Байт стаффинг

{В частности может хватить материала из лекции}

Допустим, в качестве стафф-байта я выбрал байт со значением 00. Это значит, что теперь это байт как бы выпал из набора байт, которые могут быть в моём информационном потоке. Но зато я получил однозначное обозначение границы пакетов.

пусть вместо нулевого информационного байта (00) передаётся последовательность из двух байт: 00 00. А в качестве границы между пакетами будем передавать последовательность из двух байт 00 01. Оставшиеся комбинации последовательностей из двух байтов (от 00 02 по 00 FF) будем считать ошибочными.

если программы написаны без ошибок, то таких последовательностей быть не может. Такие последовательности могут возникнуть, если при передаче проскочит помеха и изменит значение байта. Ну дак! Это и есть — «битый» байт. Какая разница — где это битый байт получился — во время приёма обычных байтов, или во время стаффинга. Однозначно — полученный пакет в мусорную корзину!

Если вы правильно выбрали значение стафф-байта, то объём трафика практически не увеличиться. Но зато теперь вы сможете чётко отслеживать границы пакетов.

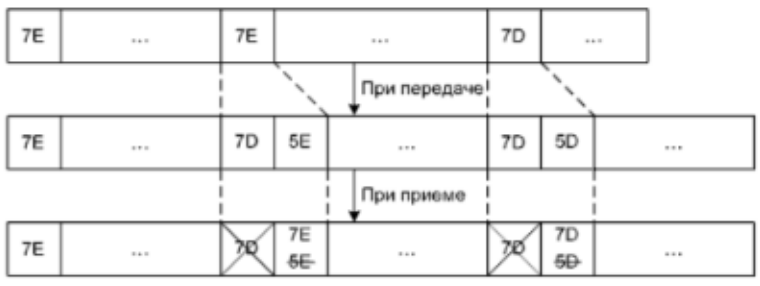
Информация из лекции

Понятно, что для правильной интерпретации пакета нужно его считать из канала полностью, причем с соблюдением последовательности. Если бы взаимодействующие станции работали бесконечно и находились в соответствующей степени готовности, то это не составляло бы особого труда. Но, поскольку станция-приемник может подключиться к каналу (да и вообще начать работать) в произвольный момент времени, возникает проблема, связанная с распознаванием флага начала пакета.

Флаг начала пакета представляет собой зарезервированную цифровую последовательность, которая собственно позволяет станции-приемнику определить начало пакета.

Проблема заключается в том, что такая же последовательность вполне может встретиться в пакете и после флага начала. Следовательно, возникает задача обеспечения уникальности флага начала пакета, то есть исключения этой последовательности из оставшейся части пакета. Это достигается за счет действия, заключающегося в модификации следующей за флагом цифровой последовательности, которое в бит-ориентированных системах называют бит-стаффингом (bit stuffing), а в байт-ориентированных -- байт-стаффингом (byte stuffing).

Цель байт-стаффинга полностью совпадает с целью бит-стаффинга. В сравнении с алгоритмами бит-стаффинга, алгоритмы байт-стаффинга манипулируют байтами, являются более сложными и более «затратными», но при программировании они позволяют избежать битовых операций (бит-стаффинг, в отличие от байт-стаффинга, обычно реализуют аппаратно).



Единственным способом обеспечения уникальности флагового байта является замена совпадающего с ним байта на некий выбранный другой. Но возникает вопрос, как принимающая сторона отличит замененный байт от такого же незамененного. Решением является применение так называемого ESC-символа. Наличие ESC-символа говорит станции-приемнику о факте замены, а следующий за ESC-символом символ -- код замены позволяет определить какая замена была осуществлена. Байт-стаффингу можно подвергать целые группы символов.

Полезный ссылки

Лекция №3
https://zhevak.wordpress.com/2018/08/17/байт-стаффинг-в-каналах-передачи-данн/ -- очень крутая статья