetros; la cuestión de cómo lograr su actualización efectiva es un aspecto clave. Para lograrlo, se utilizará Nebulas Force para actualizar DIP de forma iterativa.

Se actualiza la estructura de los bloques para que puedan contener algoritmos y parámetros en DIP (bajo la forma de LLVM IR). La Máquina Virtual de Nebulas (NVM) recibe esos parámetros y corre esos algoritmos para determinar el total de tokens que una cuenta dada debe recibir.

Si es necesario actualizar esos parámetros y algoritmos, el Grupo Nebulas trabajará en conjunto con su comunidad para permitir que los nuevos bloques contengan parámetros y algoritmos actualizados que aseguren tanto la puntualidad como la fluidez del proceso, y que eviten la necesidad de realizar un *hard fork*.

[21] El intervalo de tiempo para el envío de incentivos equivale al intervalo de muestreo descrito en la sección 3.3

7 Trabajo futuro

7.1 Votación multidimensional

En la sección 3.2, vimos que el número de llamadas a las DApp es el criterio que se utiliza para determinar la valuación de las mismas, y las razones por las cuales la transferencia de tokens durante esas llamadas no son tenidas en cuenta para la valuación. En el futuro, no obstante, es posible que se introduzca la transferencia de tokens como un criterio adicional para la valuación de las DApps, cuando sea posible analizarlas en detalle.

7.2 Llamadas entre DApps

Actualmente, la valuación de una DApp está dada por la capacidad de *voto* de cada uno de los usuarios (esto es, el valor NR de cada una de sus cuentas) que la invocan. Sin embargo, situaciones más complejas —como las llamadas entre DApps— puede transmitir aún más la capacidad de voto de los usuarios.

En una versión futura, es posible que computemos la valuación final de cada DApp otorgándole una puntuación inicial para correr luego un algoritmo análogo al de Page Rank [14].

Referencias

[1] "Nebulas yellowpaper." `https://nebulas.io/docs/NebulasYellowpaperZh.pdf`.

[2] "Nebulas whitepaper." `https://nebulas.io/docs/NebulasTechnicalWhitepaperZh.pdf`.

[3] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 632–640, ACM, 2013.

[4] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 international conference on web search and data mining*, pp. 219–230, ACM, 2008.

[5] K. Yoo and U. Gretzel, "Comparison of deceptive and truthful travel reviews," *Information and communication technologies in tourism 2009*, pp. 37–47, 2009.

[6] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 309–319, Association for Computational Linguistics, 2011.

[7] Q. Cai, A. Filos-Ratsikas, C. Liu, and P. Tang, "Mechanism design for personalized recommender systems," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 159–166, ACM, 2016.

[8] A. Salihefendic, "How hacker news ranking algorithm works," 2010.

[9] A. Salihefendic, "How reddit ranking algorithms work," *Hacking and Gonzo*, vol. 23, 2010.

[10] E. Miller, "How not to sort by average rating," 2009.

[11] "IMDB." `https://www.imdb.com/chart/top`.

[12] V. Buterin, Z. Hitzig, and E. G. Weyl, "Liberal radicalism: Formal rules for a society neutral among communities," *arXiv preprint arXiv:1809.06421*, 2018.

[13] D. Quercia and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–5, IEEE, 2010.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

Anexo A Pruebas

A.1 Prueba de propiedad 3

Proof. Sin pérdida de la generalidad, podemos asumir que las *ponderaciones* con los que el votante a_i valúa todas las DApps son $b_{i1}, b_{i2}, \ldots, b_{in}$ respectivamente, y que son valores fijos. Asumimos que los valores de contribución del votante a_i a todas las DApps son $\Gamma_{i1}, \ldots, \Gamma_{in}$ respectivamente, y que son ajustables por el votante a_i.

El objetivo de la optimización del votante i es la suma ponderada de los puntajes de las valuaciones que ofrece, definida por:

$$w_i = \sum_{j=1}^{n} b_{ij}\sqrt{\Gamma_{ij}}$$

De acuerdo a la inecualidad de Cauchy, se desprende que:

$$w_i = \sum_{j=1}^{n} b_{ij}\sqrt{\Gamma_{ij}} \leq (\sum_{j=1}^{n} b_{ij}^2)(\sum_{j=1}^{n} \Gamma_{ij}) \leq (\sum_{j=1}^{n} b_{ij}^2)\Gamma_i$$

El último término del segundo miembro en la ecuación de arriba es un valor fijo. La igualdad se mantiene sí y sólo sí:

$$\frac{b_{i1}^2}{\Gamma_{i1}} = \frac{b_{i2}^2}{\Gamma_{i2}} = \cdots = \frac{b_{in}^2}{\Gamma_{in}}$$

Así, la propiedad queda demostrada. □

A.2 Prueba de propiedad 4

Proof. Sin pérdida de la generalidad, se asume que el desarrollador de la dapp d_1 la divide en dos DApps. Para todo votante normal que pertenece al segundo caso descripto en la sección 5.2 asumiendo que las *ponderaciones* asignados a valuar las DApps antes de su división son $b_{i1}, b_{i2}, \ldots, b_{in}$ y las ponderaciones con los que valúa las dos DApps divididas son b'_{i1}, b'_{i2}, se sostiene que $b_{i1} \geq b'_{i1} + b'_{i2}$ de acuerdo a nuestra presunción.

Luego, computamos los valores de contribución del votante a_i antes de ocurrir la división. Se define $H_i = \sum_{j=2}^{n} b_{ij}^2$, de acuerdo a la conclusión en la propiedad 3 y de

acuerdo al Teorema de Particiones tenemos:

$$\frac{\Gamma_{i1}}{b_{i1}^2} = \frac{\sum_{j=1}^{n} \Gamma_{ij}}{\sum_{j=1}^{n} b_{ij}^2} = \frac{\Gamma_i}{b_{i1}^2 + H_i}$$

Similarmente, el valor de contribuciones del votante a_i a la t-a división de la DApp (denotada por $\Gamma'_{it}, t = 1, 2$) es:

$$\Gamma'_{it} = \frac{b_{it}'^2 \Gamma_i}{b_{i1}'^2 + b_{i2}'^2 + H_i}$$

Nótese que $b_{i1}^2 \geq (b`_{i1} + b'_{i2})^2 > b_{i1}'^2 + b_{i2}'^2$, tenemos

$$\Gamma_{i1} > \Gamma'_{i1} + \Gamma'_{i2}$$

Así, introducimos la restricción de los valores de contribución para un votante lo suficientemente racional. Generalmente, la mayoría de los votantes pertenece al primer caso descripto en la sección 5.2, esto es, sencillamente distribuyen los valores de contribución que se suponen para d_1 para las DApps dividdad. En cualquier caso, tenemos:

$$\Gamma_{i1} \geq \Gamma'_{i1} + \Gamma'_{i2}$$

Defínase S'_1, S'_2 como los dos puntajes de valuación de las DApps, respectivamente. Por definición:

$$S'_1 = \sum_{i=1}^{m} \sqrt{\Gamma'_{i1}}, \quad S'_2 = \sum_{i=1}^{m} \sqrt{\Gamma'_{i2}}, \quad S_1 = \sum_{i=1}^{m} \sqrt{\Gamma_{i1}}$$

Defínase U'_1 como el incentivo final del desarrollador de la DApp d_1 luego de la división. Por definición:

$$U'_1 = \frac{S_1'^2 + S_2'^2}{S_1'^2 + S_2'^2 + \sum_{j=2}^{n} S_j^2} \lambda M, \quad U_1 = \frac{S_1^2}{S_1^2 + \sum_{j=2}^{n} S_j^2} \lambda M$$

Nótese que dados $S_2, \ldots, S_n$,

$$U_1 \geq U'_1 \Leftrightarrow S_1^2 \geq S_1'^2 + S_2'^2$$

Para mostrar si la división aumenta la utilidad, sólo es necesario comparar los sigu-

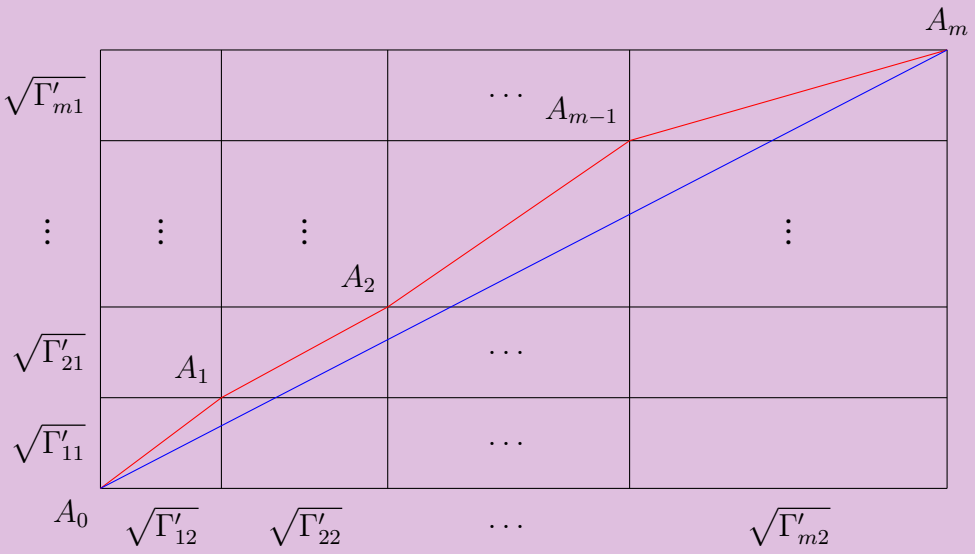

Figure 3: Prueba de la distancia más corta

ientes dos términos:

$$S_1^2 = (\sum_{i=1}^{m} \sqrt{\Gamma_{i1}})^2, \quad S_1'^2 + S_2'^2 = (\sum_{i=1}^{m} \sqrt{\Gamma'_{i1}})^2 + (\sum_{i=1}^{m} \sqrt{\Gamma'_{i2}})^2$$

En realidad, $S_1^2 \geq S_1'^2 + S_2'^2$ pueden ser demostrados de acuerdo al teorema de la distancia más corta. Como se muestra en la figura 3, construimos una grilla cuyos largo y ancho están divididos en m segmentos, cuyo i-ésimo segmento tiene una longitud de $\sqrt{\Gamma'_{i1}}$ y $\sqrt{\Gamma'_{i2}}$ respectivamente.

Entonces, $S_1'^2 + S_2'^2 = A_0 A_m^2$, esto es, es igual al cuadrado de la longitud del segmento azul. Entretanto,

$$S_1^2 = (\sum_{i=1}^{m} \sqrt{\Gamma_{i1}})^2 > (\sum_{i=1}^{m} \sqrt{\Gamma'_{i1} + \Gamma'_{i2}})^2 = (\sum_{i=1}^{m} A_{i-1}A_i)^2$$

, que es igual a la suma de los cuadrados de todos los segmentos rojos. Como la distancia más corta entre dos puntos es un segmento lineal, se mantiene que $S_1^2 > S_1'^2 + S_2'^2$.

Para los casos en los que las DApps $k > 2$ están divididas, podemos considerarlo como divisiones sucesivas y usar iterativamente el resultado en $k = 2$.

Así, la propiedad queda probada.

□

A.3 Prueba de corolario 2

Proof. Para todo votante comprado por el desarrollador de una DApp d_1, antes de su división, podemos considerar al votante como un votante normal que valúa todas las DApps con un vector de ponderación $(1, 0, 0, \ldots, 0)$, ya que otorga toda su capacidad de voto a d_1. Suponemos ahora que d_1 se divide en k DApps y que los valores de contribución de los votantes comprados a las DApps k son $\Gamma_{t1}, \ldots, \Gamma_{tk}$, cuya suma es fija. De acuerdo a la condición de que la igualdad se mantenga para la inecualidad de Cauchy en la Prueba de Propiedad 3, el votante puede ser considerado como un votante normal que valúa todas las DApps con un vector de ponderación $(\sqrt{\Gamma_{t1}}/C, \sqrt{\Gamma_{t2}}/C, \ldots, \sqrt{\Gamma_{tk}}/C, 0, 0, \ldots, 0)$, donde $C = \sum_{j=1}^{k} \sqrt{\Gamma_{tj}}$. Esto es, el votante valúa las DApps divididas con ponderaciones de acuerdo a proporción, y valúa todas las otras DApps con ponderaciones 0.[22]. Como

$$\sum_{j=1}^{k} \sqrt{\Gamma_{tj}}/C = 1$$

por lo que el caso de este corolario puede reducirse al caso de los votantes normales. (Propiedad4). Así, el corolario queda probado. □

A.4 Prueba de propiedad 5

Proof. Consideramos en primer lugar el caso de un votante que divide su cuenta en dos sub-cuentas. Consideramos c como la cuenta original y a, b como las sub-cuentas, S, S' son las puntuaciones de valoración de la DApp que el votante planea votar antes y después de la división respectivamente, U, U' son los incentivos finales para el desarrollador de la DApp que el votante planea votar, antes y después de la división respectivamente. Por definición, tenemos

$$S = \sqrt{\Gamma_c} + O, \quad S' = \sqrt{\Gamma_a} + \sqrt{\Gamma_b} + O$$

, donde O es la suma de los valores de contribución de otros votantes, que es un valor fijo.

Mediante 11 se sostiene que $S < S'$. Esto es, la valuación de la DApp no se incrementa.

[22]Nótese que al escalar todas las ponderaciones mediante una constante no se afectan los resultados, ya que el valor de contribución total del votante sólo depende de las proporciones de las ponderaciones con respecto a las ponderaciones totales

Entretanto, por definición,

$$U = \frac{S}{S+P}\lambda M, \quad U' = \frac{S'}{S'+P}\lambda M$$

, donde P es la suma de los cuadrados de los puntajes de la valuación de las otras DApp, cuyo valor es fijo.

Como $S < S'$ se sostiene que $U \leq U'$. Esto es, el incentivo final que recibe el desarrollador no se incrementa.

Para los casos en los que $k > 2$ cuentas están divididas, podemos considerarlo como divisiones sucesivas y usar iterativamente el resultado en $k = 2$.

□

Anexo B Registro de cambios

B.1 2019.4.8 registro de cambios

- Se reemplaza la función de NR a capacities (4) mediante $f(\mathcal{C}(a_i)) = \mathcal{C}(a_i)$
- Se reemplaza el cálculo del factor de participación λ (10) mediante $\min\{\frac{0.008}{1-r}, 1\}$, donde $r = \frac{\Gamma_p}{\Gamma_s}$
- Se corrigen errores de tipeo.