Распределенные алгоритмы и системы

mk.cs.msu.ru → Лекционные курсы → Распределенные алгоритмы и системы

Блок 18

Алгоритм Туэга

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2022/2023, весенний семестр

Задача и допущения

Требуется в каждом узле v распределённой системы для каждого узла w вычислить значение $next(v, w)$ следующей вершины оптимального пути из v в w ($\perp$, если $v = w$ или w недостижима из v)

Топология сети и функция веса определяются так же, как и в алгоритме Флойда-Уоршелла

Основные допущения, использующиеся дальше в обсуждении алгоритма Туэга:

1. Каждый цикл в сети имеет положительный вес
2. Каждый узел знает имена всех узлов (множество V) и их порядок (один и тот же для всех узлов)
3. Каждый узел v знает имена соседей (множество $neigh_v$) и веса каналов, соединяющих его со соседями ($\omega_{(v,w)}$, где $w \in neigh_v$)

Обсуждение алгоритма Туэга принято начинать с его упрощённой версии (для лучшего восприятия)

Упрощённый алгоритм Туэга

Отличия упрощённого алгоритма Туэга от алгоритма Флойда-Уоршелла

Переменные и операции над ними разносятся по узлам сети:

- Переменная $d[v, w]$ помещается в узел v
- Чтобы нагляднее обозначить узел этой переменной, будем её записывать как $d_v[w]$

Изменение значения $d_v[w]$ происходит в узле v, и для этого изменения узел предварительно получает все используемые значения

Вершина u, добавляющаяся в множество S в действии 3 алгоритма Флойда-Уоршелла (будем называть её опорной), отправляется всем узлам при помощи особой команды «$broadcast(u)$»

Помимо весов $\delta(v, w)$ на основании этих весов в узле v вычисляются значения $Nx_v[w] = next(v, w)$

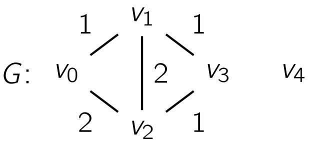
Упрощённый алгоритм Туэга

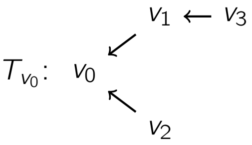
Лемма (об отсутствии циклов). Пусть:

- **Заданы множество вершин S и вершина w**
- **Для всех вершин v верно $d_v[w] = \delta^S(v, w)$**
- **Если $\delta^S(v, w) < \infty$ и $v \neq w$, то $Nx_v[w]$ — имя вершины, следующей за v в оптимальном пути из v в w**
- **$T_w = (V_w, E_w)$ — ориентированный граф, в котором**
  - $V_w = \{v \mid v \in V,\ d_v[w] < \infty\}$
  - $E_w = \{(v, u) \mid (v, u) \in V_w^2,\ v \neq w,\ Nx_v[w] = u\}$

Тогда T_w — дерево с корнем (стоком) w

Иллюстрация:





Упрощённый алгоритм Туэга

Доказательство.

По лемме о подпутях, если для вершины v, $v \neq w$, верно $D_v[w] < \infty$, то $Nx_v[w] = u \neq \perp$ и $d_u[w] < \infty$

Следовательно, для каждой вершины $x \in V_w$, $x \neq w$, существует (единственная) вершина $y \in V_w$, такая что $Nx_x[w] = y$

То есть для каждой $x \in V_w$ в E_w существует единственная дуга $x \to \_$

При этом по построению из w в T_w не исходит ни одной дуги

Следовательно, $|E_w| = |V_w| - 1$

Предположим от противного, что в T_w существует цикл $x_1 \to \cdots \to x_k \to x_1$

Так как для любой дуги $(x, y) \in E_w$ верно $\delta^S(x, w) = \omega_{(x,y)} + \delta^S(y, w)$, то верно равенство

$$\delta^S(x_1, w) = \omega_{(x_1,x_2)} + \omega_{(x_2,x_3)} + \cdots + \omega_{(x_{k-1},x_k)} + \omega_{(x_k,x_1)} + \delta^S(x_1, w)$$

Но тогда $0 = \omega_{(x_1,x_2)} + \omega_{(x_2,x_3)} + \cdots + \omega_{(x_{k-1},x_k)} + \omega_{(x_k,x_1)}$, а согласно допущению 1, каждый цикл имеет положительный вес (*противоречие*)

Итого, T_w — граф без циклов, в котором количество дуг на 1 меньше количества вершин, а значит, дерево ▼

Упрощённый алгоритм Туэга

Собственно упрощённый алгоритм Туэга для узла v:

1. $S_v := \emptyset$;
2. Для каждой $u \in V$:
    - 2.1 Если $u = v$:
        - 2.1.1 $d_v[u] := 0$;
        - 2.1.2 $Nx_v[u] := \perp$;
    - 2.2 Если $u \in neigh_v$:
        - 2.2.1 $d_v[u] := \omega_{(v,u)}$;
        - 2.2.2 $Nx_v[u] := u$;
    - 2.3 Иначе:
        - 2.3.1 $d_v[u] := \infty$;
        - 2.3.2 $Nx_v[u] := \perp$;
3. Пока $S_v \neq V$:
    - 3.1 Произвольно выбрать $w \in V \setminus S_v$ (одинаковую для всех узлов)
    - 3.2 Если $v = w$, то $broadcast(d_w)$, иначе $receive(d_w)$
    - 3.3 Для всех $u \in V$:
        - 3.3.1 Если $d_v[w] + d_w[u] < d_v[u]$:

          $d_v[u] := d_v[w] + d_w[u]$;

          $Nx_v[u] := Nx_v[w]$;
    - 3.4 $S_v := S_v \cup \{w\}$;

Упрощённый алгоритм Туэга

Теорема (о корректности упрощённого алгоритма Туэга).
Упрощённый алгоритм Туэга в каждом узле v завершается после $|V|$ итераций в действии 3, и после завершения для любого узла w верно

- $d_v[w] = \delta(v, w)$ **и**
- **если** $\delta(v, w) < \infty$ **и** $v \neq w$**, то** $Nx_v[w] = next(v, w)$**,**

и если граф G связен, то таблицы маршрутизации сети гарантируют доставку пакетов каждому адресату

Доказательство.

Завершаемость и корректность значений $d_v[w]$ следуют из корректности алгоритма Флойда-Уоршелла

Корректность значений $Nx_v[w]$ следует из того, что они изменяются тогда же, когда и соответствующие $d_v[w]$, и новое значение есть следующая вершина в пути веса $d_v[w]$

Гарантия доставки пакетов обеспечивается леммой об отсутствии циклов и леммой об ациклических таблицах ▼

Алгоритм Туэга

К сожалению, алгоритм Туэга, согласно рассказанному сейчас в лекциях, нереалистичен из-за команды *broadcast*

Осмыслить эту команду как реалистичную помогут волновые алгоритмы, но они будут обсуждаться в лекциях существенно позже

А пока, чтобы избежать этой нереалистичности, можно использовать следующее наблюдение

Если $d_v[w] = \infty$, то для каждой вершины u верно $d_v[w] + d_w[u] = \infty \geq d_v[u]$, а значит, если в действии 3 в этом случае выбирается опорная вершина w, то таблица маршрутизации узла v на этой итерации действия 3 не изменяется

Поэтому таблицу d_w в действии 3.2 достаточно отправить только в узлы дерева T_w, используя только каналы этого дерева

Для этого узел w может отправить d_w своим детям в T_w, и каждый узел T_w, получив d_w, переслать эту таблицу своим детям

Алгоритм Туэга

В начале действия 3 с опорной вершиной w каждый узел v, для которого верно $d_v[w] < \infty$, знает своего родителя в T_w, но не знает детей

Чтобы каждый узел узнал о своих детях в T_w, достаточно «заставить» каждый узел u послать всем своим соседям сообщение о том, является ли этот сосед родителем u в T_w

Получив такие сообщения от всех соседей, каждый узел v узнаёт о всех своих детях в T_w — обо всех тех, кому следует переслать таблицу d_w в случае получения

Алгоритм Туэга

Инициализация в узле v ($Init_v$):

1. $S_v := \emptyset$;
2. Для всех $u \in V$:
   - 2.1 Если $u = v$:
     - 2.1.1 $d_v[u] := 0$;
     - 2.1.2 $Nx_v[u] := \perp$;
   - 2.2 Если $u \in neigh_v$:
     - 2.2.1 $d_v[u] := \omega_{(v,u)}$;
     - 2.2.2 $Nx_v[u] := u$;
   - 2.3 Иначе:
     - 2.3.1 $d_v[u] := \infty$;
     - 2.3.2 $Nx_v[u] := \perp$;

Алгоритм Туэга

Выявление детей в узле v ($Children_v$):

1. Для всех $u \in neigh_v$:
   1.1 $send((\mathbf{par}, w), (Nx_v[w] = u))$ в канал к соседу u
   1.2

2. $Nrec_v := 0$; $children := \emptyset$;

3. Пока $Nrec_v < |neigh_v|$:
   3.1 $receive((\mathbf{par}, w), b)$ от любого соседа u
   3.2 Если b: $children := children \cup \{u\}$;
   3.3 $Nrec_v := Nrec_v + 1$;

Сообщение $((\mathbf{par}, w), \mathbb{t})$ с отправителем u и получателем v означает, что узел v является родителем узла u в дереве T_w, а $((\mathbf{par}, w), \mathbb{f})$ — что не является

Дети узла v в T_w — это все соседи, которые отправили сообщение $((\mathbf{par}, w), \mathbb{t})$

Алгоритм Туэга

Уточнение таблиц маршрутизации в узле v ($Recompute_v$):

1. Если $d_v[w] < \infty$:
   - 1.1 Если $v \neq w$:
     - 1.1.1 $receive((\mathbf{dtab}, w), d_w)$ от $Nx_v[w]$
   - 1.2 Для всех $u \in children$:
     - 1.2.1 $send((\mathbf{dtab}, w), d_w)$ соседу u
   - 1.3 Для всех $u \in V$:
     - 1.3.1 Если $d_v[w] + d_w[u] < d_v[u]$:
       $d_v[u] := d_v[w] + d_w[u]$;
       $Nx_v[u] := Nx_v[w]$;

Узел w располагает таблицей d_w и отсылает её всем детям в T_w

Каждый другой узел получает таблицу d_w от родителя (и теперь может её использовать) и пересылает детям

Так как в связной сети вершинами T_w являются все узлы, то рано или поздно каждый узел дойдёт до действия 1.3, располагая таблицей d_w

Алгоритм Туэга

Алгоритм Туэга (полный):

1. $Init_v$
2. Пока $S_v \neq V$:
    2.1 Выбрать $w \in V \setminus S_v$ (общий для всех узлов сети)
    2.2 $Children_v$
    2.3 $Recompute_v$
    2.4 $S_v := S_v \cup \{w\}$;

Алгоритм Туэга

Теорема (о корректности алгоритма Туэга). **Алгоритм Туэга в каждом узле v завершается после $|V|$ итераций в действии 2, и после завершения для любого узла w верно**

- $d_v[w] = \delta(v, w)$ **и**
- **если** $\delta(v, w) < \infty$ **и** $v \neq w$**, то** $Nx_v[w] = next(v, w)$**,**

и если граф G связен, то таблицы маршрутизации сети гарантируют доставку пакетов каждому адресату

Доказательство. Следует из

- корректности упрощённого алгоритма Туэга,
- корректности процедуры $Children_v$: после её выполнения в таблице $child_v$ значениями $\mathbb{t}$ обозначены все дети вершины v в T_w — и
- того, что каждый узел v входит в T_w и поэтому на этапе $Recompute_v$ рано или поздно принимает таблицу d_w, рассылавшуюся широковещательно в упрощённом алгоритме

Корректность процедуры $Children_v$ следует из определения дерева T_w и того, что узел u — ребёнок узла v в этом дереве $\Leftrightarrow$ u — сосед v и v — родитель u ▼

Алгоритм Туэга

Пусть

- $N = |V|$, $M = |E|$ и
- W — число битов, использующееся для записи имени вершины и веса пути

Теорема (о сложности алгоритма Туэга). **Алгоритмом Туэга отправляется**

- $O(N)$ **сообщений и** $O(N^2 W)$ **битов в каждый канал и**
- $O(NM)$ **сообщений и** $O(N^3 W)$ **битов всего, и**

в каждом узле алгоритма используется $O(NM)$ **битов памяти**

Доказательство.

Для каждой опорной вершины w в каждый канал отправляются

- два сообщения типа (**ch**, w) (по одному в каждом направлении) и
- не более одного сообщения типа (**dtab**, w)

Сообщение типа **ch** содержит $O(W)$ битов, типа **dtab** — $O(NW)$ битов

Из этого следуют оценки сложности для канала

Алгоритм Туэга

Пусть

- $N = |V|$, $M = |E|$ и
- W — число битов, использующееся для записи имени вершины и веса пути

Теорема (о сложности алгоритма Туэга). Алгоритмом Туэга отправляется

- $O(N)$ **сообщений и** $O(N^2 W)$ **битов в каждый канал и**
- $O(NM)$ **сообщений и** $O(N^3 W)$ **битов всего, и**

в каждом узле алгоритма используется $O(NM)$ **битов памяти**

Доказательство.

При выполнении алгоритма суммарно в каналы отправляется не более N^2 сообщений типа **dtab** и $2NM$ сообщений типа **ch**

Из этого, связности сети ($M \geq N - 1$) и битового размера сообщений следуют оценки сложности для всех каналов сети

Для хранения самых объёмных данных: таблиц d_v, d_w, Nx_v — требуется $O(NW)$ битов памяти: N значений, и не более W битов каждое ▼

Алгоритм Туэга

Задача 1. Зачем в алгоритме Туэга во всех сообщениях передаётся имя w опорной вершины? Можно ли не передавать w хотя бы в каких-нибудь из сообщений? Если нет, то почему алгоритм перестанет быть корректным?

Задача 2. Можно ли модифицировать алгоритм так, чтобы не передавать сообщения $((\mathbf{par}, w), \mathbb{f})$, мотивируя это тем, что узел может по умолчанию полагать, что у него нет детей? Если нет, то почему алгоритм перестанет быть корректным?

Алгоритм Туэга

Достоинства алгоритма Туэга:

- Простой
- Имеет невысокую сложность
- Вычисляет веса всех оптимальных путей

Недостатки:

- При имзенении топологии сети требуется перевычислять всё «с ноля»
- Должны быть заранее согласованы множество V и порядок выбора опорных вершин из него
- При перевычислении значений d_v используется таблица, которая в общем случае изначально не доступна ни узлу, ни его соседям и должна быть доставлена поочерёдно из всех узлов сети