

Абстрактный сетевой уровень в стандарте H.264/AVC

5 января 2023 г. 1:31

В беспроводном канале всегда происходят потери пакетов. Допустим, потерялся кадр, т.е. не был доставлен декодеру. Тогда появится искажение при декодировании. Причем ошибка будет распространяться от кадра к кадру, искажение будет накапливаться до передачи следующего ключевого кадра (т.к. кадры зависят друг от друга, от потерянного искажаются все следующие).

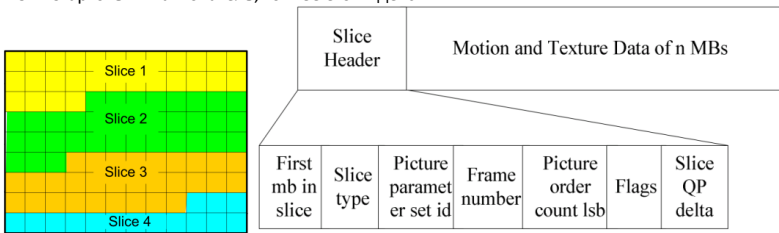
В стандарте H.264 при передаче видео по каналам с потерей пакетов:

1. Видеокадр разбивается на макроблоки (16x16) перед кодированием - это такие единицы, для которых принимаются всевозможные решения, т.е. какой будет интра-интр-блок; если это интра, то какой режим предсказания; если интр, то как разбить его и какие векторы движения найти и т.д.
2. Несколько макроблоков составляют один сегмент (slice)
3. Один кадр может состоять из нескольких сегментов
4. Сегменты могут быть типа I, P, B

Slice состоит из заголовка и данных (векторы движения, закодированные разностные коэффициенты с помощью CAVLC или CABAC)

В хедере:

1. Координаты первого макроблока на кадре
2. Тип слайса (I, P, B)
3. Номер кадра
4. Флаги
5. Стартовый квантователь, точнее его индекс



Абстрактный сетевой уровень - network abstract layer - NAL

Network abstract layer (NAL) это интерфейс между уровнем битового потока и сетевым уровнем.

NAL кодер помещает битовый поток сегмента в NAL пакет, который может быть передан в пакетной сети.

NAL может быть принят и декодирован независимо от остальных NAL.

NAL содержит в себе заголовок и payload - основную часть

В payload будет slice

А в хедер входят параметры более высокого характера



Sequence parameter set	Picture parameter set	I slice	Picture delimiter	P slice	P slice
------------------------	-----------------------	---------	-------------------	---------	---------

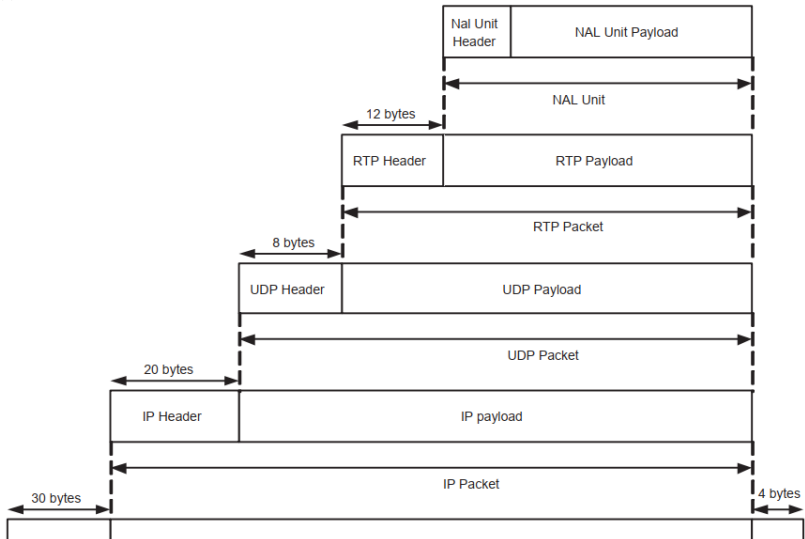
Чтобы всё дошло до физической сети:

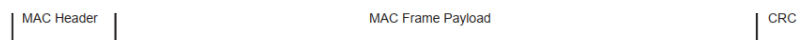
Если мы сохраняем в файл, то NAL вставить разделители между пакетами типа 00 00 00 01, чтобы можно было быстро между ними перемещаться

Либо NAL-пакеты помещаются в RTP-пакет (RTP - надстройка над UDP, т.е. по факту UDP с заголовком)

Т.е. что происходит:

Образовался NAL-пакет, у него есть хедер и payload, каждый следующий сетевой уровень добавляет свой заголовок

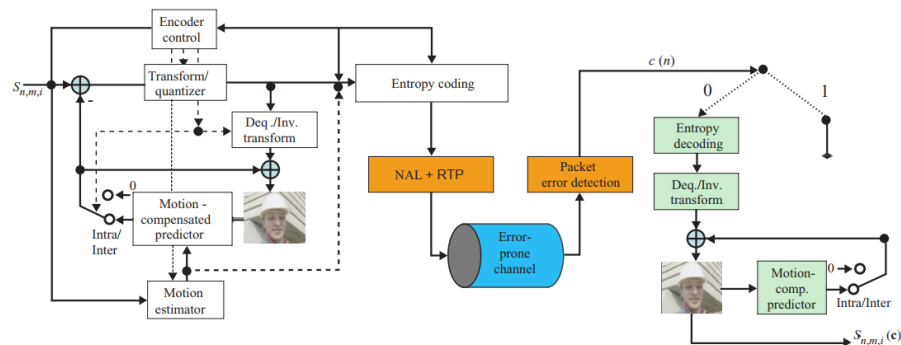




(CRC - контрольная сумма - показывает, побился ли пакет. битый пакет не доходит до уровня приложений)

Когда пакет наоборот принимается из сети, происходит обратное: каждый сетевой уровень избавляется от хедера и передает дальше.

В стандартах нет такого понятия, как что делать, если произошли потери данных. Т.е. декодер 264 стандарта работает в допущении, что все нужные данные дошли.



1. После энтропийного кодирования есть пакетизация "NAL+RTP" (NAL помещается в RTP пакет)
2. Далее происходит передача с ошибками по каналу
3. Поиск ошибок в полученном пакете (можно включить режим получения или отброса битых пакетов)

При возникновении ошибок декодер должен:

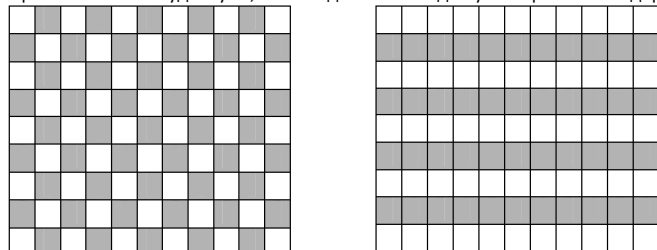
- Обнаружить ошибку;
- Ресинхронизировать декодирование;
- Компенсировать искажения, которые вызваны потерями пакетов.

Обнаружение ошибок – может быть выполнено с учетом знания о синтаксисе битового потока. Например, когда обнаруживаются синтаксические элементы за пределами разрешенного диапазона, то декодер должен **ресинхронизировать** декодирование, т.е., начать декодировать со следующего стартового кода.

Пакеты с обнаруженными ошибками не декодируются (skipped) и для соответствующей области кадра выполняется процедура компенсации искажений (error concealment), которая пытается минимизировать визуальные искажения.

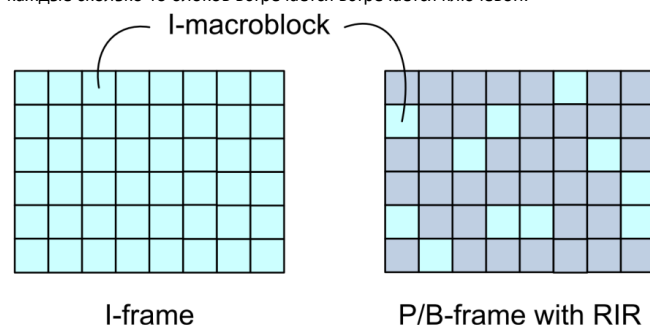
Как улучшить защиту?

В стандарте 264 происходит нарезка на сегменты "шахматной доской". И кодирование происходит так, что все "белые" блоки в одном сегменте, все "черные" - в другом. Все сегменты идут строго по строчкам. Сжатие будет хуже, т.к. соседи блоков недоступны при таком кодировании.

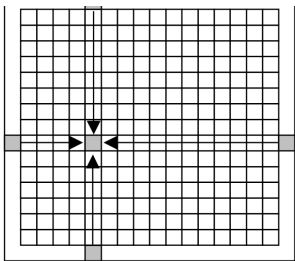


Другой вариант - передавать избыточные сегменты, т.е. можно кодировать каждый сегмент дважды: со сжатием поменьше и побольше. Если декодер получил обе версии, берет менее сжатую и декодирует, иначе - вторую.

Третий способ - периодическая вставка ключевых (интра) макроблоков. Помним, что на каждом ключевом макроблоке накопленная ошибка сбрасывается. Имеем I-frame с I-макроблоками, случайным образом можем вставлять в P или B кадр таким образом, что, например, в среднем за каждые сколько-то блоков встречается ключевой.



Если, например, соседние блоки приняты, а текущий не пришел, как исправить? Можно интерполировать недостающие пиксели на основе успешно принятых пограничных пикселей.



Или можно посмотреть у соседних блоков векторы движения и предсказать вектор движения утраченной области по ним.

