Dora Network

郭雄輝 Tyler Kot 李星
steve@dora.network tyler@dora.network star@dora.network

Version 1.1.2
2018 年 12 月 9 日

摘要

以太坊等新一代智能合約平台的出現極大地推動了區塊鏈技術的應用，但目前存在性能不足的問題，無法滿足更廣泛的應用需求。Dora 從三個方面解決性能問題：縱向擴容、橫向擴容和共識算法。縱向擴容挖掘合約間和合約內的並行度；橫向擴容採用子母鏈的技術提高性能；共識算法 DVBC 基於 DPoS，VRF 和 BFT，兼顧安全性和高性能。Dora 虛擬機兼容 EVM，基於 EVM 的 dApp 能快速移植，並通過零交易費的經濟模型進一步激勵生態建設。總的來說，Dora[1]是一個免交易費的高性能高並發公有鏈。

1 區塊鏈的現狀與展望

1.1 區塊鏈技術綜述

2008 年 9 月，中本聰發表比特幣白皮書[1]。2009 年 1 月，比特幣主鏈正式上線，並安全運行至今，開創了加密數字貨幣的新時代，並將區塊鏈這一新技術拉入公眾視野。2015 年 7 月，以太坊[2] 上線運行。以太坊的 EVM 可執行圖靈完備的智能合約，標誌著第二代區塊鏈技術走上舞台。在眾多的以太坊應用當中，加密數字通證無疑是最為重要和流行的一種。以太坊社區推出了 ERC-20 標準，並提議了 ERC-721 標準，為加密數字貨幣（cryptocurrency）、通證 (token) 和非標通證（non-fungible token）的發行和流通奠定了基礎。2017 年，整個加密數字貨幣市場增長超過五十倍，上千種新的加密數字貨幣和通證的發行和升值是主導因素。目前，區塊鏈的前沿創新技術主要體現在共識機制，區塊結構和網絡結構。

首先是共識機制。最早在比特幣上採用的是工作量證明 PoW 共識算法，大量的礦工進行哈希運算來爭奪區塊的記賬權，導致大量的電力消耗。為了克服 PoW 資源浪費的缺點，新的權益證明 PoS 共識算法根據用戶持有代幣數量，以及用戶持有代幣的時間來決定區塊的記賬權，大大減少了爭奪記賬權而消耗的電力，同時提升了效率。PoW 和 PoS 都還需要礦工爭奪記賬權，代理權益證明 DPoS 共識算法參考人類行為活動中的公司運營機制，採用去中心化的投票方式先選舉出有限的代理記賬節點，這些被選舉出的代理節點再按照規則輪流打包區塊，避免了對記賬權的爭奪，效率得到更近一步的提升。這三種共識算法理論上只要網絡中不超過 50% 的記賬節點出問題，整個網絡就是安全的。它們都屬於間接達成共識的算法族，先要爭奪出記賬權，再生成區塊，最後通過確定性方法（比如最大難度）解決分叉問題。而在聯盟鏈中更多采用拜占庭容錯 BFT 算法，它屬於直接達成共識的算法，這種算法一輪運行結束後就能確定性地在參與者

[1]路印基金會投資，香港 Loopnest 加速器孵化項目。

之間形成區塊共識，不需要先爭奪記賬權，也不會有分叉，但它只能保證在網絡中不超過 1/3 的記賬節點出問題的情況下，整個網絡是安全的。

其次是區塊的組織結構。傳統的區塊鏈採用的是樹狀結構，一個區塊有且僅有一個父區塊，通過區塊之間的父子關係來形成全局有序的線性賬本。而最新的研究則允許區塊有多個父區塊，比如 IOTA[3] 強制要求新區塊必須指向兩個父區塊，從而組織成一個有向無環圖 DAG，將多個交易全局有序的線性賬本規約為一個只記錄部分偏序關係的非線性賬本，從而加速交易的確認速度。

最後是網絡結構。將大的網絡分片組織成小的網絡，採用子母鏈的結構。比如 Ardor[4]，Cosmos[5]，Asch[6] 和 PChain[7]。Cosmos 將主鏈稱為 Hub，其他子鏈稱為 Zone，Hub 和 Zone 之間通過 IBC (Inter Blockchain Communication) 協議交互，當一個幣種通過 Hub 從一個 Zone 轉到另外一個 Zone 時，Hub 會負責保持其總量的不變性，但是 Hub 不負責驗證單個 Zone 上的交易，Hub 和 Zone 都是採用 Tendermint[8] 共識算法。PChain，類似於 Cosmos，嘗試按每個 dApp 構成子鏈，PChain 主鍊和子鏈的共識算法採用 PoS。這些方案中母鏈對子鏈的交易沒有任何記錄和檢查機制，子鏈的安全性完全交由子鏈的礦工去維護，在子鏈礦工數比較少的情況下存在潛在的安全問題。Asch 和 Ardor 考慮到這一點，設計出一個新的礦工角色，要求其把子鏈的區塊記錄打包上傳到母鏈上，Asch 設計要求子鏈的創建者上傳子鏈區塊，而 Ardor 的設計則不做角色的強制要求。Ardor 考慮的更周全一些，讓母鏈在一段時間後可以對子鏈的區塊數據做快照和裁剪，從而解決母鏈數據膨脹問題。

1.2 目前區塊鏈待解決的兩個問題: 性能和存儲

隨著需求的增加，區塊鏈面臨亟需解決的可擴展性問題。目前區塊鏈的低吞吐能力 (比特幣大概 7 筆交易每秒，以太坊大概 15 筆交易每秒)，不足以滿足全球金融交易的需求。相比之下，Visa 可處理 56000TPS 的交易，支付寶在 2017 年 11 月已實現了 200000+TPS 交易峰值。目前區塊鍊網絡的可擴展性問題嚴重限制其廣泛的應用，例如以太坊所支持的智能合約直接作為 dApp 運行在以太坊上，隨著鏈上項目越來越多，負載越來越重，像“加密貓”一個項目就能把整個以太坊弄的連基本的轉賬都很難成功。如何在不影響安全性和去中心化特質的前提下提升區塊鏈吞吐量，仍然有待探索。

隨著時間的流逝,區塊鏈還面臨數據膨脹問題。如圖1所示(數據來源,http://bc.daniel.net.nz),截止到 2018.2 月份比特幣目前的區塊總大小超過 150G，新節點需要花費 14 天才能同步完所有區塊；以太坊區塊總大小超過 650G，新節點需要花費 8 天才能同步完所有區塊，而且還在以每天約 145M 的速度增長。

1.3 Dora 的展望

Dora 採用全新的開創性區塊鏈架構設計，旨在用區塊鏈技術滿足全球範圍商業活動的需要。Dora 從 CPU 設計中受到啟發，設法將流水線模型和分支預測算法應用於區塊鏈，提出了針對區塊鏈可擴展性問題的解決方案，大幅提高區塊鏈的性能，並兼顧安全性和去中心化。

Dora 將推動區塊鏈進入下一代，建設高速高效，通過節點快照新節點可快速加入的網絡。最終，Dora 的目標是變成一條人人可用，人人易用的公有鏈。

2 Dora 縱向擴容：並行化

傳統的區塊鏈擴容解決方案分為兩種：狀態通道和多鏈分片。像閃電網絡[9] 就是利用狀態通道技術來緩解區塊鏈可擴展性問題。其基本思想是固定的一組當事人之間的頻繁交易，在所有各方都完成交易後，其中一方只發布最終結果，而無需在鏈上生成多個交易記錄 (本質上是減少

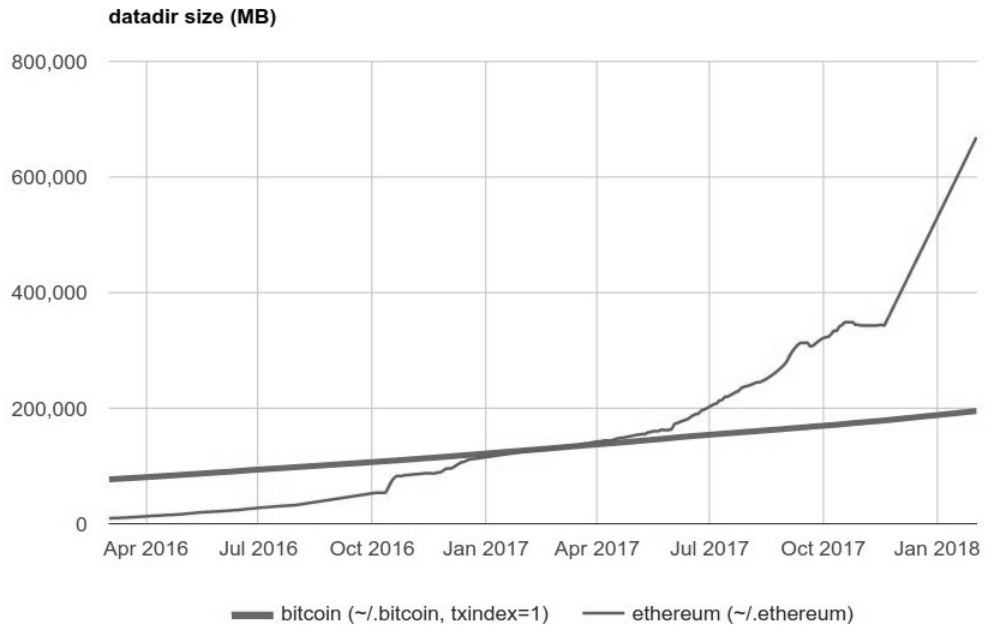

圖 1: 區塊鏈數據膨脹

中間結果的存儲)。然而，閃電網絡只適用於固定的一組當事人之間頻繁的交易，如果用戶的交易目標是隨機的並且交易行為偶爾發生的話，就會導致低效率。而多鏈分片則是一種橫向擴容技術，通過增加鍊或者片區的數量，把交易分散達到最終擴容的目的，但這種方式通常伴隨著子鍊或者子分片上的安全性問題。

這些解決方案都以交易的串行化執行為假設條件，交易的串行化使得每個礦工節點可以獨立去執行和驗證，從同一個創世塊出發經過同樣的串行操作序列，一定能得到一樣的輸出結果。很顯然，交易的串行化執行會導致系統的 TPS 總是受限於單個節點的性能。需要思考的是，交易和交易之間一定要串行執行嗎？是不是能在交易級別做並行處理從而設計一種縱向擴容技術呢？

2.1 普通轉賬交易的並行

將賬號看成一個節點，A 轉賬給 B，則在 A 節點到 B 節點之間添加一條有向邊，邊上的數字表示轉賬金額。一個區塊內的所有交易可以構成一個圖。舉一個例子，假設一個區塊中包含如下一些交易 {A 轉帳給 B 10 個 token，A 轉帳給 C 5 個 token，D 轉賬給 F 3 個 token，E 轉賬給 G 2 個 token}，如圖2所示。

從圖上明顯可以看到這些交易被劃分為三個子連通圖：{A,B,C}, {D,F} 和 {E,G}，同一個連通圖內的交易存在依賴關係，只能串行執行，但不同連通圖之間是可以並行執行的，比如 D 轉賬給 F，E 轉帳給 G 的先後執行順序只是在中間臨時狀態不同，但不會影響區塊最終的狀態。在現實區塊鏈世界中，很容易找到這些沒有依賴關係的交易集合。

2.2 智能合約間的並行

針對普通轉賬交易，可以按涉及的賬號來劃分子連通圖，但智能合約轉賬交易的劃分複雜的多。先看圖3智能合約的鏈上執行示例。一次智能合約的執行邏輯上可以分成兩部分：一部分用白色標識 EVM 解釋執行部分，另外一部分用灰色標識真正改變世界狀態的部分。灰色部分是影

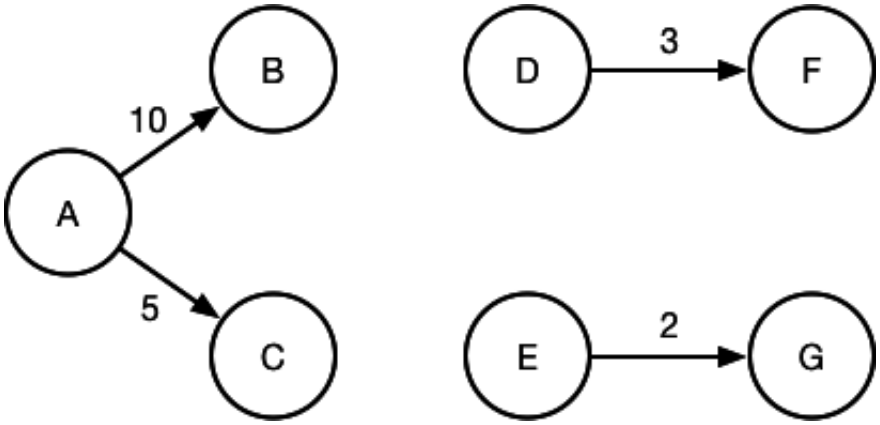

圖 2: 普通轉賬交易

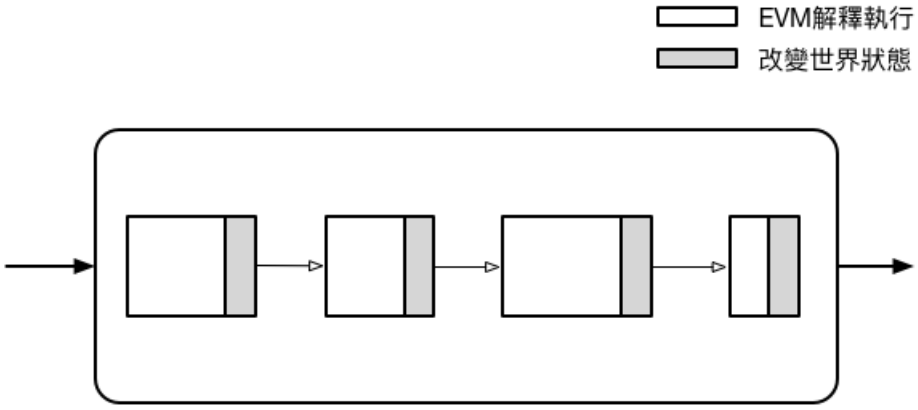

圖 3: 智能合約的串行

響最終世界狀態的關鍵組成部分。智能合約的執行過程中可以採集到該智能合約所依賴的外部賬號的世界狀態，也能採集到最終影響的外部賬號的世界狀態。

只要一個智能合約所依賴和改變的賬號集合與另外一個智能合約所依賴和改變的賬號集合有交集，這兩個智能合約是連通的。這樣將一個區塊中所有的智能合約交易劃分成不同的連通子圖，各個子圖之間可以並行執行。如圖4所示，假設這四個智能合約交易可以劃分在不同的連通子圖中，並行化處理後，該區塊的生成速度能極大提升，從而提升整個系統的 TPS。這種並行化算法既可以用單機多線程來實現，也可以用多機並行處理。

通過對以太坊區塊高度 5592867 到 5609843 之間的區塊數據分析，得出以下結論：智能合約交易占總交易量的 53.7%，平均每次智能合約交易執行時間為 1.29ms，最大執行時間為 51.1ms，最短執行時間為 0.14ms。也就是說，如果智能合約調用是串行化的話，理論上系統最大的 TPS 只能達到 775，如果並行執行，則不存在理論上的限制。

這種縱向擴容技術本質上類似於 CPU 設計中的分支預測，即通過流水線操作提前執行分支，如果預測正確，後續流水線只需按序執行，從而極大地提高系統的並行性。Dora 設計的縱向擴容技術，理論上能運用到任何目前已知的公有鏈上，提高系統的 TPS。

下面給出算法的形式化定義。

2.2.1 符號定義

定义. 首先定義後面用到的一些概念:

智能合約空間: $\mathbb{T}$

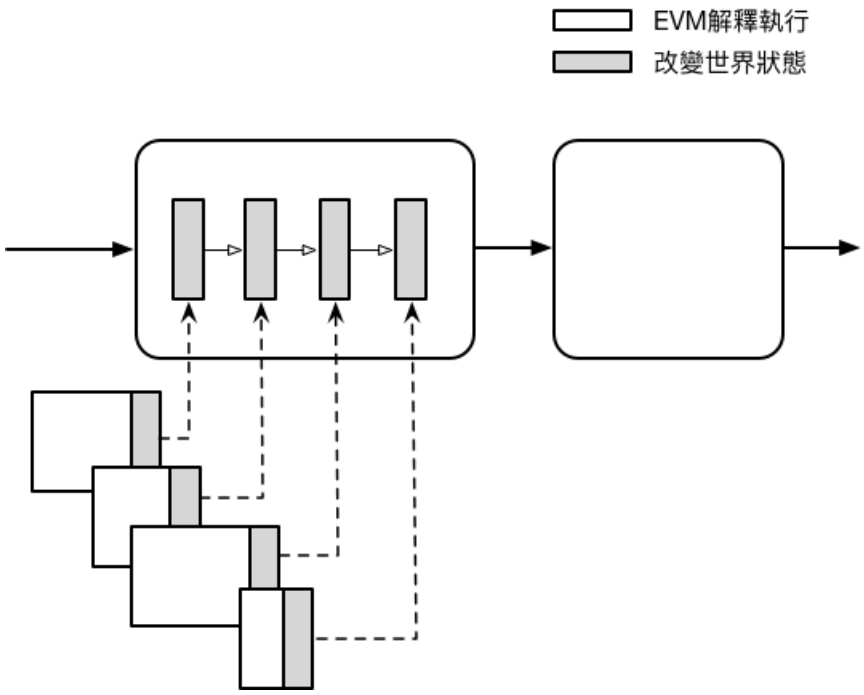

圖 4: 智能合約的並行

區塊: $B_t = \{T_{1..}T_{n_t}\} \in 2^{\mathbb{T}}$, 它是一組智能合約的集合。

賬戶地址空間: $\mathbb{A}$

賬戶狀態空間: $\mathbb{L}$

世界狀態空間: $\mathbb{L}^{\mathbb{A}}$, 它是從賬戶地址空間到狀態空間的映射的集合。

世界狀態轉換函數: $\Gamma : \mathbb{L}^{\mathbb{A}} \times 2^{\mathbb{T}} \to \mathbb{L}^{\mathbb{A}}$, 例如 $\sigma_t = \Gamma(\sigma_{t-1}, B_{t-1})$，其中 $\sigma_t, \sigma_{t-1} \in \mathbb{L}^{\mathbb{A}}$。它根據當前世界狀態, 執行一組智能合約並更新世界狀態。

串行執行: 如果對於一組合約中任意一對合約, 其中一個合約的所有操作都在另一個合約的所有操作之前, 那麼稱為這組合約是串行執行的。也就是原子化串行。

安全性: 如果並行執行的一組合約對世界狀態所產生的改變等同於這組合約的某些串行執行, 則稱該並行執行是安全的。即給定並行狀態轉換函數 Γ_p 和所有串行轉換函數的集合 $\mathbb{S}$, 需滿足 $(\exists S)(S \in \mathbb{S}) \wedge (S(\boldsymbol{\sigma}, B_{t-1}) = \Gamma_p(\boldsymbol{\sigma}, B_{t-1}))$ 。

智能合約的讀賬戶集合:$T_r \in 2^{\mathbb{A}}$

智能合約的更新賬戶集合:$T_w \in 2^{\mathbb{A}}$

賬戶的狀態:$L(T_r) \in 2^{\mathbb{L}}$

2.2.2 算法流程

下面給出 SSP 算法1的流程，它可以在保證安全性的前提下提升系統性能，減少執行時間。需要說明的是普通轉賬可以看做是一種簡化的智能合約，所以 SSP 也是適用的。

2.3 智能合約內的指令級並行

Dora 完全兼容以太坊 EVM 指令集, 並在實現層面進行性能優化：編譯時優化, 包括循環展開, 指令合併等以及運行時優化，包括 JIT 等。同時允許程序員對代碼進行標註（Annotation），告訴編譯器哪些部分可以被並行優化。

Algorithm 1: Safe Speculative Parallelization(SSP) 安全推測並行

Procedure *proposer*.`Propose()`
  repeat
    Receive $T \in \mathbb{T}$;
    Execute smart contract $\sigma = \Gamma((T_r, L(T_r)), T)$;
    Send the tuple (σ, T_r, T_w) to validator;
  until *exit*;
  return;
Procedure *validator*.`Validate()`
  Set of valid contracts $\mathbb{V} \leftarrow \varnothing$;
  Delta world state $\Delta\sigma \leftarrow$ null;
  repeat
    Receive the tuple (σ, T_r, T_w) from proposer;
    for *each valid contract* $V \in \mathbb{V}$ **do**
      if $V_r \cap T_w \neq \varnothing$ *or* $V_w \cap T_w \neq \varnothing$ *or* $V_w \cap T_r \neq \varnothing$ **then**
        Abort T;
      end
    end
    if T *is not aborted* **then**
      $\mathbb{V} \leftarrow \mathbb{V} \cup T$;
      $\Delta\sigma \leftarrow \Delta\sigma + \sigma$;
    end
  until *timeout or the* $|\mathbb{V}| > n$;
  Update world state $\sigma_{t+1} \leftarrow \sigma_t + \Delta\sigma$;
  return;

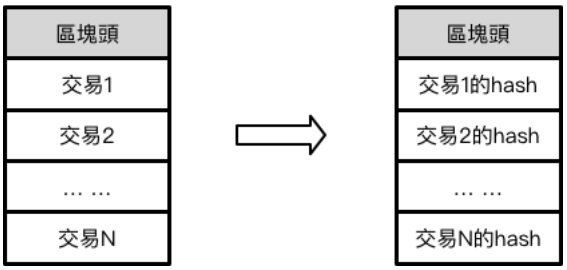

圖 5: 區塊瘦身

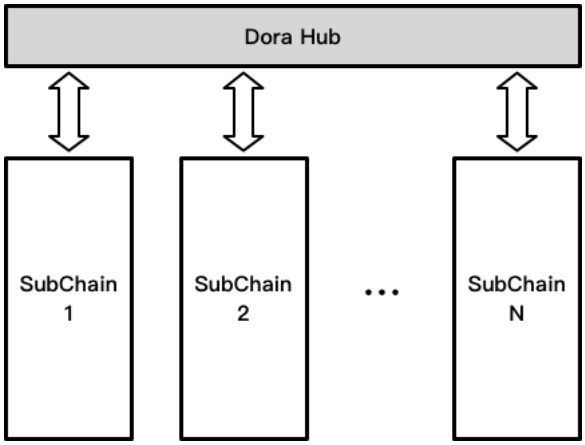

圖 6: 子母鏈

2.4 動態調節區塊大小和區塊瘦身

為了提高整個網絡的 TPS，Dora 會根據當前網絡擁堵情況動態調節區塊大小，同時引入一種區塊瘦身技術，如圖5所示：

傳統區塊鏈中，一個區塊通常會包括區塊頭和一系列的交易列表。舉以太坊為例，一筆交易大約要佔用 200 多個字節，一個節點廣播區塊的時候會把原始交易都打包到區塊中廣播出去，這對網絡帶寬資源造成不必要的浪費，因為每筆交易在區塊打包之前就已廣播到網絡節點上了。理論上，廣播區塊時，只在區塊中放入交易的 Hash 值（只需 32 個字節）即可。當節點收到區塊中交易 Hash 列表時，如在本地查找不到對應的交易數據，再向其他節點詢問獲取。採用這種壓縮模式，網絡帶寬理論上能節約 7 倍左右，對應到 TPS 也有同樣提升。

3 Dora 橫向擴容：子母鏈

Dora 以子母鏈的方式對區塊鏈做橫向擴容（如圖6）。一個智能合約可以部署在母鍊或者子鏈上。對計算性能要求高的智能合約可以單獨部署在子鏈上。

子母鏈均採用賬號模型，賬號可以在子鍊和母鏈上共用的，但其在不同鏈上的狀態會分別記錄，同一個賬號在不同鏈上各自記錄在該鏈上的世界狀態。舉個例子，用戶可以通過錢包創建一個賬號，該賬號可以同時用於接收和發送母鍊和子鏈的轉賬交易，但母鏈上擁有的代幣數記錄在母鏈上，子鏈上的代幣記錄在子鏈上，同時通過一種特殊的跨鏈交易 CrossChainTransaction 來支持子母鏈之間的代幣轉賬交易。母鏈支持多幣種轉賬。

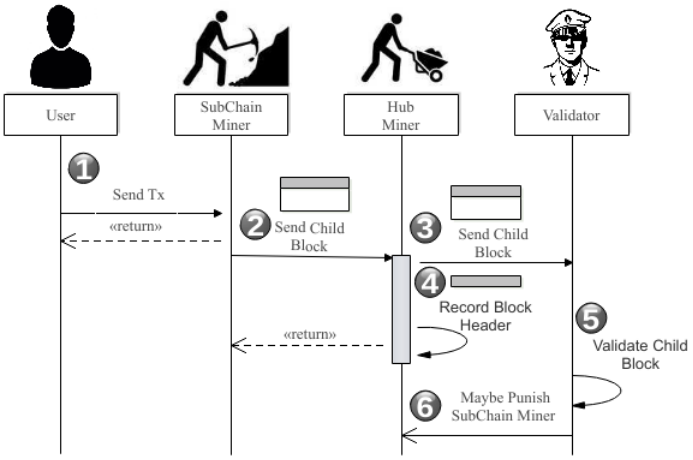

圖 7: 子鏈安全性

3.1 子鏈安全性

如圖7所示，Dora 主鏈支持一種特殊的交易，每隔一段時間子鏈中生成區塊的礦工負責把還未記錄在主鏈上的子區塊信息 ChildChainBlock 打包遞交到主鏈上，但注意母鏈上不會記錄完整的子區塊信息，而只是記錄子區塊頭信息，這些數據僅僅只是記錄，母鏈的礦工打包時不負責驗證。為了保證子鏈數據的安全性，Dora 要求子鏈的礦工抵押一定數量的 DNT 代幣，同時引入了監察者礦工。監察者礦工在發現並核實子鏈礦工的錯誤區塊後，獲取一定的獎勵。

監察者同時會監聽母鍊和子鏈的所有交易，當檢測到母鏈上有子鏈區塊打包消息 ChildChainBlock 時，結合之前收到的子鏈交易數據去做驗證，一旦檢測到有子鏈區塊存在問題，則把證據遞交到母鏈上，同時為了防止監察者發起 DoS 攻擊，監察者遞交證據時需要抵押一定數量的 DNT 代幣，一旦證據屬實，則監察者可獲得出錯子鏈礦工抵押的代幣，如果證據不屬實，監察者本次抵押的代幣會作為獎勵給母鏈礦工。

3.2 數據剪枝和快照

為了解決數據膨脹問題，Dora 母鏈支持子鏈區塊信息的剪枝裁剪，記錄在母鏈上的子鏈的數據只保留近期區塊的數據（比如 2048 個區塊），更久之前的數據可以從母鏈上移除。

同時母鏈還支持快照，一個快照包含當前區塊所有賬號的世界狀態信息，新的節點可以只從某一個快照塊出發從而快速同步，甚至輕量級節點可以刪除快照以前的所有區塊數據。

3.3 Dora 子母鏈之間代幣跨鏈轉賬和交易

Dora 設計了一個特殊的 CrossChainTransaction 交易能讓同一賬號內的代幣在子鍊和母鏈上互相流通。

CrossChainTransaction: {

blockno：區塊編號，*rawtransaction* 這筆交易在具體哪個區塊內

blockhash：*blockno* 對應區塊的 *Hash*

merklepath: 驗證 *rawtransaction* 是否在區塊 *blockno* 內的默克爾樹路徑

rawtransaction:{

owner: ""0xfbc2a4...ed"",

symbol: “*UT*”,

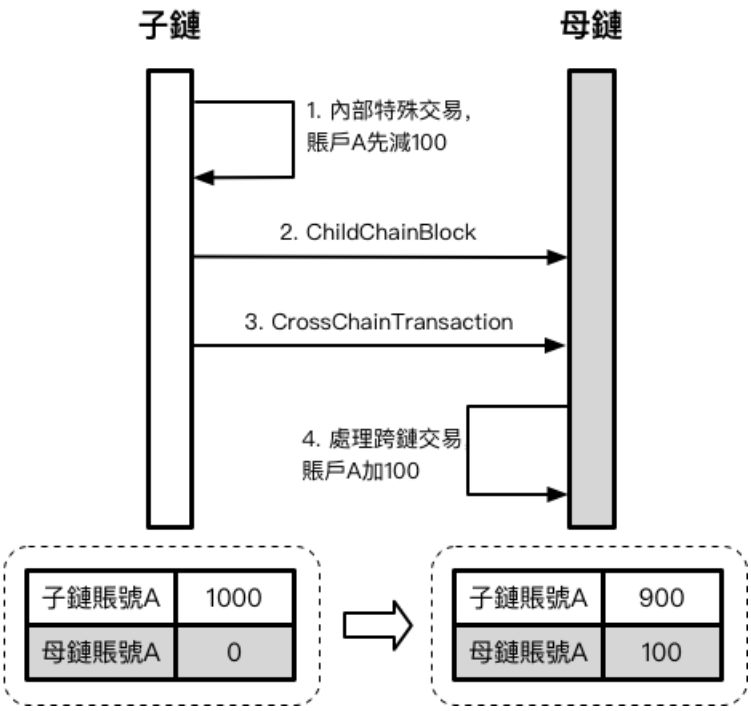

圖 8: 代幣在子母鏈間轉移

value: 100,

direction: 0 or 1（0 表示從子鏈到母鏈，1 表示從母鏈到子鏈）

}

}

如圖8所示，展示了代幣 UT 如何從子鏈轉移到母鏈。

假定賬號 A 轉移 100 個 UT 到母鏈上，當跨鏈交易發生時：

1.A 先在 UT 子鏈上發起一筆特殊的轉賬交易，表示將 100 個 UT 從子鏈賬號 A 轉移到母鏈賬號 A；

2. 該特殊交易在子鏈上得到確認後，等待子鏈的礦工將包含該交易的區塊信息記錄到母鏈上；

3.A 接著在母鏈上發起一筆從子鏈到母鏈的 CrossChainTransaction 交易，附帶上步驟 1 的轉賬交易，區塊高度和 merkle 路徑供母鏈做驗證；

4. 母鏈礦工驗證 CrossChainTransaction，如果驗證通過則打包進母鏈區塊，並修改母鏈賬號 A 的狀態記錄轉入 100 個 UT；

反之，代幣可以遵循同樣的規則就可以從母鏈上再轉移回子鏈，這種方案的好處是整個轉移過程中代幣還保留在同一個賬戶下，不像其他跨鏈解決方案引入額外的受控制賬戶。

同時 Dora 母鏈上設計了一個特殊的 TokenSwapAction，允許兩種代幣之間的原子互換，需要攜帶雙方簽名的訂單信息，並同時會檢查雙方的餘額來決定是否能執行互換。

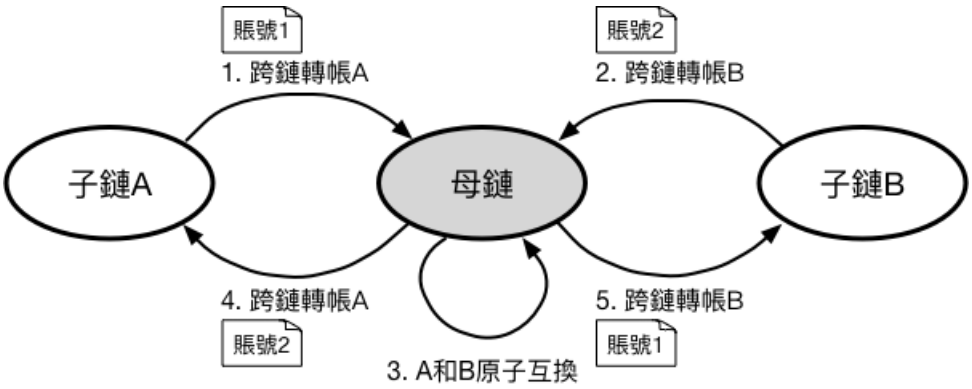

圖 9: 跨鏈交易

TokenSwapAction: {
taker: "LRC",
takerAddr: "0xfbc2a4...ed"
takerValue: 1000,
maker: "RDN",
makerAddr: "0xabc35e...ef"
makerValue: 500,
takerOrderWithSignature:
makerOrderWithSignature:
}

通過這個原子互換代幣操作 TokenSwapAction 和跨鏈操作 CrossChainTransaction，很容易實現子母鏈之間的跨鏈交易 (圖9)。

4 共識算法 DVBC

4.1 算法思想

為了平衡高性能和高安全的兩個需求，Dora 提出了一種兼顧安全和效率的混合共識算法 DVBC（Delegate Verifiable BFT Consensus)。DVBC 基於 DPoS[10]，VRF[11] 和 BFT[8]。算法分為 3 個層次。假設有 M 個節點，一共選出 $N+1$ 個節點，N 初始取為 21。在每一輪共識開始時，Dora 首先使用第一層算法 DPoS 根據選票的大小從 M 個節點中選擇出得票數排名前 $N/3$ 個節點，這些節點會自動成為本輪的記賬者。第二層算法將剩下的節點 $MN/3$ 作為候選節點，在這些節點中採用可驗證隨機數 VRF 算法隨機挑選出其中的 2*N/3+1 個候選節點作為本輪的記賬者。最後一層運用 BFT 算法在 $N+1$ 個記賬者之間來達成區塊共識。該共識算法能保證在 $N+1$ 個節點中只要不超過 $N/3$ 個作惡節點，區塊的正確性仍然得以保證，而且不存在任何分叉。顯然該算法可以防止超級節點聯合作惡。此外，由於其他候選節點也能被選中作為記賬者，能讓更多的節點願意作為候選節點，從而增加網絡的安全和魯棒性。

考慮到每個 dApp 對安全和性能的要求不同，Dora 允許在創建子鏈的時候指定子鏈的共識算法：DPoS+VRF，BFT 或者兩者混合。

4.2 形式化描述

算法2描述了 DVBC 的整體流程。其中 DPoS，VRF，BFT 分別如算法3，4和5所述。

DPoS 給出了一種根據投票比例作為候選節點被選中概率的抽樣算法。

VRF 算法首先用每個候選節點的公鑰和前一區塊的 Hash 值作為源信息，將該信息交給 VRF Hash 函數算出新的 Hash 值，利用該值進行排序選出 2*N /3+1 個節點。使用 VRF 的好處是可以快速形成共識，找出候選節點，並且可以被驗證。

BFT 算法採用 π 演算[12] [13] [14] 來描述。π 演算的基本語法如圖10所示。使用 π 演算的好處是便於分析算法的性質，例如安全性和活性，還可以使用軟件對算法做形式化驗證。該算法分為 propose，prevote，precommit 和 commit4 個的狀態。P, V, C 是 propose，prevote，precommit 的縮寫, 分別對應算法中的 3 個主函數。為了方便描述遞歸關係，還定義了子函數 PrevoteSub 和 PrecommitSub，簡寫為 VS 和 CS。getProposal 函數當提議 p 不為空時需直接返回 p，否則整合最近交易信息到一個提議中並返回。函數 $round\_robin(k) = k \mod N$ 表示根據當前輪次選出一個負責 propose 的進程，類似 leader 的角色。N 個節點之間的共識協議可以表示為 $Consensus ::= \prod_{i=1}^{N} Y_i$。每個節點的狀態記做 $s = \{k, p, v\}$，k 表示輪次，p 表示本輪提議的區塊信息，v 表示所有輪的投票信息。$v_k^{'}$ and $v_k^{''}$ 分別表示在第 k 輪的 prevote 和 precommit 投票信息。進程之間通過通信信道傳遞信息, 這裡用 cp_i，cv_i，cc_i 和 cw_i 分別表示第 i 個進程用於 propose，prevote，precommit，commit 的通信信道。

函數 Propose 首先判讀自己是否當前的提議者，如果是，則打包最近的交易到提議 pr 中，將 pr 通過 *cp* 通道發送到給其他進程；同時進入 V 狀態。如果不是 proposer，而且已經有提議, 也進入 V 狀態；否則接受提議者從 *cp* 通道發來的提議或者超時後再進入 V 狀態。函數 Prevote 會把收到的提議通過 *cv* 通道廣播給其他進程，並從其他進程的 *cv* 通道接受提議；然後創建共享通道 c，將收到的提議通過 c 發出；同時進入 PrevoteSub 子函數。在 PrevoteSub 函數中，如果從超過 2/3 的進程收到相同的提議，則進入 C 狀態；否則若收到超過 2/3 的進程的提議，但不一致，則攜帶空提議進入 C 狀態；否則還沒有收到 2/3 的進程提議, 需分情況處理。讀取通道 c 的信息，若發現是舊的消息，則繼續等待；若是當前輪信息，則合併投票信息並等待；若自己的輪次落後，回到 P 狀態，追上當前輪。

函數 Precommit 從 *cc* 通道廣播提議給其他進程, 並從其他進程的 *cc* 通道接受提議；同時創建通道 c，將收到的信息發出去。類似 Prevote 函數，如果從超過 2/3 的進程收到相同的提議則正式提交該提議；否則如果提議不一致，轉到 P 狀態，開始下一輪；否則繼續等待，如果發現自己輪次落後則回到 P 狀態。

Algorithm 2: DVBC

Function `DVBC()`

```
while true do
    list of delegates L = ∅;
    L = DPoS(N/3, candidate_votes);
    L.append(VRF(2 * N/3 + 1, candidates));
    BFT(L);
end
```

4.3 BFT 算法並行化

假設每個區塊的 propose, prevote, precommit, commit 四個狀態的時間分別為 $T1$，$T2$, $T3$, $T4$，按串行執行生成和確認 N 個區塊所需要的時間為 $N*(T1+T2+T3+T4)$ 。

從流水線的角度來考慮這個問題，四個狀態可以有序並行執行，理想情況下，如圖11所示，按照流水線的方式執行生成和確認 N 個區塊所需要的時間為 $N*T1+T2+T3+T4$。當 N 足

$P ::=$	0	*empty summation*
$\mid$	$P_1 \mid P_2$	*composition(parallelization)*
$\mid$	$P_1 + P_2$	*summation(nondeterministic choice)*
$\mid$	$(new\ x)P$	*channel creation*
$\mid$	$\alpha.P$	*prefix form*
$\mid$	$\tau.P$	*silent action*
$\mid$	$x(y).P$	*input*
$\mid$	$\overline{x} < y > .P$	*output*
$\mid$	$(x).P$	*restriction*
$\mid$	$timeout_i.P$	*timeout of process i*
$\mid$	$P^s(y)$	*function to which pass variable s and y*

$\sum_{i=1}^{n} P_i ::=$	$P_1 + P_2 + .. + P_n$	*multiple summation*
$\prod_{i=1}^{n} P_i ::=$	$P_1 \mid P_2 \mid .. \mid P_n$	*multiple composition*

圖 10: π 演算符號說明

Algorithm 3: DPoS

Function `DPoS`(*m, candidate_votes*)

list of delegates $L = \varnothing$;
$status = candidate\_votes$;
$sum = summation(candidate\_votes)$;
for $i = 0;\ i < m;\ ++i$ **do**
  $status \mathrel{+}= candidate\_votes$;
  $top = sort(status).top\_index())$;
  $L.append(top)$;
  $status(top) \mathrel{-}= sum$;
end
return L;

Algorithm 4: VRF

Function `VRF`(*m, candidates*)

list of delegates $L = \varnothing$;
$info = candidates.pub\_key + previous\_block\_hash$;
$candidates\_hash = VRF\_hash(sec\_key, info)$;
$topn = sort(candidates\_hash).topn(m))$;
$L.append(topn)$;
return L;

Algorithm 5: BFT

Procedure `Propose()`

$P_i^{k,p,v}$ ::= if $i == round\_robin(k)$ then

$\overline{cp_i} < pr > |V_i^{k,pr,v}$, where $pr = getProposal(p)$

else if $p \neq \varnothing$ then

$V_i^{k,p,v}$

else

$cp_{round\_robin(k)}(pr).V_i^{k,pr,v} + timeout_{round\_robin(k)}.V_i^{k,\emptyset,v}$

Procedure `Prevote()`

$V_i^{k,p,v}$ ::= $\overline{cv_i} < p > |(new\ c)(\prod_{j=1}^{N} cv_j(pr).\bar{c} < cv_j, pr > |VS_i^{k,p,v}(c))$

Procedure `PrevoteSub()`

$VS_i^{k,p,v}(c)$ ::= if $max_b(|\{pr \in v_k^{'} : pr.block == b\}|) > \frac{2}{3}N$ then

$C_i^{k,b,v}$

else if $|v_k^{'}| > \frac{2}{3}N$ then

$C_i^{k,\emptyset,v}$

else

$c(cv, vote)$. if $vote.round < k$ then

$cv(pr).\bar{c} < cv, pr > |VS_i^{k,p,v}(c)$

else if $vote.round == k$ then

$VS_i^{k,p,vote \cup v}(c)$

else

$P_i^{vote.round,p,vote \cup v}$

Procedure `Precommit()`

$C_i^{k,p,v}$ ::= $\overline{cc_i} < p > |(new\ c)(\prod_{j=1}^{N} cc_j(pr).\bar{c} < cc_j, pr > |CS_i^{k,p,v}(c))$

Procedure `PrecommitSub()`

$CS_i^{k,p,v}(c)$ ::= if $max_b(|\{com \in v_k^{''} : com.block = b\}|) > \frac{2}{3}N$ then

$\overline{cw_i} < b >$

else if $|v_k^{''}| > \frac{2}{3}N$ then

$P_i^{k+1,\emptyset,v}$

else

$c(pc, vote)$. if $vote.round < k$ then

$pc(com).\bar{c} < pc, com > |CS_i^{k,p,v}(c)$

else if $vote.round == k$ then

$CS_i^{k,p,vote \cup v}(c)$

else

$P_i^{vote.round,p,vote \cup v}$

夠大時，經過這種流水線操作後，時間近似為 $N * T1$。

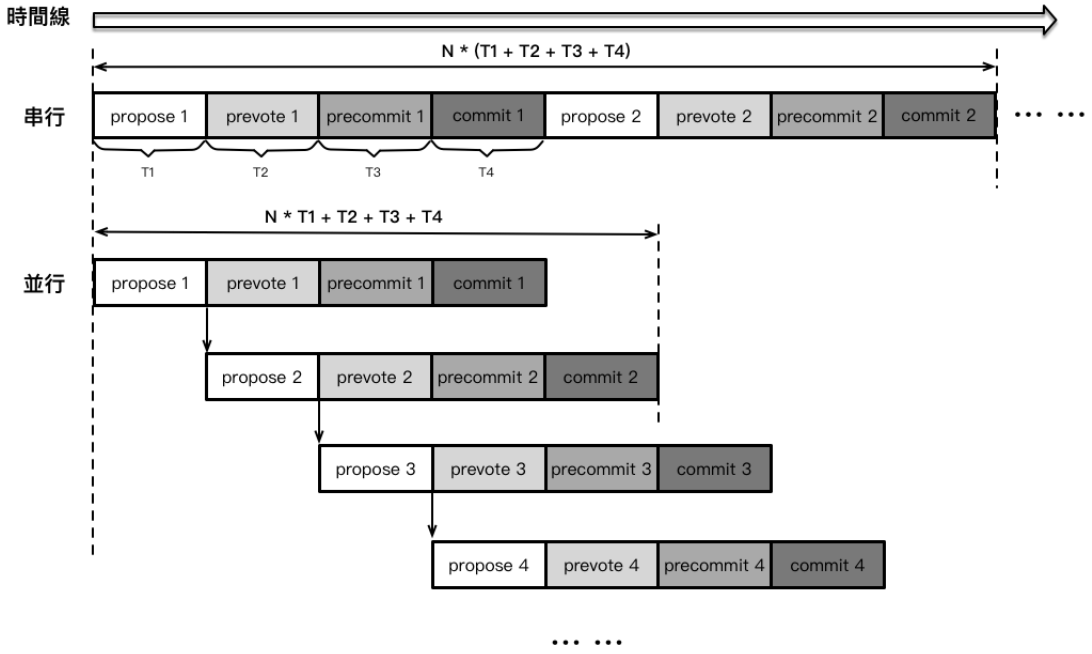

圖 11: 共識算法並行化

5 Dora 特色

5.1 橫向和縱向擴容

Dora 既支持常規的多鏈橫向擴容，還支持獨特的基於分支預測的縱向擴容機制，因此能獲得比其他系統更高的 TPS。

5.2 共識算法的並行化

Dora 的共識算法充分運用了流水線的思想，讓區塊的生成和確認兩個過程並行執行，進一步提升系統 TPS。

5.3 一鍵發幣和一鍵多轉

目前以太坊上最大的應用還是 ICO 發幣，即通過創建一個智能合約來發行代幣。ICO 代幣之間的轉賬本質上是調用智能合約的接口，在大量代幣轉賬交易時會導致網絡擁堵，浪費資源。Dora 支持特殊的鑄幣交易 MintAction，將發行代幣固化到母鏈上，讓發幣的行為變得和普通轉賬交易一樣容易。

```
MintAction: {
    name: "UToken",
    totalSupply: 100000000000000000000000000,
    decimals: 18,
    owner: "0xfbc2a4...ed",
    symbol: "UT"
}
```

同時針對目前代幣空投慢的問題，Dora 原生支持一鍵多轉賬交易 SendMultiTransaction。用戶能一次指定多個轉賬地址，把多筆交易打包在一起，節省網絡資源。

SendMultiTransaction: {
from: ""0xfbc2a4...ed"",
symbol: “UT”,
toLen: 4,
toList: {
"0xfbc2a4...e1", 100
"0xfbc2a4...e2", 500
...
}
}

6 Dora 經濟模型

Dora Network Limited 將推出 DNT(Dora Network Token) 通證以激活其內部生態系統。DNT 通證將由香港公司 Dora Network Limited 發行。DNT 旨在成為 Dora 網絡中使用的本機加密功能性通證，以及集成在 Dora 網絡內的分散式應用程序，在其自己的系統中循環。DNT 是 Dora Network 的燃料。DNT 通證的發行總量將是 10 億。DNT 通證是 Dora Network 內部交易費支付的 GAS。當用戶需要在 Dora Network 上進行計算以發送通證，或與合同進行交互等時，無論事務是成功還是失敗，用戶都必須為該計算付費。如果失敗，用戶必須支付該計算，就像它成功時需付交易費用一樣，因為礦工必須在任一場景（計算）中驗證並執行交易。該付款以 GAS 計算，並以 DNT 支付。GAS 是一個測量執行某些操作所需的計算工作量的單元。一般來說，Dora Network 有兩種類型的帳戶：外部帳戶和合同帳戶。所有外部帳戶都由私鑰控制，沒有代碼。Dora Network 用戶可以通過創建和簽署交易從外部帳戶啟動交易。合同賬戶由合同代碼控制。每當合同賬戶收到交易時，將執行合同代碼，包括讀/寫內部存儲，發送交易或創建合同等。

6.1 區塊獎勵

Dora 網絡將代幣的 5% 用作前 5 年的區塊獎勵，每年獎勵 1%。五年後，區塊獎勵的方式以及數量由社區投票決定。Dora 網絡允許節點自行設定分成比例，通過市場調節手段來激勵持幣用戶投票選取記賬節點。假設每個區塊的代幣獎勵個數為 W，某個節點願意採用比例 p 的代幣作為投票獎勵，那麼當該節點出塊時，所有投票給該節點的用戶則會按票數佔比來平攤 $W*p$ 的區塊獎勵，具體到每個投票用戶則為 $I_u = \frac{V_u}{\sum_{k=1}^{N} V_k} * W * p$。該節點獲得的區塊獎勵則包括兩部分，一部分是自己作為普通用戶投票給自己所獲得的投票獎勵，另外一部分則是作為記賬者所獲得的獎勵，具體計算公式為:

$$I_u = \frac{V_u}{\sum_{k=1}^{N} V_k} * W * p + W * (1 - p)$$

6.2 普通轉賬免費

在 Dora 網絡執行的交易都會產生交易費用。交易費用和 Dora 網絡的計算資源或者存儲資源的消耗相關。Dora 網絡中的計算資源或者存儲資源使用 GAS 數量表示。普通轉賬或者智能合約的交易費用（GAS 費用）的計算公式為：資源消耗的 GAS 數量乘以 GAS 價格。GAS 費用以及 GAS 價格都以 DNT 為計量單位。

Dora 網絡用戶普通轉賬時，可以抵押代幣換取免費轉賬額度。抵押越多則免費額度也越大。在抵押期完，代幣返還給用戶。為了避免零成本交易攻擊，Dora 網絡根據賬戶抵押代幣數來計算該賬號免費使用的 GAS 費用，超出免費使用配額的交易則會被實際扣除 GAS 費用。以 N 天為一個週期，在這個週期內賬號 U 的最大免費配額計算公式如下:

$$F_u = \frac{U\text{的當前鎖定代幣}}{\text{當前所有賬戶的鎖定代幣}} * \text{該週期內累計到當前區塊的交易總}GAS\text{費用}$$

6.3 社區治理及規劃

Dora 網絡的下列事項，需經過社區投票方式進行表決，持有 DNT 代幣的用戶擁有相應數量的投票權:

- 制定重要決策
- 區塊獎勵方式以及數量變更
- 緊急事件，如影響社區的事件、軟件安全、系統升級等

Dora 社區做出決議，必須獲得 DNT 代幣總量的 60% 以上通過（具體比例可以通過社區投票方式更改）。Dora 網絡本著公開透明的原則，由託管機構監督數字資產的流行並定期分享給社區。

Dora 基金會將提供 dApp 孵化計劃，以鼓勵 dApp 開發和生態系統建設。

7 總結和進一步工作

Dora 提出了智能合約並行化的 SSP 算法，設計了兼顧安全和效率的 DVBC 分層共識算法，採用多鏈技術，使整個系統能達到商業級別性能。並且，Dora 完全兼容以太坊 EVM，降低了 dApp 開發者的移植成本。此外，Dora 採用免費交易模式激勵生態進一步發展。

技術總是不斷進步的，Dora 會持續關注區塊鏈領域最新的研究方向，吸收和改進已有的研究成果，比如更高效安全的共識算法。另外會嘗試讓一些可信任的智能合約通過提前指定前置條件的方式來加速並行執行效率。也會考慮隨著上層應用的需要動態擴展鏈上支持的交易類型。

智能合約除了支持 EVM 之外，Dora 計劃支持 WebAssembly，讓更多的開發者能在 Dora 上快速開發 dApp。

致謝

感謝路印基金會和香港 Loopnest 加速器孵化項目對本項目的投資。本文的很多想法來源於路印[15]CEO 王東先生。他在論文撰寫過程中也給予了很多幫助和指導，在此表示由衷的感謝。

參考文獻

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`, Dec 2008.

[2] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. `https://github.com/ethereum/wiki/wiki/White-Paper`, 2014.

[3] Serguei Popov. The tangle. `https://www.iotatoken.com/IOTA_Whitepaper.pdf`.

[4] Ardor whitepaper. `https://www.jelurida.com/sites/default/files/JeluridaWhitepaper.pdf`, 2016.

[5] Jae Kwon and Ethan Buchman. Cosmos, a network of distributed ledgers. https://cosmos.network/resources/whitepaper.

[6] Asch whitepaper. https://github.com/AschPlatform/asch/blob/master/docs/whitepaper/index.md.

[7] PCHAIN Foundation. Pchain whitepaper. https://pchain.org/#whitepaper.

[8] Ethan Buchman. Tendermint: Byzantine fault tolerance in the age of blockchains. http://atrium.lib.uoguelph.ca/xmlui/bitstream/handle/10214/9769/Buchman_Ethan_201606_MAsc.pdf, Jun 2016.

[9] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network. https://lightning.network/lightning-network-paper.pdf, 2016.

[10] dantheman. Dpos consensus algorithm - the missing white paper. https://github.com/liuchengxu/blockchain-tutorial/blob/master/content/misc/dpos-consensus-algorithm-this-missing-white-paper.md, 2017.

[11] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science*, pages 120–130, New York, NY, October 1999. IEEE.

[12] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I. *Inf. Comput.*, 100(1):1–40, 1992.

[13] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, II. *Inf. Comput.*, 100(1):41–77, 1992.

[14] Robin Milner. *Communicating and mobile systems - the Pi-calculus*. Cambridge University Press, 1999.

[15] 王東, 周杰, and 王輝. Loopring（路印）: 去中心化代幣交易撮合協議. https://github.com/Loopring/whitepaper/raw/master/zh_whitepaper.pdf, 2018.

一般信息

本白皮書描述了銷售 Dora Network Token（DNT）的初始銷售。DNT 是一種加密通證，旨在按照本白皮書中的說明使用。DNT 不是，也不打算構成任何司法管轄區的證券，投資計劃，金融工具或任何其他受監管產品。本白皮書不是，也不打算構成招標，招股說明書，投資要約文件，並且不以任何方式涉及在任何司法管轄區內發行證券，投資計劃，金融工具或任何其他受管制產品。請注意，購買 DNT 是最終的，不可退款。個人，企業和其他組織應仔細權衡獲得 DNT 的風險，成本和收益。

購買者的限制

如果您是可能禁止購買 DNT 或類似加密貨幣，或代幣的銷售被視為不符合適用的法律法規的任何國家或地區的公民或居民 (稅務或其他方式)，您不能通過 Dora 網絡代幣銷售購買 DNT。為清楚起見，明確禁止任何自然人和居民 (稅務或其他)，或住所與美利堅合眾國，日本，中華人

民共和國有聯繫，參與代幣銷售和購買 DNT。DNT 的購買只能由對密碼通證和基於區塊鏈的軟件系統的使用和復雜性有重要經驗和熟練理解的自然人，實體或公司進行。購買者應該具有與其他加密通證相關的存儲和傳輸機制的功能性理解。Dora Network Limited 的任何實體及其官員和員工將不會以任何方式對因購買者採取的行動或遺漏而導致的任何加密通證，DNT 或法定貨幣的損失負責。如果您沒有所需的經驗或專業知識，那麼您不應購買 DNT 或參與通證服務。您應該仔細考慮獲得 DNT 的風險，成本和任何其他缺點，並在必要時，在這方面獲得您自己的獨立建議。如果您無法接受或了解與此通證銷售相關的風險，或本白皮書中指出的任何其他風險，則在收到必要的獨立建議之前，您不應購買 DNT。

風險

購買 DNT 會帶來風險。在購買 DNT 之前，購買者應仔細考慮下面列出的風險，並在確定是否購買 DNT 之前，在必要時諮詢律師，會計師和/或稅務專業人員。

(a) DNT 將存儲在錢包中，只能通過購買者選擇的密碼進行訪問。如果 DNT 的購買者未能保持其密碼的準確記錄，則可能導致其通證丟失。如果您的密碼保護較弱並且被其他人破解或學習，則可能還會導致通證丟失。因此，購買者必須將其密碼安全地存儲在與主要位置完全分離的一個或多個備份位置中。

(b) 購買者認識到 Dora 網絡生態系統中的某些服務目前正在開發中，並且可能在發布或可供使用之前進行重大更改。

(c) 購買者理解，儘管 Dora Network Limited 將盡最大努力按時發布 Dora 網絡，但可能會延遲正式發布。

(d) 與其他加密貨幣和密碼通證一樣，DNT 的價值可能會因各種原因，包括但不限於供求關係，整體市場狀況，政治或地理原因，任何司法管轄區的法規變更以及技術原因，出現大幅波動，導致價值下降，(包括歸零)。

(e) DNT 將在以太坊區塊鏈上發布。因此，以太坊協議的任何故障或意外功能都可能影響購買者轉移或安全保持 DNT 的能力。這種影響可能會對價值產生不利影響。

免責聲明

在適用法律，法規和規則允許的最大範圍內，Dora Network Limited，Dora Network Limited 生態系統內的任何實體及其官員和員工不對任何直接，間接，特殊，偶然，後果性或其他損失承擔責任。任何形式的侵權行為（包括疏忽），合同，法規或其他（包括但不限於收入損失，收入或利潤損失，使用或數據丟失），由您任何接受或依賴引起或與之相關在本白皮書或其任何部分。Dora Network Limited 及 Dora Network Limited 的任何實體及其官員和員工在因任何原因(包括但不限於您未能維護或備份您的密碼的準確記錄）轉移給您後，對 DNT 的任何損失不承擔任何責任。由於您的密碼維護不當，某人或密碼破解。任何承諾收購 DNT 的人都承認並理解 Dora Network Limited 不對 Dora Network 的發布或本白皮書中預期的任何其他技術特性或服務提供任何保證。因此，您承認並理解 Dora Network Limited（包括其相關機構公司，官員和員工）對因無法使用 DNT 而導致或與之相關的任何損失或損害不承擔任何責任或義務。監管機構正在仔細審查與世界上加密貨幣和代幣相關的業務和運營。在這方面，監管措施，調查或行動可能會影響未來的業務，並可能限製或阻止其未來的業務發展。任何承諾購買 DNT 的人必須知道，由於任何司法管轄區的任何適用法律的新監管和合規要求，Dora Network Limited 業務模式或 Dora 網絡可能會更改或需要進行修改。在這種情況下，購買者和任何承諾獲得 DNT 的人員承認並理解 Dora Network Limited 及其任何附屬公司均不對由此類變更引起的任何直接或間接損失或損害承擔責任。本白皮書以及 Dora Network Limited 及其官員和員工提出的任何其他

材料或解釋不得也不能被視為進行投資的邀請。它們不構成或以任何方式相關，也不應被視為在任何司法管轄區內發行證券，金融工具，投資計劃或任何其他受管制產品。本白皮書不包含也不包含任何可能被視為建議或可能被用作任何投資決策基礎的信息或指示。在任何法律，稅務或財務事宜中，Dora Network Limited 或其任何高級職員和員工均不應被視為顧問。收購 DNT 不得對 Dora Network Limited 組織和治理給予購買者任何權利或影響。

沒有陳述和保證

Dora Network Limited 不以任何形式就本白皮書中列出的信息向任何實體或個人作出或聲稱作出任何聲明，保證或承諾，包括任何與任何實體或個人的真實性，準確性和完整性有關的陳述，保證或承諾。此外，Dora Network Limited 對本白皮書中列出的任何計劃，未來預測或前景的成就或合理性均不作任何陳述或保證，本文檔中的任何內容均不得或應作為對此的承諾或陳述依賴。Dora 網絡和/或其相關服務的未來功能，實用程序或可用性。在法律允許的最大範圍內，Dora Network Limited 不對任何因本白皮書中包含的任何信息和意見採取行動而導致的任何類型（無論是否可預見）的損失或損害承擔任何責任（並且不承擔任何責任）。儘管 Dora Network Limited，其實體，官員和/或員工有任何作為或不作為，疏忽，違約或缺乏照顧，但任何進一步查詢所提供的文件或任何信息。

本白皮書以英文版本為準。如果發生法律糾紛或不一致，應以英文版本為準。