

АНАЛИЗ АЛГОРИТМОВ ВЫБОРА ПАКЕТА ДЛЯ ЗАПРОСА ПРИ МНОГОАБОНЕНТСКОЙ ПЕРЕДАЧЕ

Статья посвящена анализу различных алгоритмов выбора пакета для передачи в условиях многоабонентской передачи. Авторы рассматривают существующие подходы к выбору пакетов и дают формальное описание четырех основных – случайного, последовательного, неравномерно случайного и с приоритетами. Проведено моделирование работы указанных алгоритмов и оценена задержка получения пакета относительно времени его генерации для разных алгоритмов. Показано, что лучшие показатели имеют алгоритмы, содержащие как случайную составляющую, так и механизм приоритизации первоочередных пакетов.

The article is devoted to the problem of selection the package for the transmission – so called data scheduling for the multicast systems. According to the article, data scheduling plays important role in p2p multicasting, allowing utilization of available bandwidth. Authors provide information about existing methods and select four basic of them – random, sequential, with non-uniform distribution and with priorities. The simulation of specified algorithm is performed and the delay of package receiving is analyzed. It is shown, that algorithms, that combine the randomness with priority of the first packages, have best results.

Вступление

Увеличение объемов использования интернета поставило задачу передачи информации значительному количеству получателей. Решением этой проблемы может быть использование многоабонентской доставки информации.

Распространение потокового контента при помощи технологии Peer-To-Peer может быть жизнеспособным решением для обеспечения масштабированности и равномерного распределения нагрузки. Тем не менее нужно для решить целый ряд проблем. Как правило, при распространении потоковых данных, источник данных генерирует определенное число блоков фиксированного или переменного размера, которые должны быть доставлены получателям с фиксированной скоростью и в жесткие сроки. Не полученные вовремя пакеты могут привести к снижению качества, временной или даже полной приостановке воспроизведения [1].

Основной проблемой для р2р передачи есть временные ограничения в условиях динамичности, (так называемой churn), перегруженности сети и ограничениях на общее количество подключений. Необходимым условием тогда есть достижение эффективного использования пропускной ширины канала. Участники сети должны отдавать для нужд системы часть своей пропускной способности за счет передачи уже загруженных пакетов другим участникам [3].

Существует ряд протоколов многоабонентской передачи, которые используют технологию р2р, к ним относят например, Chainsaw, Meshcast, Coolstreaming.

Одним из важных элементов системы многоабонентской передачи алгоритм выбора пакета для передачи (пакета запроса для pull-систем и пакета для передачи для push-систем). Так в системе Chainsaw используется простейший алгоритм - случайный, Coolsteaming - с приоритетами. В [2] исследуются недостатки случайного алгоритма и исследуется разработка алгоритма с приоритетами для обеспечения непрерывности трансляции.

Случайный алгоритм хорошо обеспечивает динамичность за счет использования всех преимуществ топологии и выбора случайного пакета, но может привести к потере пакетов. Поэтому необходимо модифицировать алгоритм так, чтобы не уменьшить устойчивость алгоритма, но увеличить его шансы выбрать правильный пакет [4].

Другим крайним вариантом является выбор не случайного пакета, а первого имеющегося с наименьшим порядковым номером, что может привести к неравномерной нагрузке на узлы, так как порядок запроса пакетов будет одинаковым для всех узлов. Необходимо выбрать средний вариант - между выбором пакетов с равномерной вероятностью и исключения вероятности по выбору пакета вообще. Среди вариантов - выбор пакета с вероятностью, или например, в общем случае выбор пакета случайного, а в случае наличия пакетов, которые скоро достигнут своего таймаута - предоставление им преимущества. Задачей этой статьи исследовать как меняются характеристики получения пакетов при использовании различных алгоритмов выбора следующего пакета для передачи.

Алгоритмы выбора пакета

Следует отметить, что задача эффективного планирования передачи даже в условиях динамических связей между узлами и возможной неоднородностью узлов относится к классу NP-полных и не имеет оптимального решения, которое могло бы работать в условиях ограничения реального времени на принятие решения. Поэтому используются приближенные эвристические алгоритмы, которые работают с достаточной степенью эффективности для конкретных применений и условий.

Рассмотрены следующие алгоритмы выбора следующего пакета для запроса: равномерно случайный, последовательный, неравномерно случайный и с приоритетами.

Случайный выбор пакета запроса, можно формально описать так:

1. все пакеты из окна необходимости узла, имеющиеся у соседей со свободным подключением, расположить в порядке возрастания номеров

2. выбор случайного числа из множества целых чисел - возможных индексов пакетов в этом массиве (обычно это числа от 0 вплоть до длины массива не включительно).

3. выбор пакета из сформированного массива по случайно выбранному индексу.

Последовательный алгоритм:

1. все пакеты из окна необходимости узла, имеющиеся у соседей со свободным подключением, расположить в порядке возрастания номеров.

2. пакет запроса выбирается с наименьшим порядковым номером (то есть тот, который был сгенерирован раньше).

Одним из вариантов модификации выбора пакетов - использование не равномерного распределения для выбору индекса, а более сложного. В условиях наличия таймаута для получения пакета, следует выбирать те распределения, в которых вероятность выбора меньших индексов больше. Конечно, такой способ не гарантирует полного отсутствия потерянных пакетов, но значительно увеличивает вероятность своевременной доставки пакета. Как пример такого распределения было выбрано β распределение с параметрами $\alpha = 1$ $\beta = 3$. Замечательное свойство именно β распределения, что границы его значений четко фиксированы и становятся $[0, 1]$, в отличие от, например, экспоненциального распределения. Тогда выбор элемента из массива с этим распределением тривиальная задача.

1. Используя генератор псевдослучайных чисел, который выдает числа с β -распределением сгенерировать случайное число.

2. Умножить это число на общую длину массива и округлить до ближайшего целого.

3. Выбрать из упорядоченного массива пакетов по полученному индексу.

Другим вариантом является использование обычного случайного алгоритма с исключениями. Есть при обычных условиях, алгоритм работает в обычном режиме, но при определенных условиях наступают особые случаи. Каждый пакет может находиться в трех состояниях в зависимости от промежутка времени, который остался до окончания таймаута. Если этот промежуток меньше некоторого выбранного t_1 , то пакет считается потерянным и удаляется из окна необходимости. Этот случай описывает ситуацию, когда осталось слишком мало времени, чтобы вовремя доставить пакет, поэтому нет смысла его запрашивать и тратить время и пропускную способность. Если же оставшийся промежуток времени, находится в границах между t_2 и t_3 , то это означает, что пакет скоро будет считаться потерянным и он имеет приоритет на запрос. В противном же случае, пакет является обычным.

Тогда алгоритм выбора пакетов будет выглядеть следующим образом.

1. Назначить каждому пакету статус.

2. Очистить окно необходимых пакетов от пакетов со статусом 1.

3. Сформировать список пакетов, содержащихся в окнах доступности соседей со свободными подключениями.

4. Далее разделить пакеты на пакеты со статусом 2 и со статусом 3.

5. Если есть пакеты со статусом 2 - пакет выбирается из этой группы (случайно или по порядку). Пакеты из второй группы выбираются только в случае, когда нет пакетов в первой. Из этой группы пакеты выбираются исключительно случайно.

К достоинствам первых трех алгоритмов можно отнести простоту реализации и скорость работы, к достоинствам алгоритма с приоритетами – возможность адаптировать алгоритм под конкретное применение через манипуляцию параметрами t_1, t_2, t_3 .

Результаты моделирования

Для проверки эффективности предложенных модификаций была разработана программа на языке программирования Python по моделированию виртуальных сетей и сбора статистики состояния узлов. На вход программы подаются параметры системы: количество узлов, количество пакетов в трансляции, а также алгоритм, который следует моделировать. После модели-

рования в качестве результата возвращается пошаговый лог всех событий системы, статистика по загрузке входного и выходного каналов узлов, время получения каждого пакета различными узлами. Среди свойств сети критических для реализации в программе можно выделить следующие: ограничения на ширину канала узла (как на отдачу, так и на загрузку), ограничения на размер буфера (количества пакетов, которые могут одновременно храниться в узле).

Существует несколько вариантов метрик, которые можно использовать для оценки качества работы алгоритмов. Так как основное предназначение многоабонентской рассылки данных в настоящее время это трансляция медиа данных (видео/аудио поток), то критичным есть получение пакета в ограниченный промежуток времени. Для большинства трансляций начальная задержка начала трансляции не критична, хотя для конференций в режиме реального времени такая задержка может быть критична. В этой статье в качестве метрики была выбрана задержка относительно получения первого пакета каждым узлом, не относительно абсолютного начала трансляции.

На рисунках 1, 2, 3 и 4 представлены результаты моделирования трансляции 100 пакетов 10 узлам для разных алгоритмов выбора пакета. Для каждого узла показана задержка получения пакета относительно времени получения предыдущего пакета.

По оси абсцисс - порядковый номер пакета, по оси ординат - время его получения каждым узлом. Диагональная линия, показывает время получения пакета источником - то есть время его создания. Чем больше отклонение от этой линии в большую сторону, тем с большей задержкой был получен пакет.

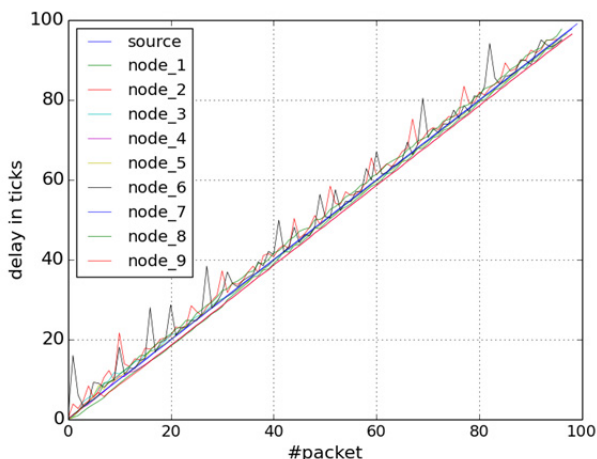


Рис. 1. Задержка получения пакетов для алгоритма случайного выбора

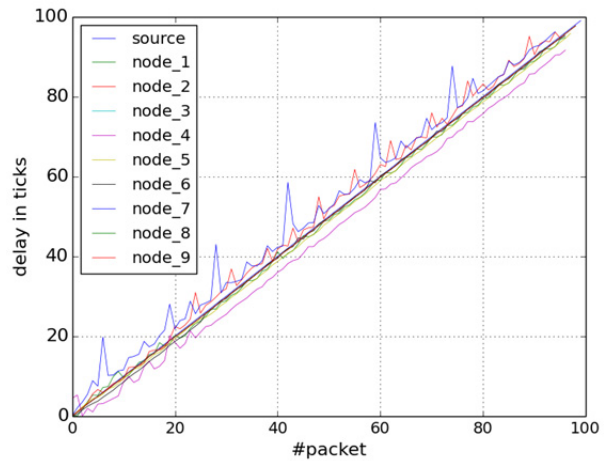


Рис. 2. Задержка получения пакетов для алгоритма последовательного выбора

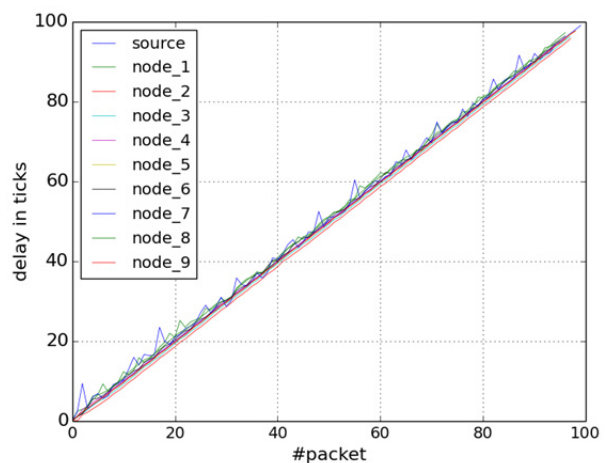


Рис. 3. Задержка получения пакетов для алгоритма с неравномерным распределением

Так как для потоковой передачи получения пакета в определенный промежуток времени является критичным, то пакеты, которые были получены с слишком большой задержкой считаются неактуальными и соответственно утраченными.

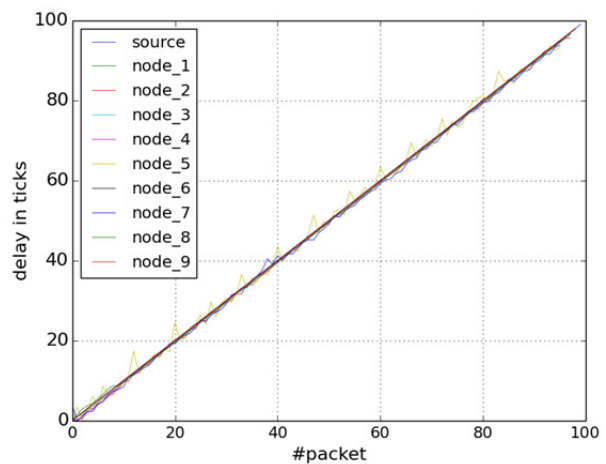


Рис. 4. Задержка получения пакетов для алгоритма с приоритетами

В таблице 1 показано начало трансляции для каждого пакета. Видно, что случайные алгоритмы имеют наиболее равномерные результаты, а худший результат у последовательного алгоритма за счет узлов, которые не смогли получить начальные пакеты в заданный промежуток времени.

Табл. 1. Задержка начала трансляции

	Rand	Seq	Beta	Priority
Mean	3.23	4.01	3.23	3.66
Std	1.15	3.1	1.23	2.55

Выводы

Если сравнить задержку пакетов для последовательного и случайного алгоритмов, то можно заметить, что результаты для случайного выбора более одинаковы для разных узлов, чем для последовательного выбора. В случае последовательного выбора существуют узлы, которые постоянно получают пакеты с большим запасом по таймауту, и узлы, которые хронически теряют пакеты. В случае же случайного выбора

такая закономерность также присутствует, но значительно меньше выражена. Если у случайного алгоритма, потери пакетов объясняются неоптимальным выбором, то последовательный алгоритм всегда выбирает оптимально, но он ограничен выбором из тех пакетов, которые имеются у его соседей.

Видно, что лучше показали себя более сложные алгоритмы: с неравномерным случайным выбором и приоритетами, которые сочетают случайную составляющую и учитывают необходимость доставить пакеты вовремя.

Выбирая между алгоритмом с beta распределением и с приоритетами, можно отметить, что у beta-алгоритма более быстрый старт трансляции, что может быть критичным для некоторых применений.

Показано, что выбор вероятностного алгоритма дает возможность использовать всю доступную входящую скорость, но увеличивает шансы на потерю пакета. Выбор алгоритма для выбора пакета должен зависеть от параметров системы, в которой планируется применение алгоритма.

Список литературы

1. Bagheri, Maryam, Ali Movaghar, and Ali Asghar Khodaparast. "Bandwidth Adapted Hierarchical Multicast Overlay." // Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on. IEEE – 2010 – P. 262-267.
2. Kwon, Oh Chan, and Hwangjun Song. Adaptive tree-based P2P video streaming multicast system under high peer-churn rate. // Journal of Visual Communication and Image Representation 24.3 – 2013. – P.203-216.
3. Meng, Xianfu, and Wei Wu. Live P2P Streaming System with High Playback Continuity. // 2014 International Conference on Computer, Communications and Information Technology (CCIT 2014) - Atlantis Press, 2014.
4. Afolabi, Richard O., Aresh Dadlani, and Kiseon Kim. "Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey." // Communications Surveys & Tutorials, IEEE 15.1 - 2013 - P.240-254.