

Лекция 3. Канальный уровень

Отвечает за передачу **кадров** между компьютерами, непосредственно связанными каналом передачи данных. Делится на два подуровня:

- MAC реализуется аппаратно, в микросхемах сетевых карт, адаптеров, модемов;
- LLC реализуется программно, в виде драйверов сетевых устройств.

Основные задачи

- передача кадров между непосредственно связанными компьютерами;
- управление доступом к линии связи;
- управление потоком;
- обнаружение и исправление ошибок.

Типы каналов (среды передачи данных)

- точка-точка (двухточечные);
- широковещательные (Ethernet, WiFi);
- последовательные (переключаемые).

Широковещательные, когда один канал используется несколькими системами для взаимодействия. В случае широковещательного канала существует проблема доступа к передающей среде. Решение — управление доступом.

Обнаружение и исправление ошибок

Для выявления ошибок передачи данных по сети применяются коды двух типов:

- коды обнаружения ошибок;
- корректирующие коды (позволяющие обнаруженную ошибку исправить).

Кодирование заключается в добавлении к передаваемым информационным битам дополнительных контрольных битов. Контрольные биты при этом могут располагаться отдельно от информационных битов (как в коде CRC, он рассматривается ниже), либо перемежку с информационными (код Хэмминга).

Контроль четности

Контрольный бит добавляется к информационным битам, исходя из правила, что общее количество единичных битов в полученном сообщении должно быть четным числом. Одиночный контрольный бит не позволяет определить положение ошибочного бита (рис. 1, а). Если же это правило применяется к массиву битов (рис. 1, б), то становится возможным вычислить положение ошибочного бита с целью его исправления.

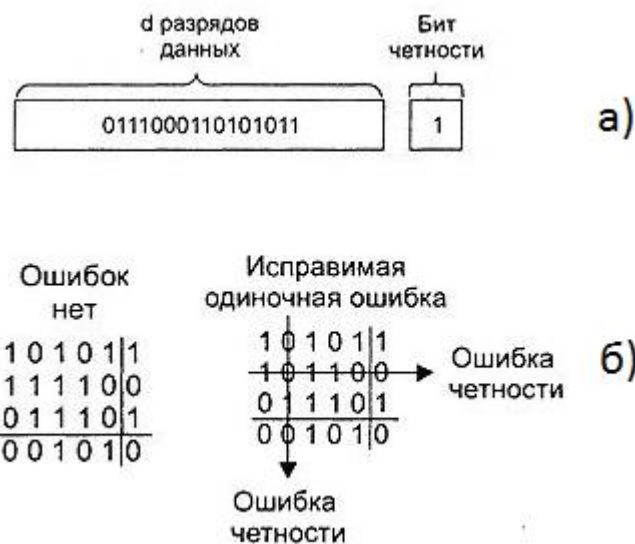


Рис. 1. Бит четности (а) и его модификация (б)

В рассмотренном ниже коде Хэмминга применяется разновидность данного алгоритма, связанная с вычислением бита нечетности.

Вычисление контрольной суммы

Алгоритм вычисления контрольной суммы **CRC** (Cyclic Redundancy Code — циклический избыточный код) — способ цифровой идентификации некоторой последовательности данных, который заключается в вычислении контрольного значения её циклического избыточного кода.

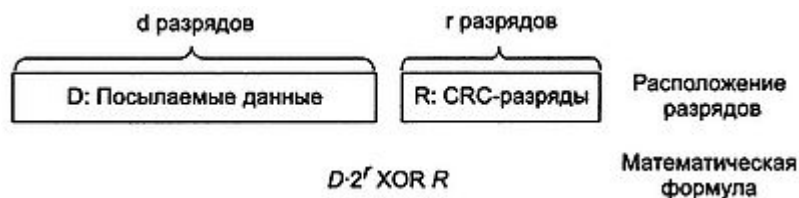


Рис.1. Код CRC (D – data, данные; R – remainder, остаток)

С точки зрения математики, контрольная сумма является разновидностью хеш-функции, используемой для вычисления контрольного кода — небольшого количества бит внутри большого блока данных, например, сетевого кадра или компьютерного файла, применяемого для обнаружения ошибок при передаче или хранении информации. Результат вычисления контрольной суммы добавляется в конец фрейма в виде трейлера непосредственно перед началом передачи или сохранения данных на каком-либо носителе информации. Впоследствии он проверяется для подтверждения её целостности. Популярность контрольной суммы обусловлена тем, что подобная проверка просто реализуема в двоичном цифровом оборудовании (на МАС подуровне), легко анализируется и хорошо подходит для обнаружения общих ошибок, вызванных наличием шума в каналах передачи данных.

Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов, то есть хеш-функция является, по сути, остатком от деления многочлена, соответствующего

входным данным, на некий фиксированный порождающий многочлен (примитивный полином). На практике же вместо полиномов применяются двоичные числа.

На рисунке 3 демонстрируется алгоритм вычисления CRCотправителем кадра.

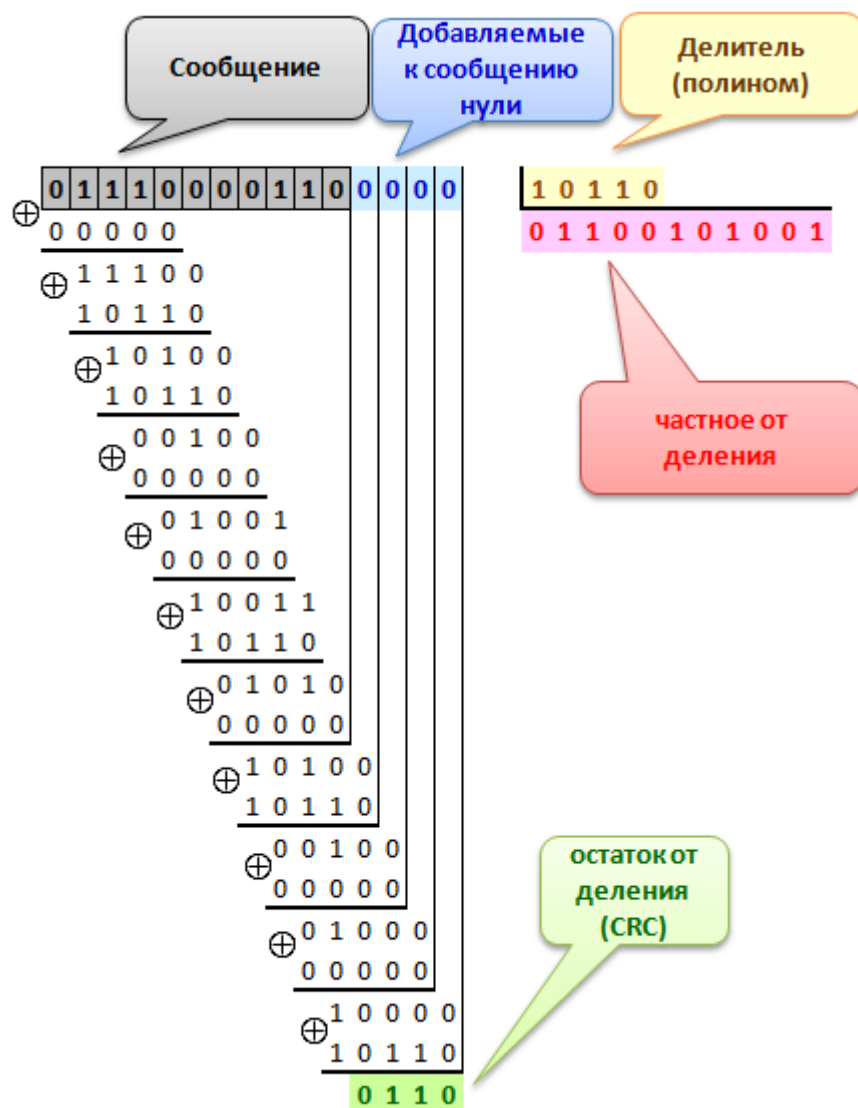


Рис.2. Вычисление CRC

Перед вычислением CRC к сообщению **0110000110** добавляются нулевые биты **0000**. Число добавляемых битов (4 бита) на единицу меньше длины делителя **10110** (5 битов). Само вычисление CRC отличается от привычной нам операции деления тем, что вместо вычитания применяется побитовая операция **XOR** (Исключающее ИЛИ), обозначаемое как $\oplus$. Результирующий бит будет равен 1 лишь в случае, если сравниваемые биты будут различны:

$$\begin{aligned} 1 \oplus 1 &= 0 \\ 0 \oplus 0 &= 0 \\ 1 \oplus 0 &= 1 \\ 0 \oplus 1 &= 1 \end{aligned}$$

Отправитель затем формирует кадр, к которому в качестве трейлера «прицепляет» вычисленный CRC (рис. 4).

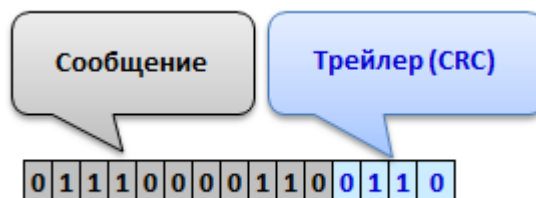


Рис. 4. Сформированный кадр

В коде CRC контрольные биты располагаются отдельно от информационных.

Как же получатель кадра удостоверится, что передача по сети полученного сообщения прошла без ошибок? Он пересчитывает контрольную сумму, используя тот же полином (рис. 5).

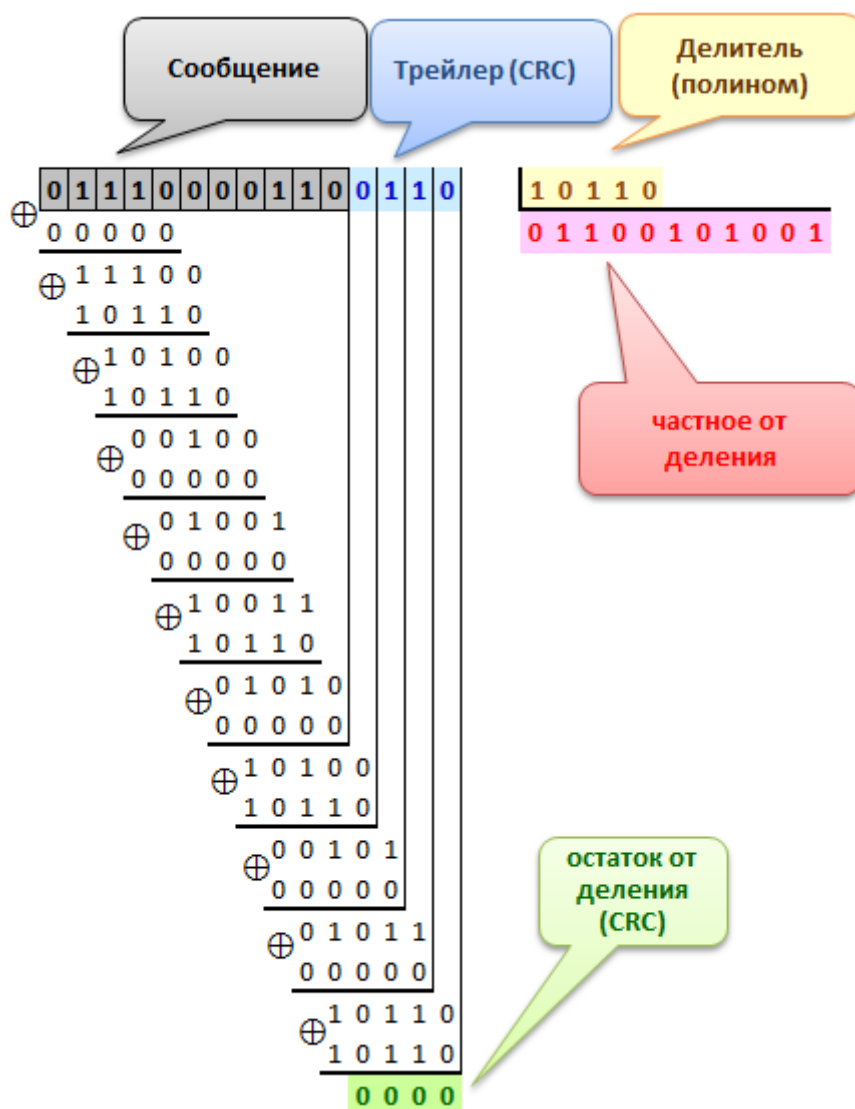


Рис. 5. Проверка полученного кадра на наличие ошибки

Если в полученном кадре отсутствуют ошибочные биты, то остаток должен быть нулевой! Смоделируем ситуацию, когда один бит в сообщении был изменен в ходе передачи (он выделен красным цветом).

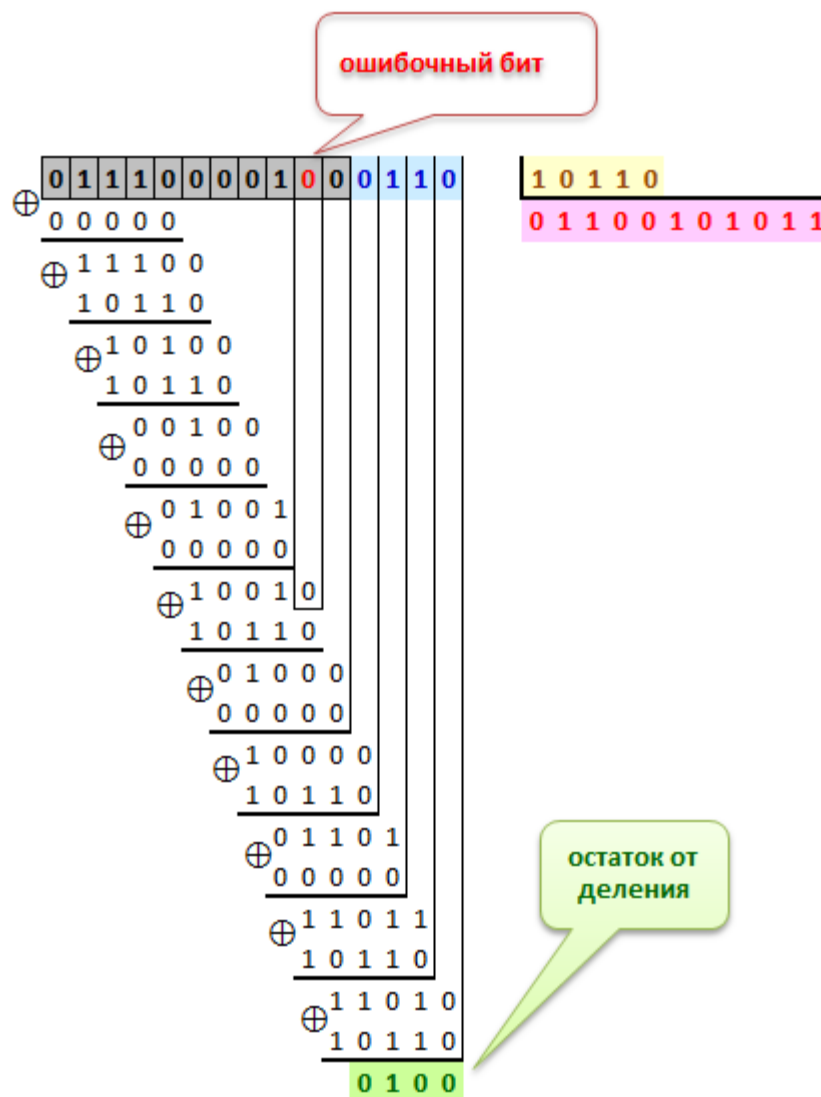


Рис. 6. Выявление ошибки в кадре

В данном случае (рис. 6) контрольная сумма отлична от нуля. Таким образом, для выявления факта наличия ошибок передачи данных достаточно заново вычислить CRC. CRC не позволяет выяснить в каком месте сообщения бит или биты были изменены, а, следовательно, этот алгоритм не относится к корректирующим кодам. Зададимся вопросом, а может ли быть так, что ошибочный бит (или несколько ошибочных битов) присутствует в принятом сообщении, а вычисленная контрольная сумма при этом оказалась равной 0? Может! Вероятность случайного выпадения комбинации в виде 0000 равна $\frac{1}{2^4}$. Это означает, что приемник будет иногда принятые ошибочные кадры принимать за правильные. Снизить вероятность принятия таких неправильных решений можно, увеличив длину порождающего полинома. Так, в Ethernet¹-технологии используется полином 32-й степени $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$, двоичное представление которого 100000100110000010001110110110111 (всего 33 бита). Более удобна запись этого полинома в шестнадцатеричном представлении: 0x04C11DB7, в котором самый старший, 33-й, единичный бит попросту игнорируется (он выделен красным цветом). Такой CRC имеет длину 32 бита и

¹ Данный полином помимо Ethernet применяется также в WiFi, FDDI.

вероятность случайного выпадения нулевого остатка (32 нулевых бита) для ошибочного кадра будет ничтожно малой: $\frac{1}{32^4}$.

Таким образом, второй уровень модели OSI – канальный – выполняет важную функцию, а именно отвечает за выявление ошибок, появляющихся при передаче сообщения по сети. Сообщение 2-го уровня (кадр) имеет концевик, называемый трейлером, куда помещается контрольная сумма (CRC). Сообщения, которые были повреждены в каком-то сегменте сети, в другие сегменты не пропускаются, так как сетевые устройства выявившие наличие ошибок в сообщениях (кадрах), их попросту уничтожают.

Код Хэмминга

Данный алгоритм относится к корректирующим. В качестве примера рассмотрим одну из ее модификаций, обозначаемую как (7,4). Здесь, 7 – общее число битов вычисленного кода, 4 – число информационных битов в коде. Остальные три бита кода являются контрольными. Биты разбиваются на группы, причем информационные могут присутствовать одновременно в нескольких группах, а контрольные – лишь в одной. Контрольные биты являются битами нечетности и располагаются в своей группе в самом младшем разряде. Номера позиций разрядов для каждой из групп следующие:

Группы	Позиции разрядов
1	7, 5, 3, 1
2	7, 6, 3, 2
3	7, 6, 5, 4

Красным цветом отмечены позиции контрольных битов, информационные же биты находятся на позициях 3, 5, 6 и 7.

В качестве примера кодирования выберем информационное сообщение, представленное в виде единственной шестнадцатеричной цифры **C₁₆**. Двоичное ее представление: **1100**. Разместим эти биты на позициях 3, 5, 6 и 7 (рис. 7).

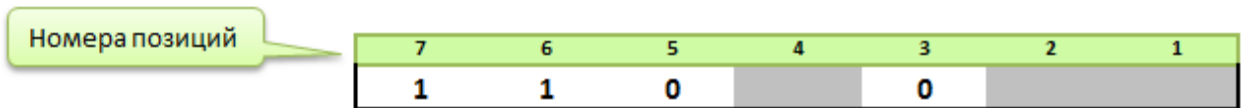


Рис. 7. Информационные биты

Далее вычисляем контрольные биты и помещаем их в соответствующие их группам места(рис. 8).

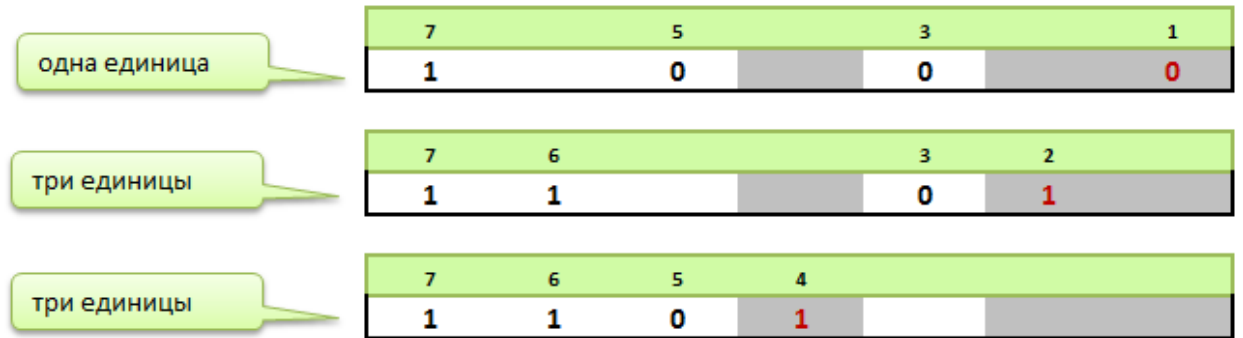


Рис. 8. Вычисление контрольных битов внутри групп

Напоминаем, что в данном случае контрольный бит является битом нечетности, т.е. количество единиц в группе должно быть нечетным числом. Таким образом, мы цифру C_{16} представили в виде 7-битовой комбинации **1101010**, называемой кодом Хэмминга.

Приемник, получив данную комбинацию, вычисляет **регистр ошибок (РО)**. РО представляет собой трехбитовое число, указывающее на номер позиции ошибочного бита. При отсутствии ошибки РО должен быть равным 0. Биты РО должны быть нулевыми, если в соответствующей им группе нечетное количество единиц. Для **1101010** во всех трех группах правило нечетности соблюдается и, соответственно РО равен **000**.

Допустим, что при передаче по сети был изменен 5-й бит, т.е. приемник получил искаженное сообщение в виде **1111010**. В таком случае вычисление РО даст значение **101**, что соответствует 5-й позиции кода (рис. 9).

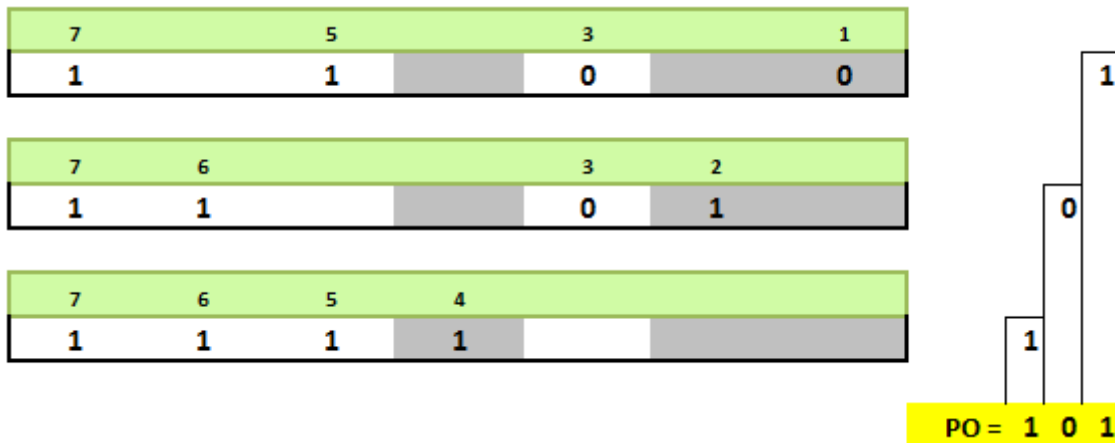


Рис. 9. Вычисление регистра ошибок

После исправления ошибки приемник отбрасывает контрольные биты и получает исходное сообщение C_{16} (рис. 10).

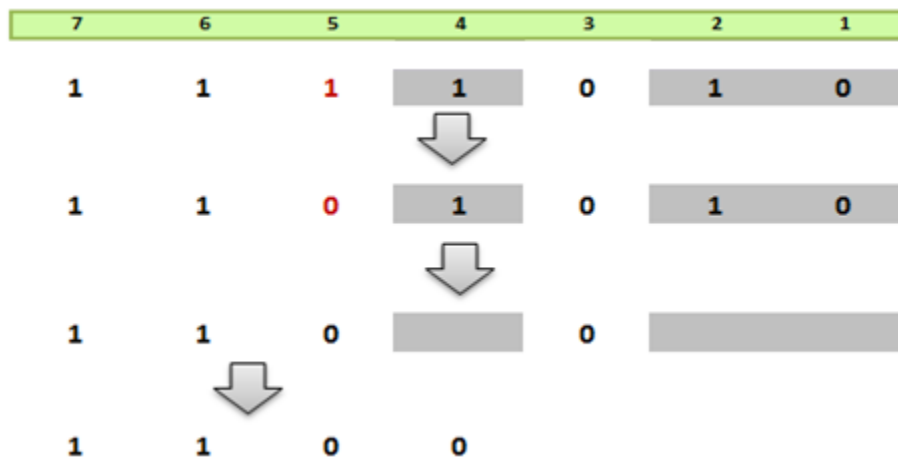


Рис. 10. Вычисление исходного сообщения

Подуровень управления доступом к среде

В зависимости от скорости передачи данных и передающей среды существует несколько вариантов технологий локальных сетей. Из них самой популярной является Ethernet.

Сеть Ethernet

Историческая справка. Зарождение Ethernet.

22 мая 1973 г. Меткалф представил к защите докторскую диссертацию, в которой описана сеть "по имени" Ethernet. Название, возможно, связано с особенностями ее первой инсталляции и изначального предназначения. В этой системе действительно использовалась радиосвязь для обеспечения работы в кампусе гавайского университета Алоха. Таким образом, жители цветущего острова стали свидетелями рождения этого феномена. В июле 1976 г. в журнале "Communicationsofthe ACM" появилась статья, написанная Бобом Меткалфом и Дэвидом Боггзом (DavidBoggs) под названием "Ethernet: DistributedPacketSwitchingforLocalComputerNetworks". В ней были изложены основные принципы работы кабельной Ethernet. Далее спецификации сети начали мутировать с невероятной скоростью, так что сравнивать творение вышеупомянутых ученых с каким-нибудь современным 1000Base-LX - весьма неблагоприятное занятие, схожее разве что с попыткой найти родство между Фордом-Т и болидом Ф-1. А пока "эфирная сеть" спокойно работала на скорости 2,94 Mbps, и даже сам Меткалф не мог предположить, как сложится дальнейшая судьба его детища. По чертежам ученого в корпорации Херох было создано вполне конкретное оборудование. Четыре года спустя изобретатель решил самостоятельно заняться развитием сетевой технологии и образовал собственную компанию 3Com. Одновременно "хитрый" Меткалф устроился консультантом в корпорацию DEC. Таким образом, если он и не стал катализатором, способствовавшим появлению стандартов семейства IEEE 802, то во всяком случае в момент принятия решения о разработке спецификаций Ethernet-подобной сети группой DEC, Intel, Xerox оказался в эпицентре событий. Замысел заключался в проектировании 10-мегабитовой сети DIX (по первым буквам названий компаний-участников альянса) без использования патентов Херох. На чем базировался столь откровенный альтруизм упомянутой фирмы, нам не известно. Впрочем, в ее адрес всегда раздавались упреки по поводу откровенно слабой политики в области управления своей интеллектуальной собственностью.

Технология Ethernet очень популярна на рынке локальных сетей. В 80-е годы и в начале 90-х ей приходилось конкурировать с множеством альтернативных технологий локальных сетей, включая сети TokenRing, FDDI и ATM. Некоторым из этих технологий удавалось временно захватывать часть рынка. Но технология Ethernet, разработанная в середине 70-х, продолжает свой рост и развитие и по сей день, а в последние годы она прочно заняла доминирующее положение на рынке. Сегодня Ethernet представляет собой преобладающую технологию локальных сетей, и, похоже, такая ситуация сохранится в обозримом будущем. Значение Ethernet в локальных сетях так же велико, как и значение Интернета в глобальных сетях. Успеху технологии Ethernet способствовало множество причин. Во-первых, Ethernet была первой локальной сетью, получившей широкое распространение. Благодаря этому сетевые администраторы очень близко познакомились с ней и уже не желали переходить на другие технологии локальных сетей, когда те появились на рынке. Во-вторых, такие технологии, как Tokenring, FDDI и ATM, были

более сложными и дорогими, чем Ethernet, что еще больше препятствовало переходу на них. В-третьих, наиболее сильным стимулом перехода на другие технологии локальных сетей (например, FDDI или ATM), как правило, была более высокая скорость передачи данных. Однако Ethernet всегда отвечала ударом на удар, и каждая новая версия Ethernet не уступала конкурентам, а то и превосходила их по данному параметру. В начале 90-х была представлена коммутируемая Ethernet-сеть, что позволило снова повысить эффективную скорость передачи данных. Наконец, благодаря популярности Ethernet аппаратура Ethernet (адаптеры, хабы и коммутаторы) стала производиться массовым тиражом, в результате чего цены на нее упали до невероятно низкого уровня. Причиной столь низких цен также является тот факт, что протокол коллективного доступа, применяемый в сетях Ethernet, полностью децентрализован, что обуславливает простоту аппаратуры.

Манчестерский код²

Ни в одной из версий Ethernet не применяется прямое двоичное кодирование бита 0 напряжением 0 В и бита 1 — напряжением 5В, так как такой способ приводит к неоднозначности. Если одна станция посылает битовую строку 00010000, то другая может интерпретировать ее как 10000000 или 01000000, так как они не смогут отличить отсутствие сигнала (0 В) от бита 0 (0 В). Можно, конечно, кодировать единицу положительным напряжением +1 В, а ноль — отрицательным напряжением -1 В (рис. 11).

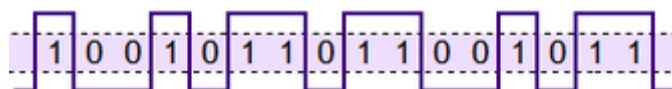


Рис. 11. Передача битов через канал связи

Но при этом все равно возникает проблема, связанная с синхронизацией передатчика и приемника. Разные частоты работы их системных часов могут привести к рассинхронизации и неверной интерпретации данных. В результате приемник может потерять границу битового интервала. Особенно велика вероятность этого в случае длинной последовательности нулей или единиц. Таким образом, принимающей машине нужен способ однозначного определения начала, конца и середины каждого бита без помощи внешнего таймера. Это реализуется с помощью манчестерского кодирования (рис. 12).

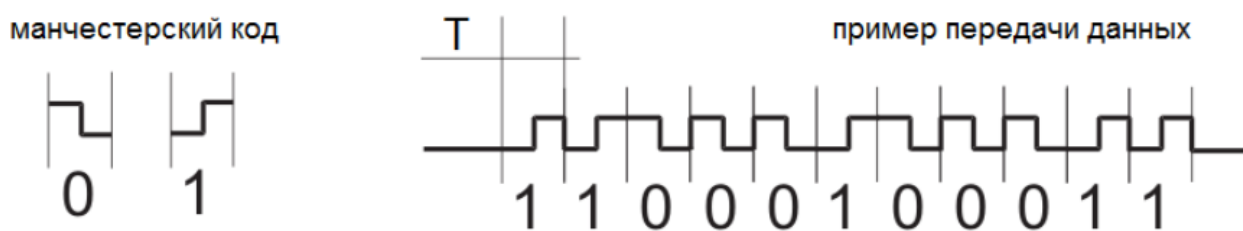


Рис. 12. Передача битов через канал связи

В манчестерском коде каждый временной интервал передачи одного бита делится на два равных периода. Бит со значением 0 кодируется высоким уровнем напряжения в первой половине интервала и низким — во второй половине, а единичный бит кодируется обратной последовательностью — сначала низкое напряжение, затем высокое. Такая

²Здесь на самом деле рассматривается физический уровень OSI

схема гарантирует смену напряжения в середине периода битов, что позволяет приемнику синхронизироваться с передатчиком. Недостатком манчестерского кодирования является то, что оно требует двойной пропускной способности линии по отношению к прямому двоичному кодированию, так как импульсы имеют половинную ширину. Например, для того чтобы отправлять данные со скоростью 100 Мбит/с, необходимо изменять сигнал 200 миллионов раз в секунду.

Формат кадра Ethernet

Преамбула (8 байт). Ethernet-кадр начинается с 8-байтового поля преамбулы. В каждый из первых 7 байт преамбулы записывается значение 10101010, а в последний байт — значение 10101011. Первые 7 байт должны «разбудить» принимающие адаптеры и помочь им синхронизировать свои таймеры с часами отправителя. Как уже отмечалось, адаптер А должен передать кадр со скоростью 10 Мбит/с, 100 Мбит/с или 1 Гбит/с в зависимости от типа локальной Ethernet-сети. Однако поскольку ничего не бывает абсолютно точным в реальном мире, скорость передачи всегда будет несколько отличаться от номинала. Величина этого отклонения скорости другим адаптерам локальной сети заранее не известна. Таким образом, первые 62 бита преамбулы, представляющие собой чередующиеся нули и единицы, позволяют приемнику с достаточной точностью настроиться на скорость передатчика, а последние два разряда (две единицы подряд) сообщают адаптеру В, что преамбула закончилась и следом идет уже первый информационный байт поля кадра.

Адрес получателя (6 байт). Это поле содержит MAC-адрес принимающего адаптера. Получив Ethernet-кадр с адресом получателя, отличным от собственного физического адреса или широковещательного адреса локальной сети, адаптер отбрасывает кадр. В противном случае он передает содержимое поля данных сетевому уровню.

Адрес отправителя (6 байт). Это поле содержит MAC-адрес адаптера, передающего кадр в локальную сеть.

Поле типа (2 байта). Поле типа позволяет локальной Ethernet-сети «мультиплексировать» протоколы сетевого уровня. Чтобы понять, что это означает, вспомним, что хосты могут помимо протокола IP использовать и другие протоколы. В самом деле, любой хост может поддерживать несколько протоколов сетевого уровня — разные протоколы для разных приложений. По этой причине, получив Ethernet-кадр, адаптер В должен знать, какому протоколу сетевого уровня он должен передать (то есть демультиплексировать) содержимое поля данных. Каждому сетевому протоколу (например, IP, Novell IPX или AppleTalk) присвоен зафиксированный в стандарте номер. Обратите внимание, что поле типа аналогично полю протокола в дейтаграмме сетевого уровня и полю номера порта сегмента транспортного уровня. Все эти поля служат для связи протокола одного уровня с протоколом уровнем выше.

Поле данных (от 46 до 1500 байт). Это поле содержит пакет (сообщение 3-го уровня OSI). Максимальная единица передачи (Maximal Transfer Unit, MTU) в Ethernet-сети составляет 1500 байт. Это означает, что если размер пакета превышает 1500 байт, тогда хост должен разбить ее на отдельные фрагменты. Минимальный размер поля данных равен 46 байт. Это означает, что если размер пакета 46 байт, то данные, помещаемые в это поле, дополняются байтами-заполнителями. При этом сетевой уровень получает пакет от канального уровня с этими дополнительными байтами и сам отсекает все лишнее, ориентируясь на поле длины в заголовке IP-дейтаграммы.

CRC (4 байта). Назначение поля CRC заключается в том, чтобы получающий адаптер мог определить, не исказился ли кадр при передаче, то есть обнаружить ошибки в кадре. Искривление битов в кадре может быть вызвано ослаблением сигнала в канале, скачками напряжения, наводками в кабелях и интерфейсных платах.