

БАЛАНСИРОВКА НАГРУЗКИ В ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ НА КАНАЛЬНОМ УРОВНЕ ИЛИ ПРОТОКОЛ «СПРОСИ СОСЕДА»

**Зайцев Е.В., Колмаков С.В.,
научный руководитель канд. техн. наук Казаков Ф.А.
Сибирский Федеральный университет**

Балансировка нагрузки в локальной вычислительной сети является процессом распределения передаваемой информации между каналами связи с целью снизить загрузку этих каналов.

Основной проблемой задержки сетевого трафика являются так называемые очереди. Очереди образуются в результате большого количества информации проходящего через коммутатор. Так как аппаратная часть коммутатора, состоящая из буферной памяти, и процессора имеет ограниченные характеристики, описанные производителем, то при определенных нагрузках устройство физически не успевает обработать все запросы, приходящие на вход.

Для решения этой проблемы был придуман протокол с использованием обратной связи для балансировки нагрузки с использованием соседних коммутаторов.

Общее описание протокола:

Основная идея данного протокола балансировки является распределение передаваемых кадров между каналами коммутатора «балансирующего» основываясь на информации об очередях соседнего коммутатора именуемого «инициатор балансировки» с целью уменьшения или полного устранения задержек передачи данных в данном сегменте.

Достоинства:

- Протокол позволяет использовать все возможные линии связи при образовании задержек передачи из-за очередей на более приоритетных маршрутах.
- Принятие решения о балансировки принимаются с участием соседей.

Недостатки:

- Коммутатор должен знать хотя бы два абсолютно разных маршрута до других коммутаторов в сети, и эти маршруты должны быть активны.

Моделирование устройства для проведения эксперимента:

Для того чтобы смоделировать работу коммутатора на ЭВМ и создать протокол балансировки за основу была взята модель очереди M/M/1. Данная модель используется в теории очередей относящихся к ветви прикладной математики для моделирования устройств работающих с очередью. Основными элементами данной модели являются:

- Входной поток абстрактных данных
- Буфер памяти
- Обслуживающее устройство
- Выходной поток обслуженных заявок

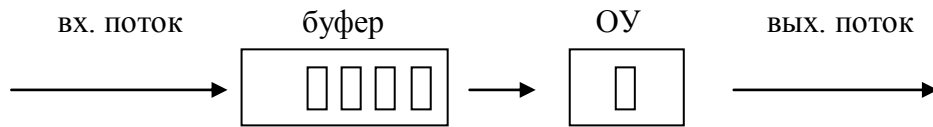


рисунок 1.

Для реализации процесса балансировки нам необходимо определить наличие очереди в буфере коммутатора, для того чтобы инициировать процесс балансировки. Иными словами нам нужно найти среднее время ожидания заявки в очереди, исходя из которого мы будем пытаться «просить соседа» перенаправить поток данных по другому маршруту минуя инициатора балансировки.

Для нахождения ω нам понадобятся следующие величины:

λ - интенсивность поступления заявок на устройство

$$\lambda = \frac{1}{T}$$

где T - среднее время между поступлениями заявок;

μ - интенсивность продвижения заявок в очереди

$$\mu = \frac{1}{b}$$

где b - среднее время обслуживания заявок устройством;

Стоит заметить, что T берется в усредненном значении, а b это техническая характеристика коммутатора, заявленная производителем.

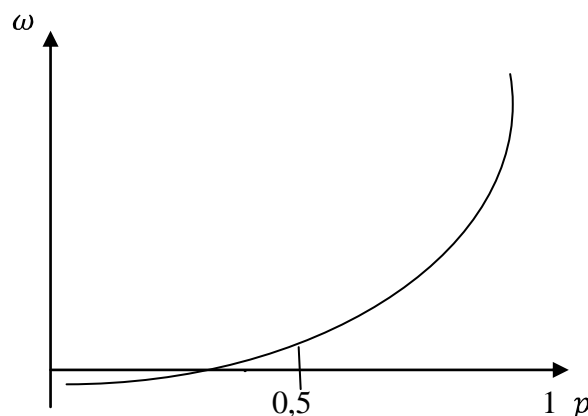
Исходя из этих данных, мы можем вычислить среднее время ожидания заявки в очереди, перед тем как она попадет в обслуживающее устройство (процессор):

$$\omega = p \frac{b}{1-p}$$

где p , это коэффициент использования обслуживающего прибора, который равен отношению интенсивности поступления заявок к интенсивности продвижения заявок т.е.:

$$p = \frac{\lambda}{\mu}$$

Зависимость среднего ожидания заявки от коэффициента использования ресурса показана на графике:



Как видно из поведения кривой, параметр p играет ключевую роль в образовании очереди. Если p близко к нулю, то среднее время ожидания тоже близко к нулю, а это означает, что заявки почти никогда не ожидают своей очереди на обработку в буфере, а сразу попадают в обслуживающее устройство.

Пример:

Теперь рассмотрим на примере работу данного алгоритма. Допустим, мы имеем компьютерную сеть, состоящую из шести коммутаторов. Каждый коммутатор имеет от одного до трех типов соединений «точка-точка» с другими коммутаторами.

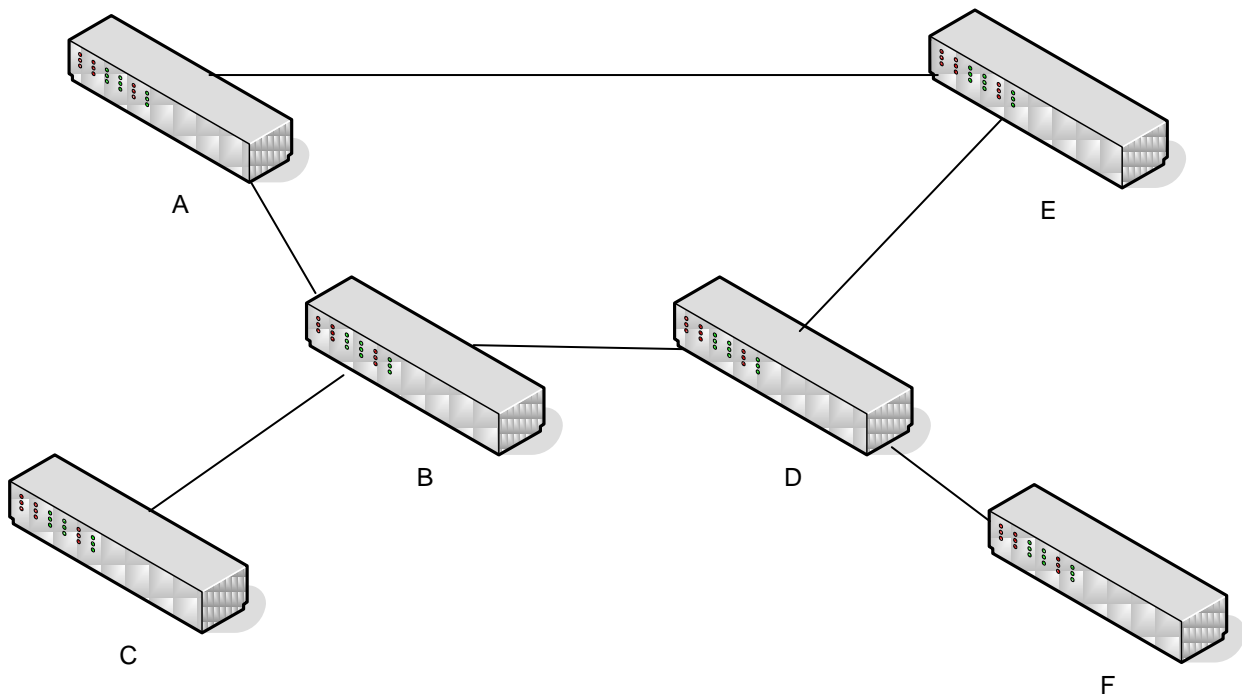


рисунок 2.

В определенный момент времени коммутатор «В» обнаружил наличие «длинной» очереди в своем буфере, в связи с тем скорость передачи данных через него в сторону коммутатора «D» до коммутатора «F» резко снизилась. В этот самый момент коммутатор «В» становится инициатором балансировки. Коммутатор В определяет с каких соседних коммутаторов идет наибольшее число данных, проверяя пришедшие кадры; спрашивает соседние коммутаторы о возможности передачи данных через другой маршрут, минуя инициатора балансировки. Если такой маршрут имеется, то коммутаторы «А» и «С» отсылают инициатору «В» свои метрики до коммутатора «F». На основе полученных данных «В» решает отправить «А» сообщение об уменьшении кол-ва передаваемого трафика через инициатора «В». Так как в коммутаторе «А» есть альтернативный маршрут до «F» через «Е», большая часть данных будет идти по этому маршруту, пока инициатор балансировки не сообщит о том, что очередь в его буфере уменьшилась до минимального значения.

Вывод: Реализация данного протокола позволит нам использовать все альтернативные пути для доставки данных, а не только наиболее приоритетные. Данный протокол является расширением к сетевым протоколам, целью которых является поиск оптимальных маршрутов.