Поиск оптимальных комиссий за пользование каналами в сети Lightning Network

Токаева Александра

МГУ им. М. В. Ломоносова

Защита проекта по научным группам Веги, 1 июля 2022

Литература

Статьи

1) Optimally reliable and cheap payment flows on the lightning network, Rene Pickhardt and Stefan Richter, 2021.

2)

2) Routing Payments on the lightning network, Giovanni Di Stasi, Roberto Canonico, Stefano Avallone, Giorgio Ventre, 2018.

3)

3) Hubs, Rebalancing and service providers in the lightning network, Marco Conoscenti, Juan Carlos De Martin, Antonio Vetro, 2019.

Репозиторий с кодом

https://github.com/ElsevierSoftwareX/SOFTX-D-21-00019 Это репозиторий, в котором лежит код с симулятором CLOTH для Lightning Network, который можно запускать.

Цель работы

Задача:

Исследовать, как надо задавать комиссии за использование каналов в сети Lightning Network, чтобы большее количество платежей успешно доходило

Проблема скорости у биткойна

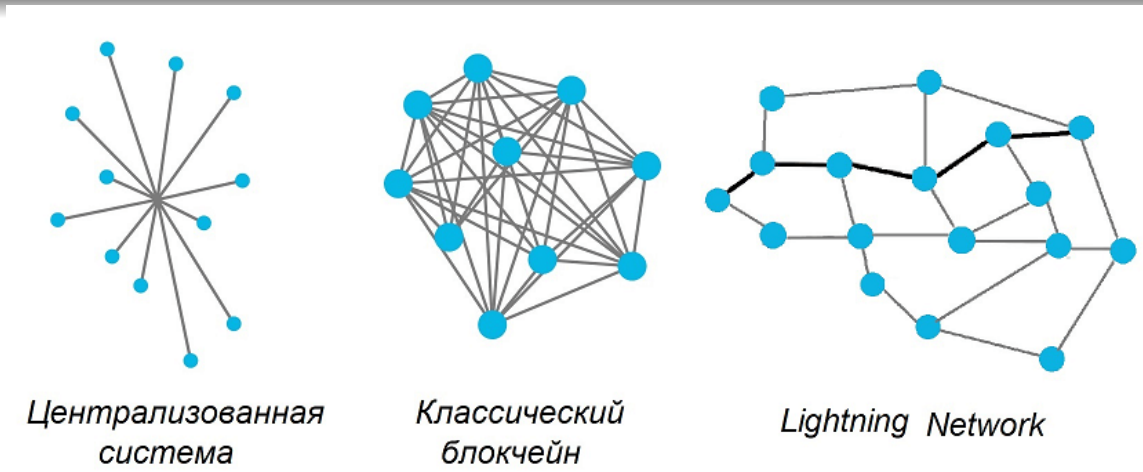
Биткойн

- Криптовалюта, созданная в 2009 году
- Реализована на базе технологии блокчейн
- Этот способ реализации решает проблему необходимости читать и писать из распределенного реестра в отсутствие единого узла, которому все доверяют

Блокчейн

- Действительно цепочка блоков
- Новые блоки добавляются примерно раз в 10 минут
- Из-за этого скорость работы биткойна очень маленькая - всего 3-7 транзакций в секунду
- Увеличение размера блока цепи имеет ограниченный эффект
- Решение проблемы - это Lightning Network

Что такое Lightning Network



- Это граф быстрых каналов, открытых поверх блокчейновских каналов
- Открыть-закрыть канал - долго, через транзакцию в основной цепи, а при уже открытом канале перегонять по нему деньги - очень быстро
- Пока канал не закрыт, то кошельки в основной цепи не видят никаких вообще изменений, хотя локальные кошельки в канале LN претерпевают изменения
- Чтобы никакие узлы не жульничали, используется протокол Revocable Sequence Maturity Contract

Проблемы передачи в LN

Недостаток баланса в промежуточном узле

- При открытии канала AB узел A вложил в канал a биткойнов, узел B вложил b биткойнов, и в канале всегда лежит a+b биткойнов
- Меняться будет только то, какая часть x из этих a+b биткойнов лежит в узле A (в узле B соответственно лежит a+b-x биткойнов).
- И если мы просим канал AB пробросить вперед платеж размера P, то если x<P - то платеж не сможет проброситься из-за недостатка баланса у узла A.

Постановка задачи

- Изменяя комиссии за пользование ребром, мы будем заставлять платежи выбирать пути, на которых узлы содержат много баланса
- Чем больше у узла баланс - тем дешевле потоку идти через него
- Тем самым у платежа больше шансов успешно дойти
- Кроме балансировки каналов (для уменьшения числа случаев недохождения платежа по причине отсутствия баланса в одном из промежуточных узлов), мы хотим научиться искать оптимальный путь для проведения конкретного платежа (при заданных fees на каждом из ребер).

Определение графа LN

Граф

Граф LN - это ориентированный граф $G = (V, E)$, где на каждом ребре (ориентированном) написано число $u : E -> N$, которое соответствует capacity ребра в эту строну.

Поток

Назовем функцию $f : E \to N_0$ потоком, если:
1) $\forall e \in E : 0 \leq f(e) = f_e \leq u_e = u(e)$
2) $\forall i \in V : \sum_{(i,j)\in E} f_{ij} - \sum_{(j,i)\in E} \gamma_{ij} f_{ji} = b_i$

Cost function

В статье 1 предложены две функции штрафов:

Мы можем хотеть максимизировать вероятность дохождения платежа, то есть:
$P(f) = \prod_{e \in E} P(X_e \geq f_e) \to max$
Это равносильно минимизации $\sum_{e \in E} -logP(X_e \geq f_e) \to min$

А можем хотеть минимизировать штрафы за пользование каналами :
$\sum_{e \in E} f_e \cdot fee(e) \to min$

А можем хотеть минимизировать линейную комбинацию первого и второго подходов :
$\sum_{e \in E} -logP(X_e \geq f_e) + \mu \cdot f_e \cdot fee(e) + \nu \to min$

Откуда какие-то вероятности?

- Обычно мы знаем только вместимость канала, но какая доля баланса канала лежит в левом узле - мы не знаем
- Мы предполагаем, что баланс левого узла $R[0, a + b]$ и дальше обновляем это распределение, учитывая, платежи какого размера смогли пройти по этому ребру, а какие-нет
- Отсюда и вероятности - ведь мы не знаем точный баланс левого узла, только его распределение

Обновление весов

- То есть было $P(X \geq a) = \frac{u+1-a}{u+1}$
- если по ребру смог пройти платеж размера h, то пишем $P(X \geq a + h | X \geq h) = \frac{(u-h)+1-a}{(u-h)+1}$
- Для всех узлов ДО упавшего узла - у них баланс не меньше, чем h, то есть $P(X \geq a | X \geq h) = 1, a \leq h$, а для $a \geq h$ - равномерное распределение $R[h, u]$
- А для упавшего узла меняем capacity на h-1, то есть $P(X \geq a) = \frac{h-a}{h}$

Алгоритм поиска пути

Как найти путь для потока?

- Оптимальный путь - это min-cost max-flow в uncertainty network
- Его находит алгоритм Форда-Фалкерсона (поиска максимально потока)
- На случай, если максимальный поток g в графе окажется больше, чем поток h, который мы хотим протолкнуть, мы перед графом добавляем ребро в точности вместимости h

Управление fees для поддержания сбалансированности узлов

В статье 3 предложено два способа управление fees для поддержания баланса узлов:

- 1) active rebalancing: для истощившегося узла искать более сытого соседа, принудительно отбирать у этого соседа часть баланса и отдавать голодному узлу;
- 2) passive rebalancing: увеличивая fees на ребрах, у которых истощился баланс в нужную сторону, и уменьшая fees на тех ребрах, у которых много баланса в нужную сторону

Active and passive rebalancing

Passive - лучше!

- В третьей статье с использованием симулятора CLOTH были проведены исследования для сравнения active или passive rebalancing.
- active rebalancing почти не дало улучшения ситуации, потому что почти всегда у истощившегося узла не было соседей, у которых баланс был бы существенно больше плоловины (и у которых, соответственно, было бы не жалко забрать часть баланса.
- passive rebalancing уменьшил число недохождений платежа из-за отсутствия баланса на 20 процентов.

Что еще в 3 статье

- для нахождения оптимального пути использовался алгоритм Дейкстры.
- Отметим, что при passive rebalancing использовалась cost function, в которой не было первого слагаемого (отвечающего пероятности прохождения платежа) и второго (пропорционального величине платежа), а третье было равно константе, равной $1/xleft$.

Идея двойного наклона во 2 статье

$Imb = |r_{a,b} - r_{b,a}|$;
if $r_{a,b} > r_{b,a}$ **then**
  if $T > Imb/2$ **then**
    $f_{a,b}(T) = b + (Imb/2) * s_{low} + (T - Imb/2) * s_{high}$;
  else
    $f_{a,b}(T) = b + T * s_{low}$;
  end
else
  $f_{a,b}(T) = b + T * s_{high}$;
end

Algorithm 1: OptimizedFees: algorithm for fee calculation that fosters channel balancing.

• Во второй статье приведена идея, что при пассивной балансировке нужно использовать не только base fee (которое обратно пропорционально xleft), а еще включать слагаемое $\mu \cdot f_e$, причем μ делать не постоянным, а кусочно-линейным с двумя кусками.

Что было добавлено в симуляции

- 1) Использовать алгоритм Дейкстры (вместо Форда-Фалкерсона) для поиска оптимального пути
- 2) При пассивной балансировке, кроме константы ν, которая обратно пропорциональна *xleft*, использовать еще второе слагаемое (причем сначала с μ константной, потом с кусочно-линейной) и первое (которое отвечает за максимизацию вероятности дохождения платежа)

3) Подобрать константы-веса у cost-function и двойного наклона оптимальным образом

Симуляции в CLOTH

Результаты

- Используем готовый симулятор CLOTH
- были воспроизведены результаты статьи 3
- были проведены симуляции, реализующие идеи 2 и 3 (а идея 1 там и так реализована)
- Если качественно подобрать параметры - то есть улучшения!

Спасибо за внимание

Спасибо за внимание!