

# Прогрессивная передача аудио через Интернет

Александр Симаков  
Сыктывкарский государственный университет

1 января 2005 г.

### **Аннотация**

Всем кому когда-либо доводилось слушать музыку или радио через Интернет знакома ситуация при которой из-за падения скорости передачи опустошается буфер и воспроизведение начинает «заикаться». Это происходит даже не смотря на то, что пользователю как правило предлагается на выбор несколько потоков, закодированных с разными битрейтами. В данной статье обсуждается система для прогрессивной передачи аудио через Интернет, которая будет лишена указанного недостатка.

# 1 Введение

Всем кому когда-либо доводилось слушать музыку или радио через Интернет знакома ситуация при которой из-за падения скорости передачи опустошается буфер и воспроизведение начинает «заикаться». Это происходит даже не смотря на то, что пользователю как правило предлагается на выбор несколько потоков, закодированных с разными битрейтами.

Дело в том, что мы не можем гарантировать, что определенный пакет, пройдя через десяток маршрутизаторов дойдет до нас в нужное время. А для воспроизведения фрейма в классических схемах (mp3, ogg и т.д.) необходимо получить его целиком. В данной статье обсуждается система для прогрессивной передачи аудио через Интернет, которая будет лишена указанного недостатка.

Прогрессивность при этом нужно понимать как свойство закодированного фрейма, при котором первые биты будут нести наиболее важную для восприятия информацию, а последующие — лишь незначительные, уточняющие детали. В таком случае, первых битов будет достаточно для воспроизведения всего фрейма целиком, правда в немного худшем качестве.

Другими словами, при падении скорости передачи вместо остановки воспроизведения будет ухудшаться его качество. Если затем скорость повысится, то и качество воспроизведения возрастет. Иначе говоря, такая система будет адаптироваться к пропускной способности канала в реальном времени.

Очевидно, что поскольку прогрессивно закодированный фрейм можно урезать до любого размера, мы с легкостью сможем управлять общим битрейтом с точностью до бита! При этом мы не будем привязаны к «стандартным» значениям: 128 Kbps, 64 Kbps и т.д.

И наконец, подобный подход позволяет хранить на сервере всего один закодированный файл, вместо нескольких для разных битрейтов. Действительно, каждый клиент получит и воспроизведет его с тем качеством, насколько ему позволит канал связи.

Программная реализация описанного в данной статье аудио кодера свободно доступна с сайта автора <http://www.entropyware.info> и распространяется на условиях лицензии GNU GPL.

## 2 Цифровой звук

Цифровой звук можно охарактеризовать тремя главными параметрами: частотой дискретизации, количеством бит на сэмпл (или иначе отсчет

или дискрет) и количеством каналов. От величины этих параметров зависит качество звучания и размер файла. К примеру, секунда музыки на аудио-CD занимает  $16 \text{ бит} \cdot 2 \text{ канала} \cdot 44100 \text{ Гц} = 176400 \text{ байт}$ . Отсюда и вытекает необходимость в эффективных методах для сжатия звуковой информации, которая по природе своей весьма избыточна.

Как правило, вначале исходный файл делится на небольшие фрагменты — фреймы, которые затем и подвергаются кодированию. Типичная протяженность фрейма по времени составляет порядка 10–20 миллисекунд.

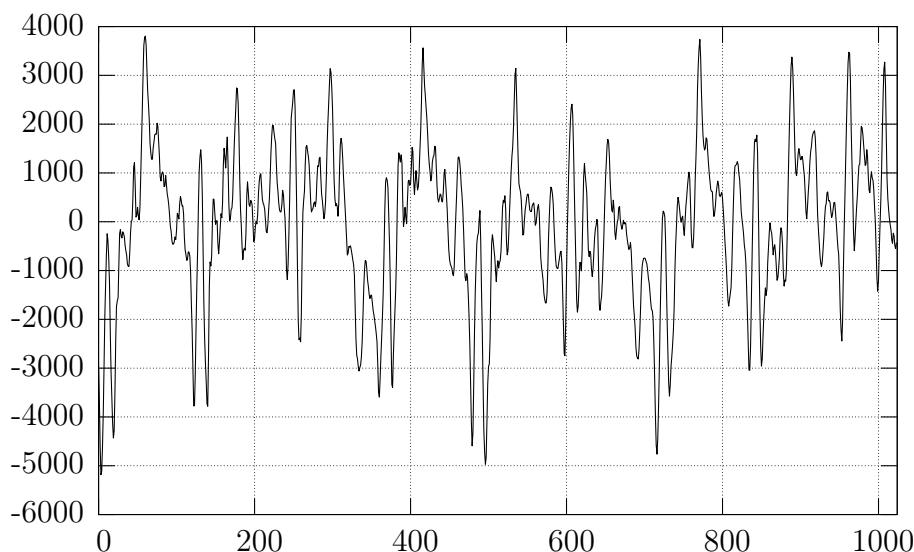


Рис. 1: Исходный сигнал

На рис. 1 показан фрейм размера 1024 сэмпла. На частоте 44100 Гц его протяженность по времени составляет  $1024/44100 \cdot 1000 \approx 23.22$  миллисекунд. Каждый сэмпл занимает по 16 бит, что обеспечивает допустимый диапазон значений  $[-32768 \dots 32767]$ . Именно на этом примере и будут производиться все дальнейшие эксперименты.

### 3 Вейвлетное преобразование

Вейвлетное преобразование служит для выделения высоко- и низкочастотной составляющей сигнала (далее НЧ и ВЧ). Уже давно установлено, что НЧ составляющая намного важнее для человеческого воспри-

ятия чем ВЧ. Отсюда и возникает идея выделения НЧ и ВЧ составляющих с последующим приоритетным подавлением последней. Для этой цели в данной работе было использовано известное вейвлетное преобразование Добеши 9/7 [1, 2]. На рис. 2 изображен результат его применения к исходному сигналу.

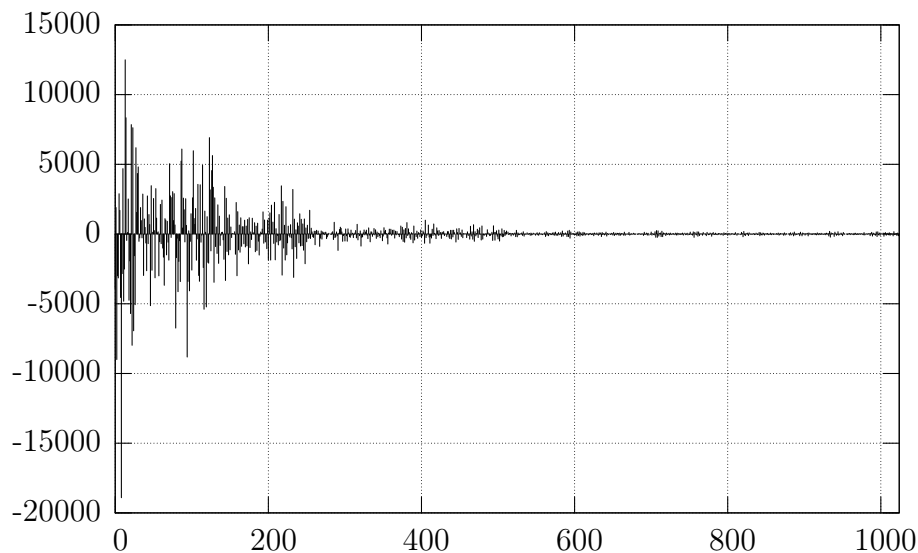


Рис. 2: Вейвлетное преобразование сигнала

Из рисунка видно, что коэффициенты вейвлетного разложения отвечающие за НЧ составляющую (ближе к началу координат) сигнала доминируют над ВЧ коэффициентами, а бóльшие по абсолютной величине коэффициенты несут больше информации об исходном сигнале.

## 4 Прогрессивное кодирование

После того как получены коэффициенты вейвлетного разложения необходимо их прогрессивно закодировать. В качестве алгоритма кодирования в этой работе был использован известный алгоритм SPIHT (Set Partitioning in Hierarchical Trees) [3, 4]. Для большей эффективности его комбинируют с арифметическим кодированием [5]. Не смотря на то, что изначально SPIHT был разработан для кодирования изображений, его с таким же успехом можно применить и к аудио сигналам.

Стоит особо подчеркнуть, что основная задача алгоритма SPIHT не заключается в непосредственном сжатии данных. Используя особенно-

сти структуры вейвлетных коэффициентов, SPIHT переупорядочивает их биты. При этом первые биты будут нести наиболее важную информацию (НЧ), в то время как последующие — лишь незначительные, уточняющие детали (ВЧ). Такое упорядочение данных часто называют *прогрессивным*.

Таким образом, чем больше битов закодированного фрейма получает декодер, тем точнее восстановленный сигнал будет повторять оригинальный. Для примера приведем серию восстановленных сигналов и сравним их с оригиналом.

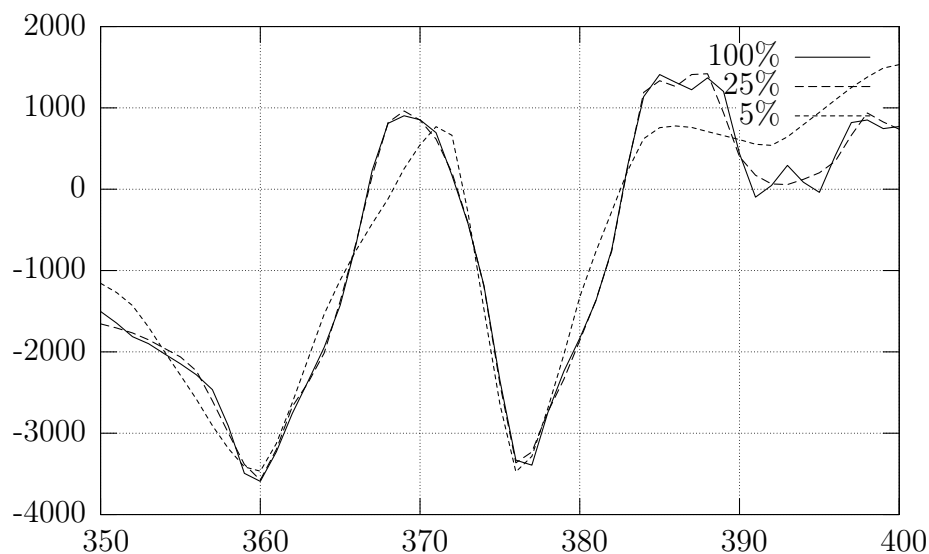


Рис. 3: Восстановленный сигнал

На рис. 3 хорошо видно как при увеличении количества получаемых декодером битов растет качество восстановленного сигнала и на определенном уровне он практически неотличим от оригинала.

## 5 Передача аудио через Интернет

Для того чтобы извлечь все преимущества, полученные при прогрессивном кодировании аудио данных, необходимо разработать специальный протокол прикладного уровня для взаимодействия клиента и сервера по сети. В качестве протокола транспортного уровня в данной ситуации наиболее подходящим является протокол TCP.

Ключевой момент клиент–серверного взаимодействия заключается в том, что клиент имеет возможность указать, а также изменять во время работы максимальный размер закодированного фрейма. Клиент выбирает это значение таким образом, чтобы минимизировать, а в лучшем случае вообще исключить возможные обрывы воспроизведения.

Увеличивая и уменьшая размер фрейма клиент может адаптироваться к изменяющейся скорости передачи: когда скорость падает — размер фрейма необходимо уменьшить, когда скорость возрастает — размер фрейма можно увеличить. Сервер, со своей стороны, должен присылать от каждого фрейма не больше байтов чем попросил клиент.

## 6 Заключение

Аудио кодер, описанный в этой статье, не использует психоакустическое моделирование, сложные фильтры для обработки сигналов и другие полезные оптимизации. Он намеренно упрощен и является скорее демонстрацией концепции: прогрессивной передачи аудио по низкоскоростным каналам связи. Не смотря на это, его полные исходные тексты доступны для скачивания, изучения, экспериментов и улучшения.

## Список литературы

- [1] I. Daubechies, W. Sweldens. Factoring Wavelet Transforms Into Lifting Steps // J. Fourier Anal. Appl. 1998. V. 4. N 3. P. 245-267.
- [2] A. Cohen, I. Daubechies, J. Feauveau. Biorthogonal Bases of Compactly Supported Wavelets // Comm. on Pure and Appl. Mathematics. 1992. V. 45. N 5. P. 485-560.
- [3] Amir Said, William A. Pearlman. A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees // IEEE Trans. on Circuits and Systems for Video Technology. 1996. V. 6. P. 243-250.
- [4] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients // IEEE Transactions on Signal Processing. 1993. V. 41. N 12. P. 3445-3462.
- [5] I. H. Witten, R. M. Neal, J. H. Cleary. Arithmetic coding for data compression // CACM. 1987. V. 30. N 6. P. 520-540.