УДК 681.3.06

О.А. ОКСЮК

СОСТАВЛЕНИЕ ОПТИМАЛЬНОГО РАСПИСАНИЯ ПЕРЕДАЧИ ПАКЕТОВ ПОТОКОВОГО ВИДЕО В СЕТИ С ПОТЕРЯМИ ДАННЫХ

O.A. Oksyuk

OPTIMAL SCHEDULING FOR VIDEO STREAMING IN PACKET LOSS NETWORK

The article is dedicated to optimizing streamed video quality in loss network without back channel. New method of packets scheduling is suggested based on forward error correction. Suggested solution has enough computation speed to work in real-time on modern computers. At the same time it could be fitted to systems with wide range of computation speed.

Работа посвящена вопросу оптимизации качества потокового видео при передаче по сети с потерями данных и отсутствием обратного канала связи. Предлагается новый метод составления расписания передачи пакетов оптимизирующей качество видеоизображения, основанный на методике прямого исправления ошибок. Решение, предлагаемое в данной работе, имеет скорость вычислений достаточную для работы в режиме реального времени на современных компьютерах, в то же время оно может быть адаптировано для систем с широким спектром вычислительной мощности.

Введение. В данной работе рассматривается случай передачи потокового видеоизображения по сети с потерями данных и отсутствием обратного канала связи, т. е. в условиях, когда нет никакой возможности узнать, передался или потерялся пакет. Пропускная способность сети ограничена и может меняться, время доставки пакета так же может меняться. Составляется расписание передачи пакетов оптимизирующее качество видеоизображения, при этом используется методика прямого исправления ошибок, основанная на передаче избыточных пакетов.

В работе [1] предлагается решение рассматриваемой задачи для прогрессивного способа кодирования видеоизображения, в котором используется метод Рида-Соломона с постоянным коэффициентом избыточности для каждого уровня видеоизображения. В данной работе предлагается новый метод решения задачи, в котором коэффициент избыточности не зависит от уровня и может применяться, в том числе, и для видеоизображений закодированных не только прогрессивным способом.

Постановка задачи. В современных медиасистемах с цифровой передачей данных видеоизображение состоит из элементов данных, между которыми существует зависимость в виде ациклического графа [2]. Элементы данных группируются в блоки данных, которые независимы друг от друга, т. е. ни один элемент данных одного блока не зависит ни от одного элемента данных другого блока. Таким образом, каждый блок данных имеет отдельное дерево зависимостей между элементами данных и может быть использован без декодирования предыдущих блоков данных. В работе отдельно решается задача составления расписания передачи элементов данных для каждого блока данных.

Каждый элемент данных характеризуется двумя основными параметрами: размером b и искажением d, т. е. ухудшением качества получаемого видеоизображения при невозможности декодировать элемент данных. Для вычисления искажения используется формальная метрика отношения максимального сигнала к шуму [3]

$$Q = 10 \ \log_{10} \frac{(2^r - 1)^2}{\sigma^2} \ \text{дБ}, \quad \text{где } \sigma = \sum\nolimits_{x=1}^{A} \sum\nolimits_{y=1}^{B} (p(x,y) - p'(x,y))^2 . \qquad (1)$$

Здесь r – количество бит в пикселе; A, B – размеры видеоизображения в пикселях; $p(x, y)$, $p'(x, y)$ – цвет пикселя с координатами (x, y) в воспроизводимом и оригинальном изображении. Таким образом, в данной работе размер элемента данных измеряется в байтах, а искажение в децибелах.

В современных сетях передачи данных, в том числе и сети Интернет, данные передаются пакетами с ограниченным размером. Для моделирования сети с потерями данных будем считать, что вероятность потери пакета равна ε. Такое допущение справедливо в случае, когда длина интервала времени, когда допускается передача элемента данных, больше времени проигрывания блока данных, что характерно для неинтерактивных приложений с достаточно большим размером буфера. Элементы данных размера большего, чем максимально допустимый, разбиваются на пакеты и для каждого передаваемого пакета вероятность потери всегда равна ε.

Для уменьшения вероятности потери элемента данных будем использовать методику прямого исправления ошибок. Методика прямого исправления ошибок основана на том, что при пересылке k пакетов отправляется $(n - k)$ избыточных пакетов, при потере не более $(n - k)$ любых пакетов (не обязательно излишних) всегда можно восстановить все k исходных пакетов, а при потере $(n - k+1)$ пакета и более исходные k пакетов восстановить не возможно. Во время проведения эксперимента использовался метод Рида-Соломона [4]. Будем говорить, что k – количество основных пакетов, а $(n - k)$ – количество избыточных пакетов.

В общем случае оптимальное расписание передачи пакетов – это расписание, дающее минимальные потери качества за счет не передающихся или потерянных пакетов. Так, в работе [2] строится модель задачи в упрощенной формулировке, когда каждый пакет передается ноль либо один раз и в сети отсутствуют потери. Для случая потери части данных введем понятие эффективного искажения D_i элемента данных i как ожидаемое улучшение качества всего воспроизводимого видеоизображения в случае передачи элемента данных, разбитого на k_i пакетов и передачи $(n_i - k_i)$ избыточных пакетов. Эффективное искажение зависит от количества основных и избыточных пакетов самого элемента данных и элементов данных, от которых он зависит, или которые зависят от него. Задачу поиска оптимального множества передаваемых пакетов можно сформулировать как задачу нахождения целочисленного вектора $\vec{x}$ количества пакетов для передачи элементов данных, дающего оптимальное качество видеоизображения

$$\sum_{i=1}^{n} D_i(\vec{x}) \to \max \tag{2}$$

при ограничении суммарного размера передаваемых данных, что равносильно ограничению количества передаваемых пакетов

$$\sum_{i=1}^{n} x_i \le V. \tag{3}$$

Эта задача является задачей дискретного программирования.

Описание системы. В работе задача аппроксимируется задачей поиска максимального значения Лагранжиана

$$\sum_{i \in k} (D_i(\vec{x}) - \lambda x_i) \to \max. \tag{4}$$

Множитель λ динамически корректируется в зависимости от размера виртуального буфера на стороне отправителя. Так, при увеличении размера виртуального буфера (обычно вызванного уменьшением пропускной способности сети либо увеличением битовой скорости видеоизображения) множитель λ увеличивается и соответственно уменьшается количество передаваемых пакетов, а при уменьшении размера виртуального буфера (обычно вызванного увеличением пропускной способности сети либо уменьшением битовой скорости видеоизображения) множитель λ уменьшается и соответственно возрастает количество передаваемых пакетов.

Решение задачи строится интерактивно. На первом шаге в расписание добавляются все основные и возможно несколько избыточных пакетов корня дерева зависимостей между элементами данных. При этом количество избыточных пакетов корня дерева вычисляется так же, как и для всех остальных элементов данных на первом этапе итерации, что будет показано далее. Заметим, что в случае, если значение функции M отрицательно для любого значения x, в расписание, тем не менее, добавляются только основные пакеты корня дерева. Каждый следующий шаг проходит в три этапа. На первом этапе в расписание добавляем пакеты одного или нескольких элементов данных, которые еще ни разу не передавались, такие, что все элементы данных, от которых они зависят, уже добавлены в расписание передачи пакетов на предыдущих шагах. Если на первом этапе в расписание не добавлено ни одного пакета, то переходим ко второму этапу, в противном случае второй этап пропускаем. На втором этапе методом поиска в глубину с ограниченным шагом ищем множество элементов данных, добавляемых в расписание передачи пакетов. Затем, на третьем этапе, для ноля, одного или нескольких элементов данных из числа тех, которые уже передавались, увеличиваем количество избыточных пакетов. Поиск заканчивается тогда, когда на первом и втором этапе следующего шага итерации не удается выбрать ни одного элемента данных.

Рассмотрим более подробно первый этап шага итерации. На этом этапе рассматривается множество элементов данных *A*, каждый из которых удовлетворяет двум условиям:

- пакет еще ни разу не передавался;

- все пакеты, от которых зависит данный пакет, уже передавались хотя бы один раз.

Основная идея первого этапа состоит в том, что из множества *A* выбираются те элементы данных, для которых Лагранжиан *i*-го элемента данных M_i положителен:

$$M_i(\vec{x}) = D_i(\vec{x}) - \lambda x_i. \tag{5}$$

В этом случае при передаче пакетов таких элементов данных функция (4) увеличится и, таким образом, решение улучшится. Легко показать, что эффективное искажение для элементов данных, удовлетворяющих условиям множества *A* равно

$$D_i(\vec{x}) = d_i p_{x_i}^{k_i} \prod_{\forall j: j \to i} p_{x_j}^{k_j}, \tag{6}$$

где $p_{x_i}^{k_i}$ – вероятность восстановления *i*-го элемента данных, состоящего из k_i основных и x_i-k_i избыточных пакетов, которая выражается формулой

$$p_n^k = \sum_{i=k}^{n} C_n^k (1-\varepsilon)^i \varepsilon^{n-i}. \tag{7}$$

Обозначим

$$F_{x_i} = \frac{\sum_{l=k_i}^{x_i} C_l^{k_j} (1-\varepsilon)^l \varepsilon^{x_i - l}}{x_i}. \tag{8}$$

Тогда, подставив (7) в (6), а (6) в (5) получим, что условие максимизации функции (7) при условии, что оно положительно, достигается при x_i максимизирующим функцию (10). Обозначим такое x_i символом x_i’, а соответствующее значение *F* символом F_i’. В этом случае значение функции *M* будет равно

$$M'_i(\vec{x}) = (d_i F'_i \prod_{\forall j: j \to i} p_{x_j}^{k_j} - \lambda)\, x_i^{'}. \tag{9}$$

Рассчитаем значения x_i’, M'_i ,для каждого пакета из множества *A*. Выберем подмножество элементов данных *A** множества *A*, для которых M'_i положительно.

Попытаемся улучшить Лагранжиан для множества *A**. Для этого определим приращение Лагранжиана для *i*-го элемента данных как разницу между функцией Лагранжиана в случае, когда *i*-й элемент данных передается на один раз больше и исходным значением функции *M*:

$$\Delta M_i(x_i) = M(x_i + 1) - M(x_i). \tag{10}$$

Для каждого пакета из множества A^*, последовательно увеличивая x_i, начиная с x_i', находим x_i'', для которого справедливы два неравенства:

$$\Delta M_i(x_i''-1) > 0, \quad \Delta M_i(x_i'') \le 0. \tag{11}$$

Если подмножество A^* непустое, то для каждого элемента данных i из множества A^* последовательно добавляем в расписание x_i'' пакетов. На этом первый этап шага итерации заканчивается.

Если подмножество A^* пустое, то переходим ко второму этапу шага итерации, в противном случае второй этап пропускаем и переходим сразу к третьему этапу. Рассмотрим более подробно второй этап шага итерации. На данном этапе производится поиск в глубину с ограниченной глубиной [5] множества B элементов данных, обладающего следующими свойствами:

- все элементы данных множества B еще не добавлены в расписание передачи пакетов;

- элементы множества B пронумерованы так, что никакой элемент данных не зависит ни от одного элемента данных следующего за ним;

- количество элементов данных множества B не больше глубины поиска N;

- для любого элемента данных множества B справедливо утверждение: “Все элементы данных, от которых зависит элемент данных множества B либо принадлежат множеству B, либо уже добавлены в расписание передачи пакетов”;

- для множества B выполняется неравенство

$$\sum_{i \in B} D'_i > 0. \tag{12}$$

Поиск останавливается после нахождения первого множества, удовлетворяющего этим условиям. В этом случае для каждого элемента данных i из множества B последовательно добавляем в расписание x_i' пакетов и считаем, что второй этап шага итерации закончен. Заметим, что глубина поиска в глубину зависит от вычислительной мощности компьютера, таким образом, поиск дает более точный результат на более мощных компьютерах.

В случае, если на первом или на втором этапе в расписание добавлено непустое множество пакетов, то переходим к третьему этапу. На третьем этапе рассматривается множество C элементов данных, каждый из которых удовлетворяет условиям:

- пакеты элемента данных уже добавлены в расписание;

- по крайней мере, для одного из элементов данных, от которого зависит данный элемент, пакеты уже добавлены в расписание.

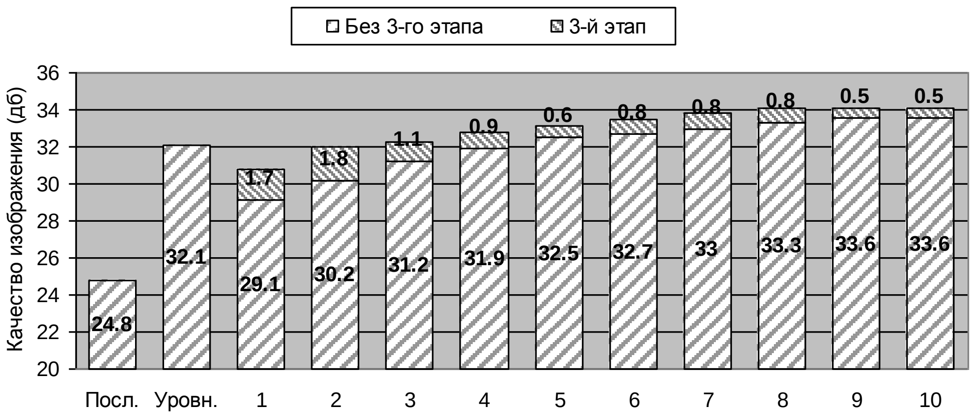
Основная идея третьего этапа состоит в последовательном вычислении приращения Лагранжиана для каждого элемента данных из множества C и числа пакетов x_i, x_i+1 и т. д. до тех пор, пока не найдено количество пакетов, удовлетворяющее условиям (11). Если это число x_i'' больше x_i, , то в расписание добавляется $x_i'' - x_i$ избыточных пакетов для i-го элемента данных. При этом для упрощения клиентской реализации, они добавляются так, чтобы все пакеты одного элемента данных передавались друг за другом. Приращение Лагранжиана для i-го элемента данных вычисляется по формуле:

$$\Delta D_i(\vec{x}) = \Big(\frac{p_{x_i}^{k_i}+1}{p_{x_i}^{k_i}} - 1\Big)(D_i^s + \sum_{\forall j: i\to q} D_j^s) - \lambda. \tag{13}$$

Заметим, что формула (3) в отличие от (8) и (11) учитывает элементы данных, зависящие от данного.

Экспериментальные результаты. Для исследования использовалось видеоизображение “Строитель”, закодированное с помощью алгоритма MPEG-4 – прогрессивный с 4-мя уровнями, частотой 30 кадров в сек., всего 3000 кадров, средним качеством видеоизображения около 36 децибел. Эксперимент проводился на эмулированной сети с потерями в ней 5 %, пропускной способностью 150 кб/с и максимально допустимым размером пакета 1500 байт.

На рисунке показана зависимость качества видеоизображения для разных алгоритмов составления расписания. Кроме предложенного алгоритма с разной глубиной поиска (1–10), вычисления проводятся для еще двух алгоритмов: алгоритма последовательной передачи всех элементов блока данных до тех пор, пока буфер на клиенте не станет пустым, и алгоритма прямого исправления ошибок методом Рида-Соломона с постоянным коэффициентом избыточности (отношение количества избыточных пакетов к общему количеству пакетов) для каждого уровня [1].



Обозначения: посл. – последовательная передача всех элементов блока данных до тех пор, пока буфер на клиенте не станет пустым; уровн. – прямое исправление ошибок методом Рида-Соломона с постоянным коэффициентом избыточности (отношение количества избыточных пакетов к общему количеству пакетов) для каждого уровня.

РИСУНОК. Зависимость качества получаемого видеоизображения от алгоритма

Как видно из рисунка, качество видеоизображения при использовании предложенного метода даже без поиска в глубину (глубина поиска 1) на 6 дБ выше, чем при передаче кадров подряд, но на 1.3 дБ ниже, чем при использовании постоянного коэффициента избыточности для каждого уровня. С использованием поиска в глубину уже с глубиной поиска 3 предложенный метод дает незначительно лучший результат, чем использование постоянного коэффициента избыточности для каждого уровня. Максимально возможное качество достигается при поиске в глубину с глубиной поиска 8 и перестает расти при дальнейшем увеличении глубины поиска, при этом качество видеоизображения уже на 1.9 дБ выше, чем при использовании постоянного коэффициента избыточности для каждого уровня.

Показано, что на современных компьютерах для составления расписания в режиме реального времени оптимально использовать поиск в глубину с шагом 3–5 и с использованием 3-го этапа предложенного метода.

Заключение. В настоящей работе поставлена и формализована задача оптимизации расписания передачи пакетов потокового видео по сети с потерями данных без обратного канала связи. Построена аналитическая модель задачи, являющейся задачей дискретного программирования. Предложен новый метод решения задачи, основанный на поиске по дереву в глубину с ограниченной глубиной. Следует отметить, что разработанный метод имеет скорость вычислений, достаточную для работы в режиме реального времени на современных компьютерах, в то же время он может быть адаптирован для систем с широким спектром вычислительной мощности.

1. *Tan W., Zakhor A.* Video multicast using layered FEC and scalable compression // IEEE Transactions on Circuits and Systems for Video Technology – Piscataway, N.J., USA: IEEE, March 2001. – Vol. 11. – N 3. – P. 373 – 386.
2. *Оксюк О.А., Петрухин В.А*. Оптимизация передачи потокового видео по сети с пропускной способностью, близкой к скорости битового потока видеоизображения // Компьютерная математика. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2006. – № 1. – С. 152 – 159.
3. *Prashant J. Shenoy, Harrick M. Vin.* Efficient Support for Interactive Operations in Multiresolution Video Servers // Multimedia Systems – Berlin, Germany: Springer Verlag, May 1999. – Vol. 7. – N 3. – P. 241 – 253.
4. *Mandelbaum D.* An adaptive-feedback coding scheme using incremental redundancy // IEEE Transactions on Information Theory – Piscataway, N.J., USA: IEEE, May 1974. – Vol. 20. – N 3. – P. 388 – 397.
5. *Вирт Н.* Алгоритмы и структуры данных. – М.: Мир, 1989. – 212 с.

Получено 26.02.2007