

Канальный уровень

24 апреля 2026 г.

Второй уровень - канальный уровень (data link layer). Мы обсудим алгоритмы, обеспечивающие надежную и эффективную передачу целых блоков информации, фреймов (сравните с физическим уровнем, задачей которого является передача отдельных битов), между двумя смежными устройствами.

Имеются в виду два компьютера, физически соединенные каналом связи, действующим по принципу провода (это может быть коаксиальный кабель, телефонная линия или беспроводной канал). Основное свойство такого канала заключается в том, что биты принимаются в том же порядке, в каком передаются.

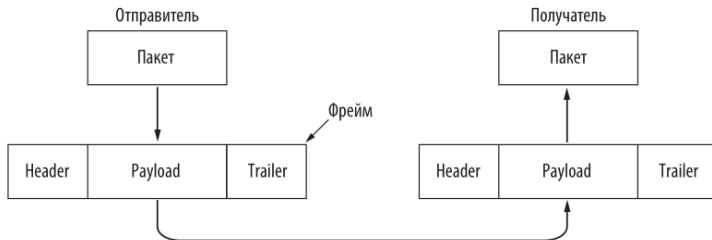
Канальный уровень использует определенные службы нижележащего физического уровня для отправки и получения битов по потенциально ненадежным коммуникационным каналам, которые могут терять данные.

Функции канального уровня:

- Обеспечение строго очерченного служебного интерфейса для сетевого уровня.
- Формирование отдельных фреймов из последовательностей байтов.
- Обнаружение и исправление ошибок передачи.
- Управление потоком данных, исключая «затопление» медленных приемников быстрыми передатчиками.

Для этих целей канальный уровень берет пакеты, полученные с сетевого уровня, и вставляет их в специальные **фреймы** (frames), также называемые кадрами, для передачи. В каждом фрейме содержатся поля **Header** (Заголовок), **Payload** (Пользовательские данные) и **Trailer** (Трейлер).

Канальный уровень



Канальный уровень управляет фреймами.

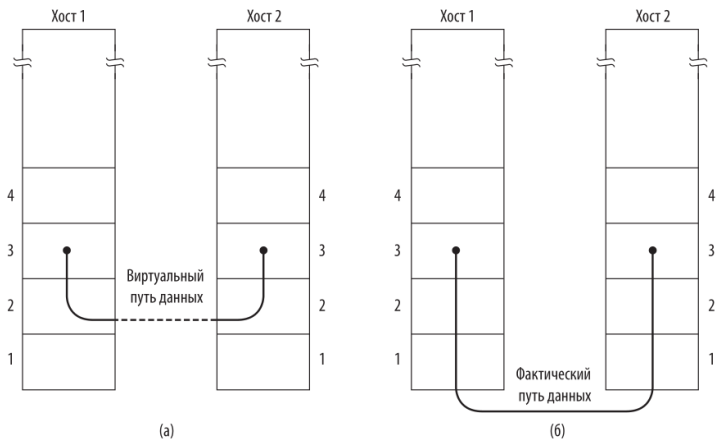
Канальный уровень предоставляет службы сетевому уровню.

Основная служба канального уровня передаёт данные с сетевого уровня отправляющего на сетевой уровень принимающего компьютера.

На сетевом уровне отправителя находится некая сущность (назовем ее **процессом**), которая передает пакеты на канальный уровень для их дальнейшей отправки по назначению.

Канальный уровень должен отправить данные адресату так, чтобы они достигли его сетевого уровня. В действительности данные передаются по пути, через физический уровень, однако проще представить себе два канальных уровня, которые связываются друг с другом с помощью протокола передачи данных.

Канальный уровень



Канальный уровень может предоставлять различные службы. Их набор может отличаться в разных протоколах. Как правило, встречаются следующие варианты, которые мы рассмотрим далее более подробно.

- Служба без подтверждений и без установки соединения.
- Служба с подтверждениями и без установки соединения.
- Служба с подтверждениями, ориентированная на установление соединения.

Алгоритм работы службы без подтверждений и без установки соединения заключается в том, что отправляющее устройство посылает независимые фреймы принимающему, которое не отвечает подтверждением о получении.

Хороший пример канального уровня, предоставляющего службу такого класса, — Ethernet. Соединения заранее не устанавливаются и не разрываются после передачи. Если какой-либо фрейм теряется из-за шума в линии, то на канальном уровне не предпринимается никаких попыток восстановить его.

Этот класс служб приемлем при очень низком уровне ошибок. В этом случае задача восстановления потерянных данных может быть решена на верхних уровнях.

Кроме того, такие службы подходят для трафика в реальном времени, например голосовых или видеоданных, поскольку в этом случае лучше получить данные в искаженном виде, чем с большой задержкой.

Следующим шагом в сторону повышения надежности является служба с подтверждениями и без установки соединения. При ее использовании соединение также не устанавливается, но получение каждого фрейма подтверждается.

Если в течение установленного интервала времени подтверждения не поступает, фрейм отправляется снова. Такая служба полезна при передаче по ненадежным каналам, в частности беспроводным. Хороший пример — протокол 802.11 (Wi-Fi).

Наиболее сложная служба, предоставляемая канальным уровнем, — ориентированная на установление соединения служба с подтверждениями. При использовании данного метода источник и приемник устанавливают соединение перед обменом данными. Каждый посылаемый фрейм нумеруется, а канальный уровень гарантирует, что все фреймы действительно приняты на другой стороне линии, что каждый фрейм принят всего один раз и что все они получены в правильном порядке.

Таким образом, ориентированная на установление соединения служба предоставляет процессам сетевого уровня эквивалент надежного потока битов. Она подходит для длинных ненадежных связей, таких как спутниковый канал или междугороднее телефонное соединение.

При использовании службы, ориентированной на установление соединения, передача данных состоит из трех фаз. В первой фазе возникает соединение, при этом обе стороны инициализируют переменные и счетчики, необходимые для слежения за тем, какие фреймы уже приняты, а какие — еще нет. Во второй фазе передаются фреймы с данными. Наконец, в третьей соединении разрывается, при этом освобождаются все переменные, буферы и прочие ресурсы, использовавшиеся его для поддержания.

Для обслуживания сетевого уровня канальный уровень использует службы, предоставляемые ему физическим уровнем.

Физический уровень принимает необработанный поток битов и пытается передать его по назначению. Однако поток битов, получаемый на уровне передачи данных, не застрахован от ошибок. Канальный уровень должен обнаружить ошибки и, если нужно, исправить их.

Формирование фрейма

Обычно канальный уровень разделяет поток битов на отдельные фреймы и вычисляет для каждого из них короткий маркер, называемый контрольной суммой. Контрольная сумма добавляется во фрейм перед тем, как он пересылается дальше.

Когда целевое устройство получает фрейм, контрольная сумма вычисляется заново на его основе. Если она отличается от содержащейся во фрейме, то канальный уровень понимает, что при передаче произошла ошибка, и принимает меры (например, игнорирует испорченный фрейм и посылает отправляющему устройству сообщение об ошибке).

Мы рассмотрим четыре метода маркировки границ фреймов.

- Подсчет байтов.
- Флаговые байты с байт-стаффингом.
- Использование сигнальных битов с бит-стаффингом.
- Применение запрещенных сигналов физического уровня.

Формирование фрейма

Первый метод формирования фреймов заключается в указании количества байтов фрейма в поле заголовка. Канальный уровень на принимающем устройстве видит это поле, узнает, сколько байтов последует, и таким образом определяет, где находится конец фрейма.

Формирование фрейма



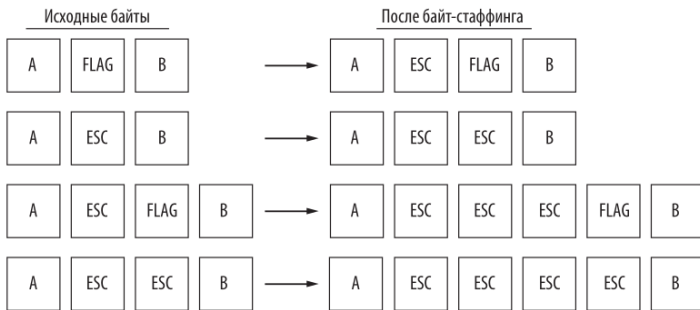
Недостаток такой системы в том, что при передаче может быть искажен сам счетчик. Например, если при подсчете байтов второго фрейма произойдет ошибка в единственном бите и число 5 превратится в 7, то целевое устройство потеряет синхронизацию и не сможет правильно обнаружить начало следующего фрейма.

Второй метод формирования фреймов решает проблему восстановления синхронизации после сбоя при помощи маркировки начала и конца каждого фрейма специальными байтами. Зачастую в качестве разделителя используется один и тот же байт, называемый **флаговым** (flag byte).

Формирование фрейма



(a)



(b)

Формирование фрейма

Два соседних флаговых байта говорят о том, что закончился один фрейм и начался другой. Таким образом, если приемник теряет синхронизацию, ему необходимо просто найти два флаговых байта, с помощью которых он распознает конец текущего фрейма и начало следующего.

В передаваемых данных, особенно если это двоичные данные (например, фотографии или музыка), запросто может встретиться последовательность, используемая в качестве флагового байта. Возникновение такой ситуации, скорее всего, собьет синхронизацию.

Один из способов решения проблемы — добавление специального escape-символа (знака переключения кода, ESC) непосредственно перед случайно совпавшим флаговым байтом внутри фрейма.

Формирование фрейма

Таким образом, настоящий флаг можно отличить по наличию или отсутствию перед ним ESC. Канальный уровень получателя вначале убирает эти escape-символы, затем передает фрейм на сетевой уровень. Такой метод называется **байт-стаффингом** (byte stuffing).

Следующий логичный вопрос: а что, если и символ ESC случайно окажется в данных? Решение такое же: вставить перед фиктивным escape-символом настоящий. На стороне получателя первый символ ESC будет удален, а следующий байт данных останется, даже если это еще один байт ESC или флаговый байт.

Формирование фрейма

Схема байт-стаффинга, представленная на илл. — это немного упрощенная модель протокола PPP (Point-to-Point Protocol, протокол «точка-точ-ка»), который служит для передачи пакетов по коммуникационным каналам и широко используется в сети интернет.

Формирование фрейма

Третий метод разделения потока битов на фреймы позволяет обойти недостатки байт-стаффинга, который обязывает использовать исключительно 8-битные байты.

Формирование фрейма

Делить данные на фреймы можно на уровне битов, причем фреймы могут содержать произвольное число битов и состоять из блоков любого размера. Каждый фрейм начинается и завершается специальной последовательностью битов, 01111110.

Формирование фрейма

Если в битовом потоке встретится пять единиц подряд, уровень передачи данных автоматически вставит в выходной поток нулевой бит. **Бит-стаффинг** (bit stuffing) аналогичен байт-стаффингу, при котором во фрейм вставляется escape-символ перед случайным флагом.

Формирование фрейма

(a) 0110111111111111111111110010

(б) 011011111011111011111010010

↑
Вставленные биты

(в) 0110111111111111111111110010

Этот метод гарантирует минимальную плотность передачи, помогающую сохранять синхронизацию на физическом уровне. По этой причине бит-стаффинг применяется в протоколе USB.

Формирование фрейма

Когда принимающая сторона встречает пять единиц подряд, за которыми следует ноль, она автоматически удаляет его.

Бит-стаффинг, как и байт-стаффинг, является абсолютно прозрачным для сетевого уровня обоих устройств. Если флаговая последовательность 01111110 встречается в данных пользователя, она передается в виде 011111010, но в памяти целевого устройства сохраняется в исходном виде: 01111110.

Формирование фрейма

Благодаря бит-стаффингу границы между двумя фреймами могут быть безошибочно распознаны с помощью флаговой последовательности. Таким образом, если принимающая сторона потеряет границы фреймов, ей нужно всего лишь отыскать в полученном потоке битов флаговый байт, поскольку он встречается только на границах фреймов и никогда не встречается в данных пользователя.

Последний метод формирования фреймов напрямую связан с особенностями физического уровня. При кодировании битов в виде сигналов для облегчения работы получающей стороны добавляются избыточные данные. Это означает, что в самих данных некоторые сигналы не появляются. Некоторые из зарезервированных сигналов можно применять для обозначения начальной и конечной границы фреймов.