Алгоритмы распространения блока в сети.

engi@sumus.team, a@sumus.team

12 февраля 2019 г.

Аннотация

В статье рассматриваются алгоритмы распространения блока информации по сети с фиксированной топологией и ограниченной пропускной способностью.

1 Обозначения

Введём обозначения:

- N — общее количество узлов
- M — количество “нисходящих” связей на один узел
- S — количество уровней, первичный узел не учитывается
- T — время передачи одного (целого) блока
- K — количество фрагментов блока
- R — время распространения блока в сети

Формулы, связывающие N, M, S. Точные:

$$N = \frac{M^S - 1}{M - 1} \tag{1}$$

$$S = \frac{\log(N(M-1)+1)}{\log M} \tag{2}$$

Упрощённые:

$$N \approx M^{S-1} \tag{3}$$

$$S \approx \frac{\log N}{\log M} + 1 \tag{4}$$

2 Описания алгоритмов

Общие условия распространения (фрагмента) блока в сети:

- от одного узла нельзя одновременно передавать более одного блока
- один узел не может одновременно принимать более одного блока
- узлы могут передавать блоки только по заданным связям
- узел может начать передачу блока только после полного приёма этого блока от другого узла
- узел **0** содержит блок в начальный момент времени

2.1 Алгоритм 00 (A00)

- каждый узел (кроме узлов последнего уровня) передаёт блок “своим” узлам следующего уровня в порядке нумерации

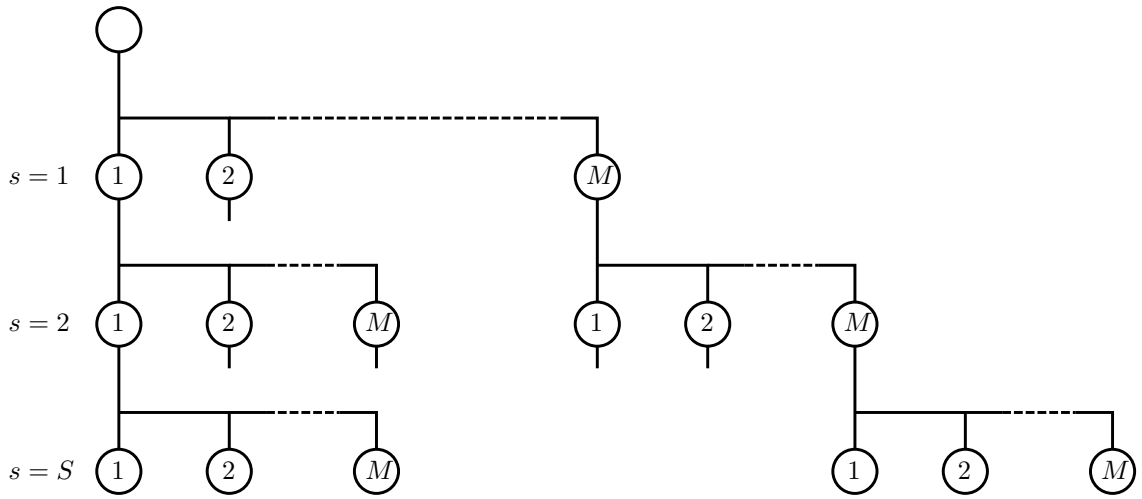

Рис. 1: Модель сети.

2.2 Алгоритм 01 (A01)

- каждый узел передаёт блок “своим” узлам следующего уровня
- количество узлов следующего уровня неограниченно

2.3 Алгоритм 10 (A10)

- блок разделён на фрагменты одинакового размера
- каждый узел (кроме узлов последнего уровня) передаёт фрагменты блока в порядке перечисления: узел, фрагмент

2.4 Алгоритм 11 (A11)

- блок разделён на фрагменты одинакового размера
- каждый узел (кроме узлов последнего уровня) передаёт фрагменты блока в порядке перечисления: узел, фрагмент

3 Расчёт

При заданной модели сети, узел получающий блок последним — узел последнего уровня, последний по порядку. Определив время получения блока этим узлом, получим время распространения блока в сети R.

3.1 Алгоритм 00 (A00)

Время получения блока первым узлом первого уровня T. Время получения блока вторым узлом первого уровня $2T$. Время получения блока последним узлом первого уровня $R(1) = MT$.

Каждый следующий уровень увеличивает время R распространения блока в сети на величину MT.

Время получения блока последним узлом уровня s будет $R(s) = MTs$.

Время получения блока последним узлом последнего уровня S будет $R(S) = MTS$.

$$R_{A00} = MTS \tag{5}$$

3.2 Алгоритм 10 (A10)

Главным отличием алгоритма **A10** от **A00** является возможность передавать фрагменты блока, не имея блок целиком.

Время передачи блока на первый уровень не отличается от времени по алгоритму **A00** и составляет $R(1) = MT$.

Рассмотрим ретрансляционный узел подробно. После получения фрагмента, узел имеет достаточно времени, чтобы передать этот фрагмент всем "своим" узлам следующего уровня до получения следующего фрагмента. Передача фрагмента на следующий уровень занимает $MT\frac{1}{K}$ времени. Каждый следующий уровень увеличивает время R распространения блока в сети на эту величину $MT\frac{1}{K}$.

Время получения блока последним узлом уровня s будет $R(s) = MT(\frac{s-1}{K} + 1)$.

Время получения блока последним узлом последнего уровня S будет $R(S) = MT(\frac{S-1}{K} + 1)$.

$$R_{A01} = MT(\frac{S-1}{K} + 1) \tag{6}$$

Если задать тривиальное значение $K = 1$, то формула получит вид (5).

Если величина $K \gg 1$ то распространение блока в сети может существенно приблизиться к времени передачи блока между соседними узлами.

4 Визуализация

5 Реализация

5.1 Алгоритм 10 (A10)

Реализация алгоритма **A10** ...

Блок разделяется на фрагменты f_i, из каждого фрагмента вычисляем хеш h_i. Полученные хеши подписываются узлами консенсуса. Таким образом получается содержательная часть **анонса** процесса, который распространяется по сети. Потом производится распространение фрагментов.

Узел вначале получает **анонс** процесса, проверяет его подписи и использует для получения фрагментов блока. Получив фрагмент блока f_i, узел вычисляет h_i и сверяет с хешем из **анонса**. При получении фрагмента нет проверки подписи. Проверка подписи происходит только при получении анонса, и у большинства узлов есть дополнительное время расчёта подписи до приёма первого фрагмента блока.