Отчет по научным группам: Поиск оптимальных комиссий за пользование каналами в сети Lightning Network

Александра Токаева

30 июня 2022

1 Описание

Работа выполнена под руководством Ростислава Березовского на основании трех статей. В работе исследуется, как нужно оптимально задавать комиссии за пользование каналами в сети Lightning Network для того, чтобы уменьшить число случаев, когда платех не дойдет из пункта A в пункт B из-за того, что в каком-то из промежуточных узлов не хватило баланса.

2 Использованная литература

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

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

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

4)https://github.com/ElsevierSoftwareX/SOFTX-D-21-00019

Это репозиторий, в котором лежит код с симулятором CLOTH для Lightning Network, который можно запускать с нужными параметрами графа и делать выводы о работе графа Lightning Network.

3 Откуда берется Lightning Network

Сеть Lightning Network - это граф из быстрых каналов, построенный поверх сети биткойна. Проблема биткойна заключается в том, что из-за того, что он реализован посредством технологии блокчейн (цепочка блоков), то чтобы провести какую-то транзакцию, нужно дождаться, чтобы майнеры добавили в основную цепь биткойна новый блок (в котором находится не только интересующая нас транзакция). Новый блок добавляется в среднем один раз в 10 минут. Это значит, что производительность биткона сильно ограничена, он значительно медленне, чем транзакции по карте Visa, например. Увеличение размера блока решает проблему только частично, и при этом еще предъявляет дополнительные технические требования к узлам цепи, что нежелательно. Как альтернативное решение, появилась сеть Lightning Network. Она позволяет двум узлам открыть канал с определенным фиксированным количеством денег a+b в канале (один раз долго подождав, пока операция открытия канала исполнится в основной цепи биткойна), а потом по уже открытому каналу перегонять деньги можно очень быстро, при этом количество денег в канале всегда остается равным a+b, меняется только локальный баланс узла A (то есть сколько денег из этих a+b принадлежит узлу A). Для закрытия канала тоже придется отправить транзакцию в основную сеть и долго подождать. За быстроту обмена деньгами в открытом канале платится тем, что кошельки узлов в основной сети ничего не знают об изменении балансов этих узлов в сети Lightning Network (и увидят эти изменения толко после закрытия канала). Это значит, что даже если в глобальном кошельке у узла A много денег, а в локальном кошельке у него только x биткойнов из a+b биткойнов, лежащих в канале- то платеж больше x, узел A пробросить вперед не сможет. С этой проблемой недостаточного баланса у одного из узлов на пути платежа мы и будем бороться - выставляя большие комиссии за пользование каналами, у левого узла которых мало баланса, и маленькие комиссии- за пользование каналами, у которых в левом узле много баланса.

4 Математическая постановка

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

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) $\forall i \in V : \sum_{(i,j)\in E} f_{ij} - \sum_{(j,i)\in E} \gamma_{ij} f_{ji} = b_i$

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 $\sum_{e \in E} -logP(X_e \geq f_e) \to min$
В статье 1 предложены две функции штрафов:

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 взялись? Обычно мы знаем только вместимость канала, но какая доля баланса канала лежит в левом узле - мы не знаем. Поэтому мы предполагаем, что баланс левого узла распределен как $R[0, a+b]$ 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.
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 $R[h, u]$

5 Алгоритм поиска пути и управление комиссиями

5.1 Как найти путь?

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.

5.2 Active and passive rebalancing

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.

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.
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 $1/xleft$.

5.3 Идея двойного наклона

$Imb = |r_{a,b} - r_{b,a}|$;
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.

6 Результаты

Были предложены следующие улучшения:
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.

Используя выложенный на гитхабе код симулятора CLOTH, были воспроизведены результаты статьи 3, а также были проведены симуляции, реализующие идеи 2 и 3 (а идея 1 там и так реализована). Выяснено, что если качественно подобрать параметры - то есть улучшения!