

Gossip

Требуется реализовать разновидность протокола Gossip.

Описание Gossip

Узел N в сети получает от узла M сообщение msg , которое нужно распространить по всей сети. После этого узел N через равные интервалы времени t отправляет сообщение какому-либо из своих соседей, кроме узла M . Узел N отправляет msg ровно tll раз, после этого считается, что сообщение msg устарело. Если узел N ранее получал сообщение msg , то только что полученное сообщение msg игнорируется.

Задание

Требуется реализовать разновидность протокола Gossip.

Узел под номером 0 начинает распространять по сети сообщение с помощью вышеописанного протокола.

Помимо рассылки исходного сообщения каждый узел при получении сообщения начинает распространять по сети сообщение о том, что он получил сообщение.

Нужно посчитать через какое количество времени T узел 0 получит уведомление о получении сообщения от каждого узла. T измеряется в количестве интервалов t . Параметр tll для этой задачи можно игнорировать.

Каждый узел сети нужно запустить в отдельной goroutine внутри одного процесса. Общение между узлами ведётся посредством UDP.

Составить графики зависимости времени T от доли потерь сообщений в сети.

Для симуляции потерь на localhost можно использовать iptables.

Следующая команда добавит в iptables правило для потери 10% входящих udp пакетов на localhost.

```
$ iptables -A INPUT -p udp -i lo \
-m statistic --mode random --probability 0.1 -j DROP
```

Следующая команда удалит из iptables созданное правило.

```
$ iptables -D INPUT -p udp -i lo \
-m statistic --mode random --probability 0.1 -j DROP
```

Для представление сети нужно использовать пакет graph (<https://gitlab.com/n-canter/graph>).

Пример udp-сервера и клиента можно посмотреть по ссылке: <https://goo.gl/ds6wFK>.

Для сдачи задания нужно прислать письмо на max.shegai@gmail.com со ссылкой на репозиторий с кодом. В репозитории помимо кода должен содержаться построенный график.

Репозиторий может быть на любом трэкере (github, gitlab, bitbucket итд).

В теме письма ОБЯЗАТЕЛЬНО указать номер группы.

Формат сообщений

Сообщения отправляются в формате JSON. Сообщения состоят из 5 полей:

1. id – целое число, id сообщения
2. type – тип сообщения (multicast, notification)
3. sender – номер узла, отправляющего сообщение
4. origin – номер узла, отправившего исходное сообщение
5. data – строка, содержащая данные

Пример исходного сообщения:

```
{
  "id": 100500,
  "type": "multicast",
  "sender": 1,
  "origin": 0,
  "data": "bla-bla-bla"
}
```

Пример сообщения-уведомления о получении:

```
{
  "id": 32,
  "type": "notification",
  "sender": 20,
  "origin": 777,
}
```