

1) Зачем в mlt3 чередование трёх уровней 1?

Ответ: Три уровня необходимы для плавного нарастания сигнала для снижения частоты сигнала, чтобы была возможность передать сигнал по медному кабелю.

2) Зачем нужно скремблирование?

Ответ: Скремблирование необходимо для защиты данных. Поток данных шифруется по случайным последовательностям, расшифровывается с помощью обратных алгоритмов.

3) Как расширение спектра позволяет бороться с межсимвольной интерференцией?

Ответ: Отражённый сигнал приходит на предыдущей частоте, не мешая текущему прямому сигналу.

4) Зачем в витой паре дифференциальный приём?

Ответ: Электрический сигнал передаётся в виде дифференциальной пары сигналов, каждый по своему проводнику, но один представляет инвертированный сигнал другого, противоположный по знаку. Приёмник дифференциального сигнала реагирует на разницу между двумя сигналами.

5) Чем свёрточное кодирование отличается от блочного помехоустойчивого?

Ответ: Блочный код обрабатывает данные блоками по  $k$  бит, выдавая на выходе блок из  $m$  бит ( $m > k$ ). Свёрточное кодирование обеспечивает преобразование  $K_2$  (где  $K_x$  - число битов, переносимых одним блоком) входных бит типа 2, полученных в результате блочного кодирования, в  $K_3$  бит выходной последовательности, причём  $K_3 > K_2$ . Каждый бит выходной последовательности получается как результат суммирования по модулю 2 нескольких следующих друг за другом битов входной последовательности.

6) Зачем скремблирование с точки зрения синхронизации?

Ответ: Синхронизация необходима для того, чтобы передатчик и приёмник работали синхронно в такт. Чтобы в момент передачи потока информации, приёмник был готов её принять. Скремблер позволяет зашифровать поток так, что на выходе в каждой последующей битовой позиции вероятность появления 0 или 1 одинаковая. Поэтому возможно выделить тактовую частоту и постоянную мощность сигнала, что обеспечивает синхронизацию.

7) Чем удобнее HTTP2 по сравнению с HTTP1? Что такое проблема HOL в HTTP1? Чем обусловлена проблема HOL в HTTP2? В чем недостаток текстового протокола?

Ответ: Блокировка заголовка строки в терминах HTTP часто относится к тому факту, что каждый браузер / клиент имеет ограниченное количество подключений к серверу и при выполнении нового запроса по одному из этих подключений необходимо дожидаться завершения предыдущих, прежде чем он сможет сработать.

Запросы заголовка строки блокируют последующие.

HTTP / 2 решает эту проблему, вводя мультиплексирование, чтобы вы могли отправлять новые запросы через одно и то же соединение, не дожидаясь завершения предыдущих.

Теоретически конвейерная обработка HTTP / 1.1 также предлагала способ обойти HOL, но на практике это было сложно и подвержено ошибкам. Из-за этого он не получил широкого распространения в Интернете даже до сегодняшнего дня.

Однако HTTP / 2 по-прежнему страдает от другого типа HOL, а именно от уровня TCP. Один потерянный пакет в потоке TCP заставляет все потоки ждать, пока этот пакет не будет повторно передан и получен. Этот HOL обрабатывается с помощью протокола QUIC - это «TCP-подобный» протокол, реализованный через UDP, где каждый поток независим, так что потерянный пакет останавливает только конкретный поток, которому принадлежит потерянный пакет, в то время как другие потоки могут продолжаться.

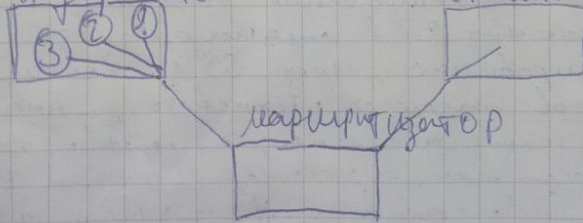
8) Нарисуйте таблицу NAT для случая 3 млн компьютеров в сети, и с внешним адресом класса C, как будто три узла из сети отправляют пакет.

Ответ:

$2^{32} = 4194304$  всего машин. 4194301

класс C: маска /24: 255.255.255.0

$32 - 22 = 10$ ,  $\Rightarrow$  маска /10: X.X.X.X/10 или  
внутр. сеть X.X.X.X mask 255.192.0.0



192.168.1.2	58789	$\rightarrow$	37.21.254.78	<del>594175-221</del>
192.168.1.3	52801	$\rightarrow$	37.21.254.78	59418
192.168.1.6	52313	$\rightarrow$	37.21.254.78	59418