



Формирование кадра. Byte stuffing и Bit stuffing

Автор: Г.И. Осипова
ст. группы ВТм-112

Формирование кадра

На физическом уровне для разделения потока битов, соответствующих разным кадрам, могут использоваться различные способы:

- указание в заголовке кадра его длины и подсчет количества символов в процессе приема потока данных (основной недостаток - неустойчивость к помехам);
- использование в качестве границы кадров запрещенных сигналов физического уровня;
- использование в качестве границы кадров специальных стартовых и стоповых символов (байтов);
- использование в качестве границы кадров специальных последовательностей битов.

Подсчет количества символов

Использует поле в заголовке для указания количества символов в кадре.

Когда уровень передачи данных на принимающем компьютере видит это поле, он узнает, сколько символов последует, и таким образом определяет, где находится конец кадра.

На рисунке ниже - четыре кадра размером 5, 5, 8 и 8 символов соответственно.



Подсчет количества символов

Недостаток - при передаче может быть искажен сам счетчик. Например, если размер второго кадра из числа 5 станет из-за ошибки в канале числом 7, как показано на рисунке ниже, то принимающая машина потеряет синхронизацию и не сможет обнаружить начало следующего кадра.



По этому метод подсчета символов теперь практически не применяется.

Использование флагов

Второй метод формирования кадров решает проблему восстановления синхронизации после сбоя при помощи маркировки начала и конца каждого кадра специальными байтами.

Большинство протоколов использует в качестве стартового и стопового байта один и тот же байта, называемый **флаговым**.

Это показано на рисунке ниже как FLAG.

FLAG	Заголовок	Поле данных	Концевик	FLAG
------	-----------	-------------	----------	------

Использование флагов.

Проблемы метода

В передаваемых данных может встретиться последовательность, используемая в качестве флагового байта.

Одним из способов решения проблемы является добавление специального ескаре-символа (знака переключения кода, ESC) непосредственно перед случайно совпавшим с флаговым байтом внутри кадра.

Уровень передачи данных получателя вначале убирает эти ескаре-символы, затем передает кадр на сетевой уровень.

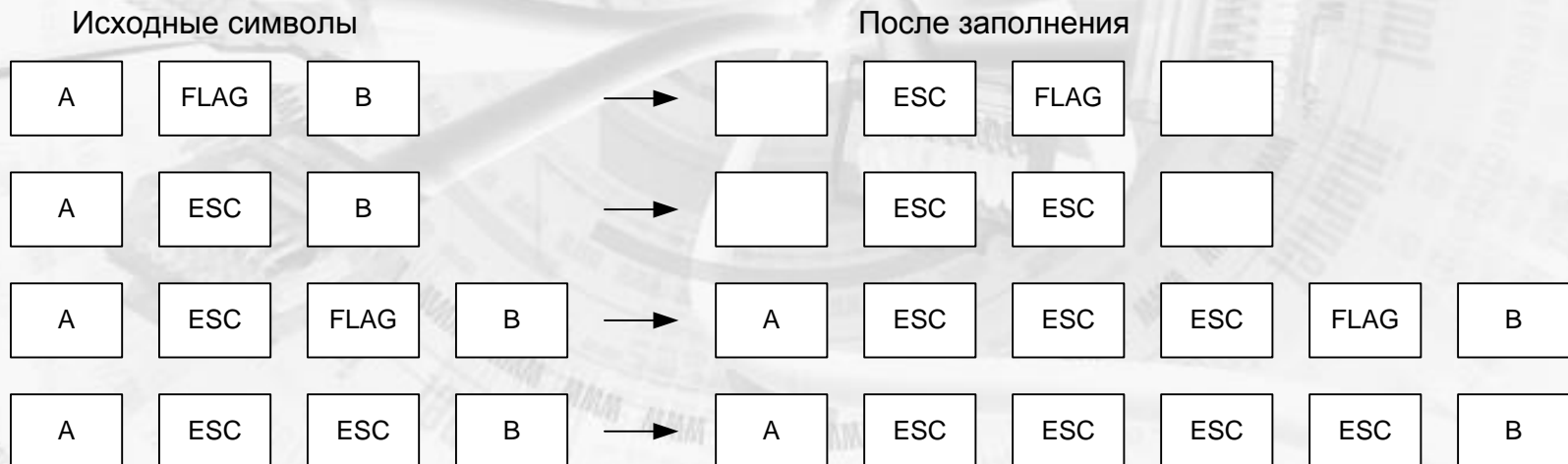
Такой метод называется **символьным заполнением**.

Использование флагов.

Проблемы метода

Если символ ESC случайно окажется среди прочих данных, нужно вставить перед этим фиктивным ескаре-символом настоящий. Тогда любой одиночный ESC будет частью ескаре-последовательности, а двойной будет указывать на то, что служебный байт случайно оказался в потоке данных.

Некоторые примеры показаны на рисунке ниже.



Использование флагов

Главный недостаток этого метода заключается в том, что он тесно связан с 8-битными символами. Между тем не во всех кодировках один символ соответствует 8 битам. Например, UNICODE использует 16-битное кодирование.

Byte stuffing

При использовании в качестве границы кадров специальных стартовых и стоповых символов (байтов) реализуется **«байт-стаффинг»** - техника вставки, а точнее замены байтов, совпадающих с граничными в тексте кадра, на определённые последовательности других символов.

Byte stuffing. Реализация

Реализация:

- начало и конец каждого кадра отмечается одной и той же 8-битовой последовательностью - **01111110**, называемой флагом;
- поток сканируется приемником на побитовой основе для обнаружения стартового флага, а затем во время приема для обнаружения стопового флага;
- поэтому длина кадра в этом случае не обязательно должна быть кратна 8 бит.

Bit stuffing

При этом необходимо, чтобы флаг не присутствовал в поле данных кадра.

Иначе может возникнуть такая проблема - последовательность битов флага (01111110) может встретиться внутри кадра и будет ошибочно воспринята аппаратурой передачи данных как обрамление, т.е. как начало следующего кадра.

Для исключения этого используется процедура обеспечения прозрачности канала - **бит-стаффинг**.

Bit stuffing

Бит-стаффинг (bit stuffing - вставка битов) - техника вставки и стирания битов, используемая в высокоскоростных цифровых каналах связи с большим числом линий связи, не имеющих взаимной синхронизации, а также средство синхронизации в протоколах управления каналом связи типа HDLC.

Bit stuffing. Реализация

На передающем узле после пяти подряд следующих единиц внутри кадра принудительно вставляется 0, который автоматически изымается на приемном узле.

Таким образом, исключается возможность появления внутри кадра последовательности битов 01111110, используемой для разделения кадров.

Bit stuffing. Реализация

Допустим, что необходимо передать кадр, в котором встречаются (см. пункт 1 ниже):

- шесть подряд идущих единиц (которые находятся между двумя нулевыми битами и могут быть восприняты как граница кадра);
- ровно пять единиц;
- более шести единиц.

1000111111011001110101101111101100001111111

Bit stuffing. Реализация

- 1) 1000111110110011101011011110110000111111
- 2) 10001111101011001110101101111001100001111011

Реализация бит-стаффинга - после пяти любых подряд идущих единиц принудительно будут вставлены нулевые биты (см. последовательность 2).

При этом нули вставляются не зависимо от того, совпадает или не совпадает внутрикадровая битовая последовательность с флагом 0111110, используемым для разделения кадров.

В результате такой процедуры по каналу связи будет передана последовательность битов:

0111110100011111010110011101011011110011000011110110111110

Если после пяти подряд поступивших в узел единиц два последующих бита имеют значения 1 и 0, то такая комбинация рассматривается как **граница кадра**. Если же после пяти единиц следующий бит равен 0, то он изымается, и текст кадра принимает исходный вид (см. последовательность 1).

Byte stuffing и Bit stuffing

Бит-стаффинг гораздо более экономичен, чем байт-стаффинг, так как вставляется бит а не байт.