Network: Часть 4. Канальный уровень

02.09.2026 14:23 edited 02.09.2026 14:23

Введение

Канальный уровень (Data Link Layer, L2) — второй уровень модели OSI, который превращает «сырой» физический канал в надежный канал передачи кадров между соседними узлами. Если физический уровень оперирует битами, то канальный уровень работает с кадрами (frames). Это один из важных уровней сетевой архитектуры, поскольку именно здесь начинается настоящая «сетевая магия» — структурирование хаотичного потока битов в осмысленные блоки данных с адресацией, контролем ошибок и управлением доступом к среде.
В типичной передаче данных путь выглядит следующим образом: HTTP → TCP → IP → Ethernet (канальный уровень). На канальном уровне IP-пакет упаковывается в Ethernet-фрейм с добавлением служебных полей, таких как MAC-адреса и контрольная сумма CRC.

Основные задачи канального уровня

Канальный уровень решает четыре фундаментальные задачи.

Предоставление сервисов сетевому уровню

Канальный уровень может предоставлять три типа сервиса сетевому уровню:

1. **Без подтверждений (ненадежный)** — кадры просто отправляются без проверки доставки. Пример: классический Ethernet.
2. **С подтверждением (надежный)** — каждый кадр подтверждается с помощью ACK.
3. **С установлением соединения** — перед передачей выполняется handshake, затем осуществляется обмен данными.

В современном интернете чаще всего используется ненадежный канальный уровень, а надежность обеспечивается на более высоком уровне протоколом TCP.

Формирование кадров (Framing)

Физический уровень представляет данные как непрерывный поток битов. Задача канального уровня — определить, где начинается и где заканчивается каждый кадр. Без решения этой проблемы невозможно корректно интерпретировать пакеты.
Существует несколько подходов к формированию кадров:

Подсчет длины

В заголовок кадра включается поле длины данных. Недостаток: ошибка в этом поле делает невозможным корректное восстановление границ всех последующих кадров.

Специальные флаги (фрейм-делимитеры)

Используются специальные байты-ограничители (флаги) для маркировки начала и конца кадра. Пример: протокол HDLC.

Byte stuffing (символьное экранирование)

Если внутри данных встречается байт-ограничитель (флаг), он экранируется специальным управляющим символом (ESC). Получатель, обнаружив ESC, интерпретирует следующий байт как данные, а не как разделитель.

Bit stuffing (битовое экранирование)

Используется на уровне битов. После N подряд идущих единиц вставляется нулевой бит. Получатель удаляет вставленные биты, восстанавливая исходный поток. Это позволяет избежать ложных границ кадров и делает протокол независимым от содержимого данных.

Обнаружение и исправление ошибок

Физическая среда передачи (кабель, радиоэфир) подвержена шумам, помехам и затуханию, что неизбежно приводит к ошибкам.

Обнаружение ошибок

Parity bit (бит четности) — простейший метод, но обладает низкой надежностью.

CRC (Cyclic Redundancy Check) — основной метод обнаружения ошибок в реальных сетях. Данные рассматриваются как многочлен над полем GF(2), где биты выступают коэффициентами. Процесс CRC включает следующие шаги:

1. К данным D добавляется r нулей (r — степень генераторного полинома G).
2. Выполняется деление на генераторный полином G.
3. Остаток от деления является CRC-кодом.

На приемной стороне принятые данные делятся на тот же полином G. Если остаток равен нулю, ошибок с высокой вероятностью нет.
В Ethernet используется CRC-32 с полиномом:

$$x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1$$

CRC-32 обеспечивает очень высокую надежность обнаружения ошибок, включая одиночные ошибки, пакетные ошибки (burst errors) и многие комбинации множественных ошибок.

Исправление ошибок

FEC (Forward Error Correction) — метод, при котором к данным добавляется избыточность, позволяющая исправить ошибки без повторной передачи. Используется в спутниковой и мобильной связи.

Код Хэмминга — позволяет не только обнаружить, но и найти позицию ошибки. Контрольные биты размещаются среди битов данных таким образом, чтобы проверять различные комбинации позиций. На приеме вычисляется «синдром», указывающий на конкретный бит с ошибкой. Базовая версия исправляет только одну ошибку; для исправления множественных ошибок применяются более сложные коды, такие как коды Рида-Соломона.

Interleaving (перемежение) — метод борьбы с пакетными ошибками. Данные перемешиваются перед передачей (например, вместо A B C D / E F G H передается A E B F C G D H). Пакетная ошибка превращается в набор одиночных ошибок, которые легче исправить. Используется в Wi-Fi, мобильных и спутниковых сетях.

Управление потоком (Flow Control)

Проблема возникает, когда отправитель работает быстрее получателя. Для решения применяются два основных подхода:

- **Stop-and-Wait** — отправитель передает один кадр и ожидает подтверждения (ACK). Простота реализации, но низкая эффективность.
- **Sliding Window (скользящее окно)** — отправитель может передавать несколько кадров подряд без ожидания ACK для каждого. Это значительно повышает эффективность использования канала.

Надежная передача: ARQ

ARQ (Automatic Repeat reQuest) — механизм автоматического запроса повторной передачи в случае потери кадра.

Stop-and-Wait ARQ

Простой алгоритм: отправил кадр → жди ACK → при тайм-ауте повтори. Канал простаивает большую часть времени.

Go-Back-N

Отправитель может передавать N кадров подряд. При обнаружении ошибки или потере кадра повторно передаются все кадры, начиная с потерянного. Пример: отправлены кадры 1 2 3 4 5, ошибка в кадре 3 → повторная передача 3 4 5.

Selective Repeat

Повторно передаются только потерянные кадры. Требует буферизации на стороне получателя и более сложной логики, но обеспечивает более высокую эффективность по сравнению с Go-Back-N.

Повышение эффективности передачи

Ожидание ACK снижает эффективность использования канала. Ключевая метрика:

$$\text{Utilization} \approx \text{Transmission Time} / \text{Transmission Time} + \text{RTT}$$

Для Stop-and-Wait эффективность равна:

$$\text{Efficiency} = T_t / T_t + 2T_p = 1 / 1 + 2a$$

где T_t — время передачи кадра, T_p — время распространения сигнала, $a = T_p / T_t$.

При $T_p \gg T_t$ (например, в спутниковых каналах) эффективность стремится к нулю.

Sliding Window решает эту проблему. При размере окна W:

$$\text{Efficiency} = \min(1, W / 1 + 2a)$$

Для полной загрузки канала требуется:

$$W \geq 1 + 2a$$

Это правило используется в TCP как BDP (Bandwidth-Delay Product).

Sequence Numbers и ограничения

Номера последовательности имеют конечный размер (например, k бит): 0 ... 2k - 1. При слишком быстром зацикливании номеров получатель может спутать новый кадр со старым дубликатом.
Условия корректности:

- Для Go-Back-N: $W \leq 2^k - 1$
- Для Selective Repeat: $W \leq 2^k - 1$

Для Selective Repeat условие строже, поскольку получатель принимает кадры вне порядка и хранит их в буфере, что повышает риск путаницы.

Таймеры и неоднозначность ретрансляции

При потере ACK отправитель выполняет ретрансляцию. Получатель может получить как оригинал, так и повтор. Sequence numbers позволяют получателю отбрасывать дубликаты, но ACK все равно отправляется. Это фундаментальный принцип надежной передачи.

Piggybacking

При двунаправленной передаче данные и ACK могут быть объединены в одном кадре, что уменьшает количество кадров и служебные накладные расходы. Недостаток: возможна задержка ACK в ожидании данных.

Кумулятивные и селективные ACK

Кумулятивные ACK (Go-Back-N) — ACK = 5 означает, что все кадры до 5 получены.
Селективные ACK (SACK) (Selective Repeat) — позволяют указать, какие именно кадры получены (например, ACK: 1, 2, 4, 5 → кадр 3 потерян). Этот механизм напрямую перешел в TCP SACK.

Подуровень MAC (Media Access Control)

Подуровень MAC решает ключевую проблему: как несколько устройств делят один канал передачи.

ALOHA

Протокол случайного доступа: устройство отправляет данные, когда хочет, при коллизии выполняет повторную передачу.

Pure ALOHA: пропускная способность S=Ge^(−20), максимум ≈ 18%.
Slotted ALOHA: S = Ge^{-G}, максимум ≈ 37%.

Случайный доступ принципиально неэффективен по сравнению с детерминированными методами.

CSMA/CD (Ethernet)

Carrier Sense Multiple Access with Collision Detection — протокол, используемый в классическом Ethernet.
Алгоритм:

1. Прослушивание канала.
2. Если канал свободен — передача.
3. Во время передачи контроль коллизий.
4. При обнаружении коллизии — немедленное прекращение передачи, ожидание случайного времени (экспоненциальный backoff) и повтор.

Экспоненциальный backoff: после каждой коллизии диапазон случайной задержки увеличивается, что снижает вероятность повторных коллизий.

Минимальный размер кадра Ethernet

В CSMA/CD необходимо обнаруживать коллизию во время передачи. Передача кадра должна длиться дольше, чем 2x время распространения сигнала. Поэтому минимальный размер кадра Ethernet составляет **64 байта**. Если кадр короче, коллизия может произойти

после окончания передачи, и ее невозможно будет обнаружить. Именно поэтому поле данных должно содержать не менее 46 байт (14 байт заголовка + 46 байт данных + 4 байта CRC = 64 байта).

CSMA/CA (Wi-Fi)

Collision Avoidance вместо Detection. В беспроводных сетях невозможно одновременно слушать и передавать (из-за радиочастотных ограничений). Алгоритм: прослушивание канала → ожидание случайного времени → передача → ожидание ACK от получателя. ACK заменяет обнаружение коллизий.

Проблема скрытых узлов (Hidden Terminal Problem)

Классическая проблема Wi-Fi: станции A и C не слышат друг друга, но обе слышат станцию B. A и C могут одновременно передавать данные на B, вызывая коллизию.
Решение: механизм RTS/CTS (Request to Send / Clear to Send):

1. A → RTS (запрос на передачу).
2. B → CTS (разрешение на передачу).
3. A передает данные.

Все станции, слышащие CTS, «замолкают», предотвращая коллизии.

Детерминированный vs случайный доступ

Детерминированный доступ (TDMA, FDMA, CDMA) — заранее известно, кто и когда передает. Обеспечивает предсказуемость, но менее гибок.
Случайный доступ (Ethernet) — кто успел, тот и отправил. Обеспечивает гибкость, но менее предсказуем.
TDMA (Time Division Multiple Access) — разделение по времени.
FDMA (Frequency Division Multiple Access) — разделение по частоте.
CDMA (Code Division Multiple Access) — разделение по кодам.

Ethernet — реальный канальный уровень

Формат кадра Ethernet

Современный Ethernet-кадр (IEEE 802.3) имеет следующую структуру:

Поле	Размер	Назначение
Preamble	7 байт	Синхронизация
SFD	1 байт	Начало кадра (последовательность 10101011)
Destination MAC	6 байт	MAC-адрес получателя
Source MAC	6 байт	MAC-адрес отправителя

Поле	Размер	Назначение
Type / Length	2 байта	Тип протокола (например, IP) или длина данных
Data	46–1500 байт	Полезные данные
CRC (FCS)	4 байта	Контрольная сумма

Минимальный размер кадра — 64 байта, максимальный — 1518 байт (без VLAN-тега).

Особенности Ethernet

- Использует MAC-адреса.
- Работает без подтверждений (ненадежный).
- Надежность обеспечивается на более высоких уровнях (TCP).

Ethernet — это ненадежный, но быстрый канальный протокол.

Современный Ethernet: отказ от CSMA/CD

В современных сетях Ethernet работает в полнодуплексном режиме через коммутаторы. Каждый порт коммутатора представляет собой отдельный канал, поэтому разделяемая среда отсутствует, и коллизии полностью исключены. CSMA/CD в таких условиях не используется.
На сегодняшний день стандарты Ethernet достигли скоростей 400 Гбит/с и 800 Гбит/с (IEEE 802.3df-2024).

Wi-Fi (IEEE 802.11)

Wi-Fi принципиально отличается от проводного Ethernet:

Характеристика	Ethernet	Wi-Fi
Среда	Проводная	Беспроводная
Обнаружение коллизий	Есть (CSMA/CD)	Нет (CSMA/CA)
Полудуплекс/Полнодуплекс	Полнодуплекс (современный)	Полудуплекс

Wi-Fi сложнее Ethernet по ряду причин:

- Невозможность одновременной передачи и приема.
- Помехи и перекрытие каналов.
- Проблема скрытых узлов.
- Переменная мощность сигнала.

Поэтому Wi-Fi использует CSMA/CA, ACK на каждый кадр, RTS/CTS и адаптацию скорости передачи.

Rate Adaptation (адаптация скорости)

Скорость передачи в Wi-Fi динамически изменяется в зависимости от качества сигнала: при хорошем сигнале скорость высокая, при ухудшении — снижается. Это компромисс между пропускной способностью и надежностью.

Wi-Fi 7 (IEEE 802.11be)

Новейший стандарт Wi-Fi 7 (802.11be) предлагает следующие улучшения:

- Каналы шириной до 320 МГц (вдвое больше, чем в Wi-Fi 6).
- Модуляция до 4096-QAM.
- До 16 пространственных потоков.
- Multi-Link Operation (MLO).
- Максимальная теоретическая скорость до 46 Гбит/с.

Коммутация на канальном уровне

Устройства L2

Повторитель (Repeater) — усиливает сигнал, работая на физическом уровне.
Хаб (Hub) — рассылает принятые сигналы на все порты. Работает на физическом уровне.
Мост (Bridge) — фильтрует трафик по MAC-адресам, работая на канальном уровне.
Коммутатор (Switch) — «умный мост», хранит таблицу MAC-адресов и портов, принимает решение о пересылке кадров на основе MAC-адреса назначения.

Внутренняя логика коммутатора

Коммутатор хранит MAC-таблицу, сопоставляющую MAC-адреса с портами. Алгоритм работы:

1. Получение кадра.
2. Проверка MAC-адреса назначения.
3. Если адрес известен — отправка на соответствующий порт.
4. Если адрес неизвестен — широковещательная рассылка (flood) на все порты.

Коммутатор обучается автоматически, анализируя source MAC-адреса входящих кадров. Это делает сеть самоорганизующейся.
Архитектура коммутатора включает входные порты, выходные порты и коммутационную матрицу (switching fabric).

Проблемы коммутации

Head-of-Line Blocking — если один выходной порт занят, кадры, ожидающие этого порта, блокируют кадры, направляемые на другие порты. Решение: буферизация на выходе и crossbar-коммутация.

Broadcast Storm — широковещательные кадры (MAC FF:FF:FF:FF:FF:FF) рассылаются на все порты. При наличии петель в топологии возникает бесконечное размножение кадров, перегрузка сети и падение L2-сегмента. Это одна из самых опасных ошибок в сетях.

Spanning Tree Protocol (STP)

STP решает проблему петель в сетях с избыточными соединениями. Алгоритм:

1. Выбор корневого моста (Root Bridge) по минимальному Bridge ID.
2. Для каждого коммутатора выбор пути к корню с минимальной стоимостью.
3. Блокировка избыточных портов.

Состояния портов STP: blocking, listening, learning, forwarding. Это распределенный алгоритм без центрального управления.

VLAN (Virtual LAN)

VLAN позволяет логически разделить одну физическую сеть на несколько изолированных сегментов. В Ethernet-кадр добавляется VLAN-тег (802.1Q):

```
[MAC][MAC][VLAN TAG][TYPE][DATA]
```

VLAN-тег содержит 12-битный VLAN ID и 3-битное поле приоритета (PCP — Priority Code Point). Приоритеты варьируются от 0 (низший) до 7 (высший), что позволяет реализовать QoS на канальном уровне.
VLAN обеспечивает изоляцию трафика, безопасность и сегментацию сети. Пример: VLAN 10 — разработка, VLAN 20 — бухгалтерия.

QoS на канальном уровне

Не все данные одинаково важны: VoIP чувствителен к задержке, файловые передачи — нет. QoS на L2 реализуется через приоритетные очереди, расписание (scheduling) и приоритетные кадры (802.1p). Это влияет на задержку (latency) и вариацию задержки (jitter).

MAC-адреса

Определение и формат

MAC-адрес (Media Access Control address) — уникальный идентификатор сетевого интерфейса (NIC), используемый на канальном уровне. Обычно имеет длину 48 бит и записывается в виде шестнадцатеричных групп: `00:1A:2B:3C:4D:5E`.

Структура:

- Первые 24 бита — OUI (Organizationally Unique Identifier), идентифицирующий производителя.
- Вторые 24 бита — уникальный номер, назначаемый производителем.

Назначение и область действия

MAC-адрес решает строго локальную задачу доставки между соседними узлами в одном сегменте сети. Он используется внутри Ethernet-кадра:

```
[Dest MAC][Src MAC][Data]
```

При передаче данных:
1. Формируется IP-пакет (L3).
2. Он упаковывается в Ethernet-кадр (L2).
3. В кадр добавляются MAC-адреса отправителя и получателя.

MAC vs IP

Характеристика	MAC	IP
Уровень	L2	L3
Область	Локальная сеть	Глобальная
Назначение	Доставка кадра	Маршрутизация
Изменяется	На каждом хопе	Остается тем же

Критический момент: MAC-адрес меняется на каждом сетевом переходе, тогда как IP-адрес остается неизменным на протяжении всего пути. Например, при передаче от клиента к серверу через роутер на каждом шаге используется новый MAC-адрес, но тот же IP-адрес.

Специальные MAC-адреса

- **Broadcast**: `FF:FF:FF:FF:FF:FF` — отправляется всем устройствам в сегменте.
- **Multicast**: отправляется группе устройств.
- **Unicast**: отправляется конкретному устройству.

ARP (Address Resolution Protocol)

ARP связывает IP-адреса (L3) с MAC-адресами (L2). Процесс:

1. Широковещательный запрос: «Кто имеет IP X?»
2. Устройство с этим IP отвечает: «Это я, вот мой MAC».
3. Запись сохраняется в ARP-таблице.

Без ARP невозможно отправить пакет, поскольку канальный уровень требует знания MAC-адреса следующего hop'а.

Безопасность L2

Канальный уровень уязвим для ряда атак:

MAC flooding — переполнение MAC-таблицы коммутатора, приводящее к его работе в режиме хаба (рассылка на все порты).
ARP spoofing — подмена MAC-адреса в ARP-ответах, позволяющая злоумышленнику перехватывать трафик.

Методы защиты:

- Port Security — ограничение количества MAC-адресов на порту.
- Dynamic ARP Inspection (DAI) — проверка ARP-пакетов на соответствие.
- VLAN filtering — ограничение меж-VLAN трафика.
- Gateway-controlled ARP (GCA).

VLAN hopping — атака, позволяющая переходить между VLAN.

LLC (Logical Link Control)

LLC — подуровень канального уровня, обеспечивающий интерфейс к сетевому уровню и мультиплексирование протоколов. Ethernet сам по себе не знает, какой протокол находится внутри кадра (IP, ARP и т.д.). LLC помогает различать эти протоколы.

Конкретные протоколы канального уровня

HDLC (High-Level Data Link Control)

Бито-ориентированный протокол, использующий bit stuffing. Поддерживает надежную передачу и управление потоком. Типы кадров:

- I-frame (данные).
- S-frame (служебные).
- U-frame (управление).

PPP (Point-to-Point Protocol)

Используется в прямых соединениях (модем, VPN-туннели). Особенности:
- Framing (как в HDLC).
- Аутентификация (PAP, CHAP).
- Мультипротокольность.

PPP — это аналог Ethernet для point-to-point соединений.

Сетевая карта (NIC) — реализация канального уровня в железе

Сетевая карта (Network Interface Card, NIC) — устройство, реализующее канальный и частично физический уровень в аппаратуре.

Передающий тракт (Transmit path)

1. **Получение данных от ОС** — драйвер передает payload (например, IP-пакет) и иногда уже готовый заголовок L2.
2. **Формирование кадра** — NIC добавляет MAC-адреса отправителя и получателя, EtherType.
3. **CRC** — вычисляется аппаратно и добавляется в конец кадра.
4. **Буферизация** — кадр помещается в TX-буфер.
5. **Доступ к среде (MAC)** — в зависимости от технологии (CSMA/CD для старого Ethernet, CSMA/CA для Wi-Fi, или просто отправка в полнодуплексном режиме).
6. **Передача в физический уровень** — кадр преобразуется в битовый поток и сигнал.

Приемный тракт (Receive path)

1. **Прием сигнала** — преобразование в биты.
2. **Сборка кадра** — определение границ (preamble, SFD).
3. **Проверка CRC** — при ошибке кадр отбрасывается.
4. **Фильтрация MAC** — NIC проверяет, совпадает ли MAC назначения с ее MAC, или это broadcast/multicast. Если нет — кадр отбрасывается на уровне железа, CPU даже не узнает о нем.
5. **Передача в ОС** — данные помещаются в RX-буфер, генерируется прерывание (или используется polling), драйвер передает данные в сетевой стек.

Внутренняя архитектура NIC

```
[ Host (CPU) ]
        ↓
[ Driver ]
        ↓
[ NIC ]
   ├── MAC unit
   ├── PHY unit
   ├── Buffers (TX/RX)
   ├── DMA engine
```

MAC unit — ключевая часть, реализующая канальный уровень в железе: framing, MAC-адреса, CRC, фильтрацию, flow control.
PHY (Physical Layer) — отвечает за кодирование сигналов, напряжение/радиосигнал, синхронизацию.
Буферы (TX/RX) — очереди на отправку и прием. NIC работает асинхронно относительно CPU.
DMA (Direct Memory Access) — позволяет NIC читать и писать в память без участия CPU, что критично для производительности. Без DMA CPU вынужден копировать данные вручную, с DMA NIC сама помещает данные в RAM.

Interrupt / Polling — NIC может генерировать прерывание при наличии данных, либо CPU может опрашивать NIC. Современные системы используют гибридные подходы (NAPI в Linux).

Offloading (разгрузка CPU)

Современные NIC выполняют задачи, выходящие за рамки L2:

- **Checksum offload** — NIC вычисляет контрольные суммы для TCP/IP.
- **TSO (TCP Segmentation Offload)** — NIC самостоятельно разбивает большие данные на сегменты MTU.
- **LRO (Large Receive Offload)** — объединение нескольких входящих сегментов одного потока.

Эти механизмы значительно снижают нагрузку на CPU и повышают пропускную способность.

Почему канальный уровень реализован в железе

1. **Скорость** — 1–100 Гбит/с, CPU не справится с такой нагрузкой.
2. **Параллельность** — NIC работает независимо от CPU.
3. **Снижение нагрузки на CPU** — освобождение ресурсов для прикладных задач.

Сетевая карта — это не просто «переходник», а полноценный процессор канального уровня, формирующий кадры, проверяющий ошибки, фильтрующий трафик и управляющий передачей.

Модели сервиса канального уровня

Канальный уровень может предоставлять три типа сервиса:

1. **Unacknowledged connectionless service** — без соединения, без подтверждений. Пример: Ethernet.
2. **Acknowledged connectionless service** — без соединения, но с ACK. Используется в шумных каналах (беспроводных).
3. **Acknowledged connection-oriented service** — установка соединения, гарантированная доставка.

Канальный уровень может быть как надежным, так и ненадежным — это дизайн-выбор.

Классификация ошибок

- **Одиночные ошибки (single-bit error)** — один бит искажен.
- **Пакетные ошибки (burst error)** — несколько бит подряд искажены.

CRC хорошо обнаруживает пакетные ошибки, код Хэмминга — одиночные. Поэтому на практике часто используются комбинации методов.

Bandwidth-Delay Product

Произведение пропускной способности на задержку определяет «объем канала» — сколько данных может находиться в полете. Это напрямую связано с размером скользящего окна и эффективностью протокола.

Итоговое обобщение

Канальный уровень решает три фундаментальные проблемы:

1. **Где границы данных?** → Framing.
2. **Что делать с ошибками?** → CRC + ARQ.
3. **Как делить канал?** → MAC-протоколы.

Полный набор функций канального уровня включает:

- Передачу данных (framing, error detection/correction).
- Надежность (ARQ, sliding window).
- Доступ к среде (CSMA/CD, CSMA/CA).
- Адресацию (MAC).
- Коммутацию (switch, MAC-таблицы).
- Реальные протоколы (Ethernet, Wi-Fi, PPP, HDLC).
- Расширения (VLAN, QoS).
- Связь с сетевым уровнем (ARP, LLC).

Ключевые инварианты

1. **Надежность не гарантируется каналом полностью** — она может отсутствовать (Ethernet), быть частичной (Wi-Fi) или усиливаться на верхних уровнях (TCP).
2. **Ошибки неизбежны** — поэтому необходимы detection (CRC), correction (FEC) и retransmission (ARQ).
3. **Общий канал требует координации** — MAC-протоколы решают конфликт доступа.
4. **Производительность ограничена** RTT, размером окна и вероятностью коллизий.

Глубокое понимание

Канальный уровень работает только между соседними узлами, не знает о маршрутах, оперирует MAC-адресами. Он НЕ решает глобальную доставку, маршрутизацию и end-to-end надежность. Поэтому Ethernet ненадежен, а TCP делает надежность выше.
Канальный уровень — это компромисс между:

- Надежностью (ARQ, CRC).
- Задержкой (большие окна → быстрее, но сложнее контроль).
- Пропускной способностью (MAC-протоколы, борьба с коллизиями).
- Сложностью (Stop-and-Wait простой, Selective Repeat сложный).

Канальный уровень как локальная оптимизация

Канальный уровень решает задачи эффективности передачи, локальной надежности и доступа к среде. Но он не гарантирует end-to-end доставку, отсутствие потерь и порядок на уровне всей сети. Поэтому TCP все равно необходим, даже если L2 «надежный».

Связь канального и транспортного уровней

Многие механизмы канального уровня (скользящее окно, sequence numbers, ARQ) напрямую переходят в TCP. Понимание L2 является фундаментом для понимания TCP-контроля перегрузок, управления потоком и надежной доставки.

Канальный уровень превращает хаотичную физическую среду в структурированную, управляемую и разделяемую сеть. Это не просто «про передачу» — это система, которая создает основу для всех вышележащих уровней сетевой архитектуры.