

Управление доступом к среде (medium access control MAC)

21 мая 2026 г.

Главная проблема любой ширококвещательной сети — кому предоставить канал, если его запрашивают несколько пользователей. Протоколы, определяющие очередность использования линии, относятся к канальному уровню, точнее, к подуровню **управления доступом к среде** (MAC, Medium Access Control)

Традиционный способ распределения канала (например, телефонной линии) между многочисленными конкурирующими пользователями — разделение его пропускной способности с помощью одной из схем мультиплексирования, рассмотренных нами ранее (например FDM).

При наличии N пользователей полоса пропускания делится на N диапазонов одинаковой ширины, и каждому пользователю предоставляется один из них. Поскольку при такой схеме у каждого пользователя своя личная частотная полоса, то конфликтов не возникает.

При небольшом и постоянном числе абонентов, передающих стабильный поток или большие партии трафика, это простой и эффективный механизм распределения. Аналогичный пример для беспроводной связи — радиостанции FM-диапазона. Каждая станция получает часть FM-полосы и большую часть времени передает по ней свой сигнал.

Однако при большом и постоянно меняющемся числе пользователей или неравномерном трафике FDM не обеспечивает достаточно эффективного распределения.

Если в какой-то момент отправителей меньше, чем частей спектра, то значительная часть полосы пропускания простаивает. И наоборот, если пользователей больше, то некоторым из них придется отказать в доступе, даже если уже подключенные абоненты почти не используют канал.

Низкую эффективность статического FDM можно увидеть на примере простых вычислений теории массового обслуживания.

Среднее время задержки отправки фреймов

T - среднее время задержки для отправки фрейма по каналу емкостью C бит/с.

λ фреймов в секунду - средняя скорость приходов фреймов.

$1/\mu$ бит - средняя длина фреймов.

Теория массового обслуживания говорит о том, что

$$T = \frac{1}{\mu C - \lambda}$$

Требуется, чтобы случайность длительности промежутков между фреймами и длины фреймов соответствовала экспоненциальному распределению или, что эквивалентно, являлась результатом пуассоновского процесса.

Среднее время задержки отправки фреймов

В нашем примере C равно 100 Мбит/с, средняя длина фрейма $1/\mu = 10000$ бит, скорость входного потока $\lambda = 5000$ фреймов в секунду. Тогда $T = 200$ мкс.

Среднее время задержки отправки фреймов

Обратите внимание: если бы мы не учли задержки при формировании очереди и просто посчитали, сколько времени нужно на передачу фрейма длиной 10 000 бит по сети с пропускной способностью 100 Мбит/с, то получили бы неправильный ответ: 100 мкс. Этот результат верен лишь при отсутствии конкуренции за канал.

Среднее время задержки отправки фреймов

Теперь разделим канал на N независимых подканалов, каждый из которых имеет пропускную способность C/N бит/с. Средняя входная скорость в подканале теперь равна λ/N фреймов в секунду. Вычислив новое значение средней задержки T , получим:

$$T_N = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

В разделенном канале средняя задержка в N раз хуже, чем в канале, в котором все фреймы волшебным образом организованы в одну общую очередь.

Среднее время задержки отправки фреймов

Тот же вывод можно сделать на следующем примере: предоставляя доступ к банкоматам в холле банка, лучше организовать посетителей в одну общую очередь. Отдельные очереди могут привести к тому, что одни банкоматы будут простаивать, в то время как перед другими выстроится много людей.

Аргументы, применимые к FDM, относятся и к другим способам статического распределения канала. Можно использовать схему TDM и выделять каждому пользователю N-й слот. Но если абонент не будет передавать данные, этот слот просто пропадет.

С тем же успехом можно разделить сети физически. Если взять 100-мегабитную сеть и сделать из нее десять 10-мегабитных, статически распределив по ним пользователей, то в результате средняя задержка возрастет с 200 мкс до 2 мс.

Таким образом, ни один статический метод распределения каналов не подходит для неравномерного трафика, поэтому далее мы рассмотрим динамические методы.

Допущения, связанные с динамическим распределением каналов.

1. **Независимый трафик.** Модель состоит из N независимых станций (компьютеров, телефонов, персональных средств связи и т. д.), в каждой из которых программа или пользователь формируют фреймы для передачи. Ожидаемое число фреймов в интервале времени Δt равно $\lambda \Delta t$, где λ — постоянная величина (скорость прибытия новых фреймов). Как только фрейм сформирован, станция блокируется и ничего не делает, пока он не будет успешно передан.

Допущения, связанные с динамическим распределением каналов.

2. **Единый канал.** Он доступен для всех. Все станции могут передавать и принимать по нему данные. Они обладают одинаковой производительностью, хотя программно протокол может устанавливать для них различные роли (например, приоритеты).

Допущения, связанные с динамическим распределением каналов.

3. Наблюдаемые коллизии. Если два фрейма передаются одновременно, они перекрываются по времени, и в результате сигнал искажается. Такое событие называется коллизией (collision). Обнаруживать их могут все станции. Фрейм, искаженный из-за коллизии, должен быть передан повторно. Других ошибок, кроме тех, которые вызваны коллизиями, нет.

Допущения, связанные с динамическим распределением каналов.

4. Непрерывное/дискретное время. Если допустить, что время непрерывно, то передача фреймов может начаться в любой момент. В противном случае время разделяется на дискретные интервалы (слоты). Отправка фрейма происходит только с началом слота. Один слот может содержать 0 (свободный интервал), 1 (успешная передача) или более фреймов (коллизия).

Допущения, связанные с динамическим распределением каналов.

5. Контроль (опрос) несущей/его отсутствие. При контроле несущей станции определяют, свободна ли линия, прежде чем начать ее использовать. Если канал занят, станции не будут пытаться передавать по нему фреймы, пока тот не освободится. Если контроля несущей нет, то станции не могут заранее получить эту информацию. Они просто начинают передачу и только потом выясняют, была ли она успешной.

О приведенных выше допущениях следует сказать несколько слов. Первое утверждает, что фреймы приходят независимо друг от друга (как на разные станции, так и в пределах одной) и что они формируются непредсказуемо, но с постоянной скоростью.

В действительности это не слишком хорошая модель сетевого трафика, поскольку давно известно, что пакеты прибывают целыми последовательностями в определенные диапазоны временной шкалы

При этом пуассоновские модели, как их часто называют, находят широкое применение, в том числе потому, что они легко поддаются математическому описанию. Они позволяют анализировать протоколы, чтобы составить общее представление об изменении производительности с течением времени и о разнице между реализациями.

Причина существования двух альтернативных допущений для времени заключается в том, что дискретное время иногда помогает повысить производительность. Однако использующие его станции должны синхронизироваться с тактовым генератором или друг с другом, а это не всегда возможно. Мы рассмотрим оба варианта. В каждой конкретной системе работает только одно из возможных допущений.

Существует множество алгоритмов коллективного доступа. Мы рассмотрим наиболее интересные из них и примеры их применения на практике.

Решение было основано на использовании радиосистемы ближнего радиуса действия. Терминал каждого пользователя передавал фреймы на центральный компьютер в пределах общей полосы частот. Также был найден простой и элегантный метод решения проблемы распределения каналов.

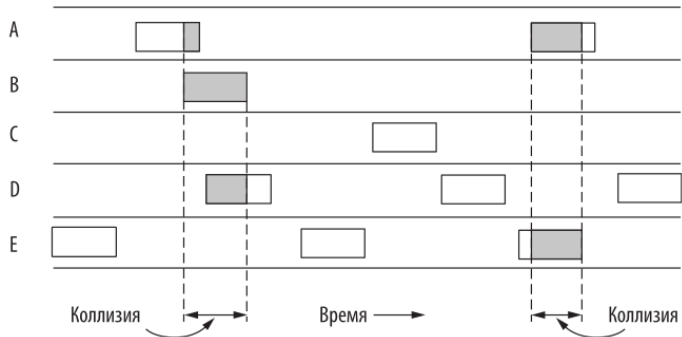
В основе ALOHA лежит простая идея: разрешить пользователям передачу, как только у них появляются данные для отправки. Конечно, будут возникать коллизии, приводящие к повреждению фреймов. Отправителям необходимо уметь обнаруживать такие ситуации.

Каждая станция отправляет свой фрейм центральному компьютеру, а тот передает его на все остальные станции. Отправитель прослушивает широкопередатную передачу, чтобы понять, насколько она была успешной.

Если фрейм был уничтожен, отправитель выжидает в течение случайного временного интервала и пытается переслать его снова. Время ожидания должно быть случайным, иначе одни и те же фреймы будут синхронно попадать в коллизию снова и снова.

Системы, в которых несколько пользователей используют один общий канал так, что время от времени возникают конфликты, называются **системами с конкуренцией** (contention systems).

Пользователь



Когда два фрейма одновременно пытаются занять канал, происходит коллизия (как видно на илл.). Оба фрейма искажаются. Даже если только один первый бит второго фрейма перекроет последний бит первого, оба фрейма полностью теряются (то есть их контрольные суммы не совпадут с правильными значениями). Позже они должны быть переданы повторно. В контрольной сумме нет (и не должно быть) различия между полным и частичным искажением информации. Потеря есть потеря.

Какова эффективность канала ALOHA? Другими словами, какая часть передаваемых фреймов избежит коллизии при таком беспорядке?

Пусть «время фрейма» означает интервал, требуемый для отправки стандартного фрейма фиксированной длины (это длина фрейма, деленная на скорость передачи). Допустим, новые фреймы, порождаемые станциями, хорошо распределены по Пуассону со средним значением N фреймов за интервал.

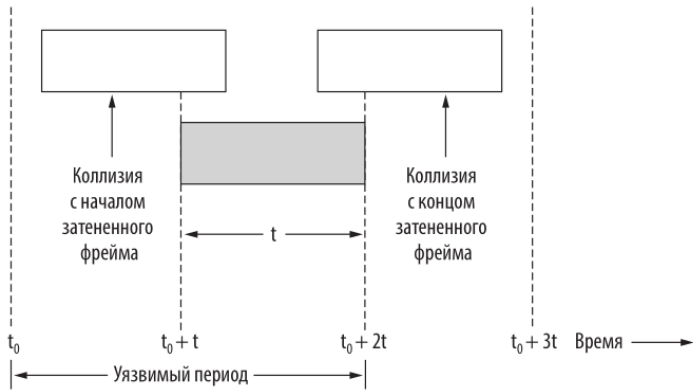
Если $N > 1$, это означает, что сообщество пользователей формирует фреймы с большей скоростью, чем может быть передано по каналу, и почти каждый фрейм будет искажаться. Разумнее предположить, что $0 < N < 1$.

Помимо новых, станции повторно отправляют старые фреймы, пострадавшие от столкновений. Предположим, что старые и новые фреймы хорошо распределены по Пуассону со средним значением G фреймов за интервал.

Очевидно, что $G \geq N$. При малой загрузке канала (то есть при $N \approx 0$) коллизий возникает мало, как и повторных передач, то есть $G \approx N$. При большой загрузке коллизий много, а следовательно, $G > N$.

Какая бы ни была загрузка, производительность канала S будет равна предлагаемой загрузке G , умноженной на вероятность успешной передачи P_0 , то есть $S = GP_0$, где P_0 — вероятность того, что фрейм не повредится в результате коллизии.

Фрейм не пострадает, если в течение интервала времени его передачи больше ничего не отправлять, как показано на илл.



При каких условиях фрейм, затененный на рисунке, будет передан без повреждений? Пусть t — это время, требуемое для передачи фрейма. Если пользователь сформирует фрейм в интервале времени между t_0 и $t_0 + t$, то его конец столкнется с началом затененного фрейма. При этом судьба затененного фрейма предрешена еще до того, как будет послан его первый бит.

Но в чистой ALOHA станции не прослушивают линию до начала передачи и у них нет способа узнать, что канал занят и по нему уже передается фрейм. Аналогичным образом, любой другой фрейм, передача которого начнется в интервале от $t_0 + t$ до $t_0 + 2t$, столкнется с концом затененного фрейма.

Вероятность того, что за время фрейма вместо G будет сформировано k фреймов, можно вычислить по формуле распределения Пуассона:

$$Pr[k] = \frac{G^k e^{-G}}{k!}$$

Таким образом, вероятность генерации нуля фреймов в течение этого интервала равна e^{-G} . Среднее количество фреймов, сформированных за интервал длиной в два фрейма, равно $2G$.

Вероятность того, что никто не начнет передачу в течение всего уязвимого периода, равна $P_0 = e^{-2G}$. Учитывая, что $S = GP_0$, получаем:

$$S = Ge^{-2G}$$



Максимальная производительность достигает значения $S = 1/(2e)$, что приблизительно равно 0.184 при $G = 0.5$. Другими словами, лучшее, на что мы можем надеяться, — это использовать канал на 18%. Этот результат несколько разочаровывает, но если каждый передает данные тогда, когда хочет, трудно ожидать стопроцентного успеха.

Вскоре после появления системы ALOHA Робертс (Roberts, 1972) опубликовал метод, позволяющий удвоить ее производительность. Он предложил разделять время на дискретные интервалы, соответствующие одному фрейму. Их называют слотами (slots).

При таком подходе пользователи должны согласиться с определенными временными ограничениями. Одним из способов достижения синхронизации является установка специальной станции, которая генерирует синхронизирующий сигнал в начале каждого интервала.

Дискретная система ALOHA

Дискретная ALOHA Робертса отличается от чистой ALOHA тем, что станция не может начинать передачу сразу после ввода пользователем строки. Вместо этого она должна дождаться начала нового слота. Таким образом, система ALOHA с непрерывным временем превращается в дискретную.

Уязвимый временной интервал теперь в два раза короче. Чтобы понять это, взгляните на илл. и представьте, какие теперь возможны коллизии. Вероятность отсутствия трафика в течение того же интервала, в котором передается тестовый фрейм, равна e^{-G} . В результате получаем:

$$S = Ge^{-G}.$$

Как видно из илл. , дискретная ALOHA имеет пик при $G = 1$.
Производительность канала составляет $S = 1/e$, что приблизительно равно 0,368, то есть в два раза больше, чем в чистой ALOHA.

Если система работает при $G = 1$, вероятность появления пустого слота равна 0,368 (согласно уравнению). Для дискретной системы ALOHA в оптимальной ситуации 37% интервалов будут пустыми, 37% — с успешно переданными фреймами и 26% — с коллизией.

При более высоких значениях G количество пустых интервалов уменьшается, но коллизий становится значительно больше.

Чтобы увидеть, как быстро растёт их число, рассмотрим передачу тестового фрейма. Вероятность того, что он избежит коллизии, равна e^{-G} . Фактически это означает, что все остальные станции будут молчать в этом слоте. Таким образом, шанс возникновения коллизии равен $1 - e^{-G}$. Вероятность передачи фрейма ровно за k попыток (то есть после $k - 1$ коллизий, за которыми следует успешная отправка) равна

$$P_k = e^{-G}(1 - e^{-G})^{k-1}$$

Ожидаемое число попыток передачи E для одной строки, введенной на терминале, равно

$$E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}(1 - e^{-G})^{k-1} = e^G$$

Поскольку E экспоненциально зависит от G , небольшое увеличение нагрузки в канале может серьезно снизить его производительность.

Протоколы множественного доступа с контролем несущей

В дискретной системеALOHA максимальный коэффициент использования канала равен $1/e$. Такой скромный показатель неудивителен, поскольку станции передают данные, когда хотят, не считаясь с тем, что делают остальные. В такой системе неизбежно возникает большое количество коллизий.

Протоколы множественного доступа с контролем несущей

Однако в локальных сетях можно организовать процесс так, что станции будут учитывать поведение друг друга. За счет этого можно достичь намного большего значения, чем $1/e$.

Протоколы множественного доступа с контролем несущей

Протоколы, в которых станции прослушивают среду передачи данных и действуют в соответствии с этим, называются **протоколами с контролем (опросом) несущей** (carrier sense protocols).

Настойчивый и ненастойчивый CSMA

Мы начнем с протокола CSMA с настойчивостью 1 (Carrier-Sense Multiple Access — множественный доступ с контролем несущей).

Настойчивый и ненастойчивый CSMA

Когда у станции появляются данные для передачи, она сначала прослушивает канал, проверяя, свободен ли он. Когда канал простаивает, станция сразу отправляет данные. Если же он занят — она ждет освобождения линии, а затем передает фрейм.

Настойчивый и ненастойчивый CSMA

В случае коллизии станция ждет в течение случайного интервала времени, затем снова прослушивает канал, и если он свободен, повторяет передачу. Такой протокол называется CSMA с настойчивостью 1, поскольку станция передает фрейм с вероятностью 1, как только обнаружит, что канал свободен.

Может показаться, что в этой схеме отсутствуют коллизии (за исключением редких случаев одновременной отправки), но это не так. Допустим, две станции пришли в состояние готовности во время работы третьей. Обе они ждут, пока та не закончит отpravку, после чего начинают передавать одновременно, и в результате происходит коллизия. Если бы станции не были столь нетерпеливы, коллизий было бы меньше.

На число коллизий значительное влияние оказывает задержка распространения сигнала. Существует небольшая вероятность того, что как только одна станция начнет передачу, другая также придет в состояние готовности и опросит канал. Если сигнал от первой станции еще не успел достичь второй, она решит, что канал свободен, и также начнет передачу, в результате произойдет коллизия. Эта вероятность зависит от числа фреймов, помещающихся в канал, то есть от показателя **«полоса пропускания, умноженная на задержку»** (bandwidth-delay product) для данного канала.

Если канал вмещает лишь небольшую часть фрейма, как бывает в большинстве LAN, где задержка распространения невелика, то и шанс коллизии мал. Чем больше время распространения сигнала, тем выше вероятность коллизий и ниже производительность протокола.

Однако даже такая система значительно лучше чистой ALOHA, так как обе станции воздерживаются от передачи, пока передает третья станция, в результате чего ее фрейм остается неповрежденным.

Далее мы рассмотрим **ненастойчивый протокол CSMA**. В нем предпринята попытка сдержать стремление станций начинать передачу, как только канал освобождается.

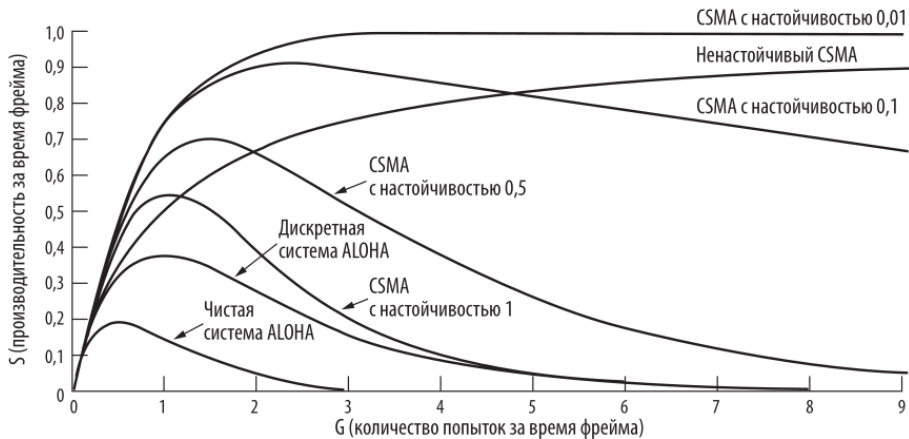
Как и в случае с предыдущим протоколом, прежде чем начать передачу, станция опрашивает канал. Если в данный момент никто ничего не передает, станция сразу же начинает передачу сама. Однако если канал занят, станция не ждет его освобождения, постоянно прослушивая его и пытаясь захватить сразу, как только он освободится, как в предыдущем протоколе. Вместо этого она ждет в течение случайного интервала времени, а затем снова прослушивает линию.

В результате количество коллизий уменьшается и повышается эффективность использования канала. При этом увеличивается интервал ожидания по сравнению с протоколом CSMA с настойчивостью 1.

Третий протокол с контролем несущей — CSMA с настойчивостью р.
Он применяется в дискретных каналах и работает следующим образом.

Когда станция готова передавать, она опрашивает канал. Если канал свободен, она с вероятностью p начинает передачу. С вероятностью $q = 1 - p$ она отказывается от передачи и ждет начала следующего слота.

Этот процесс повторяется до тех пор, пока фрейм не будет передан или какая-либо другая станция не начнет передачу. В последнем случае станция ведет себя так же, как в случае коллизии. Она ждет в течение случайного интервала времени, после чего начинает все заново. Если при первом прослушивании канала выясняется, что он занят, станция ждет следующий слот, а затем применяет тот же алгоритм.



Настойчивый и ненастойчивый протоколы CSMA, несомненно, более эффективны по сравнению с системой ALOHA, поскольку гарантируют, что ни одна станция не начнет передачу, если канал уже занят. Однако если две станции, обнаружив, что канал свободен, одновременно начали передачу, коллизия все равно произойдет.

Еще одно улучшение — станции способны быстро распознать коллизию и немедленно прекратить передачу (а не доводить ее до конца), так как данные все равно искажены. Эта стратегия обеспечивает более экономное использование времени и пропускной способности канала.

Протокол CSMA с обнаружением коллизий (CSMA/CD) лежит в основе классических локальных сетей Ethernet и потому заслуживает подробного рассмотрения.

Оборудование станции должно прослушивать канал во время передачи. Если считываемый сигнал отличается от пересылаемого, становится понятно, что произошла коллизия. Полученный сигнал не обязательно должен идеально совпадать с отправленным.

В протоколе CSMA/CD, как и во многих других протоколах LAN, применяется концептуальная модель, показанная на илл. В момент времени t_0 одна из станций заканчивает передачу фрейма. Все остальные станции, готовые к передаче, теперь могут попытаться отправить свои фреймы.

Если две или несколько станций одновременно начнут передачу, то произойдет коллизия. Обнаружив ее, станция прекращает передачу, ждет в течение случайного периода времени, после чего пытается снова, при условии, что к этому моменту не начала передачу другая станция.



