КЛЮЧЕВОЕ РАЗЛИЧИЕ

НАДЁЖНЫЙ ПОТОК VS. БЫСТРАЯ ПЕРЕДАЧА

Одна задача. Противоположные подходы.

TCP

Протокол управления передачей

СОЕДИНЕНИЕ

Трёхстороннее рукопожатие

ДОСТАВКА

Гарантированная, по порядку

ПОТЕРИ

Обнаруживает и повторно отправляет

ЗАГОЛОВОК

20 байт заголовок
+ состояние

UDP

Протокол пользовательских датаграмм

СОЕДИНЕНИЕ

Нет соединения
просто отправка

ДОСТАВКА

Максимум усилий,
неупорядоченная

ПОТЕРИ

Потерянный пакет
потерян навсегда

ЗАГОЛОВОК

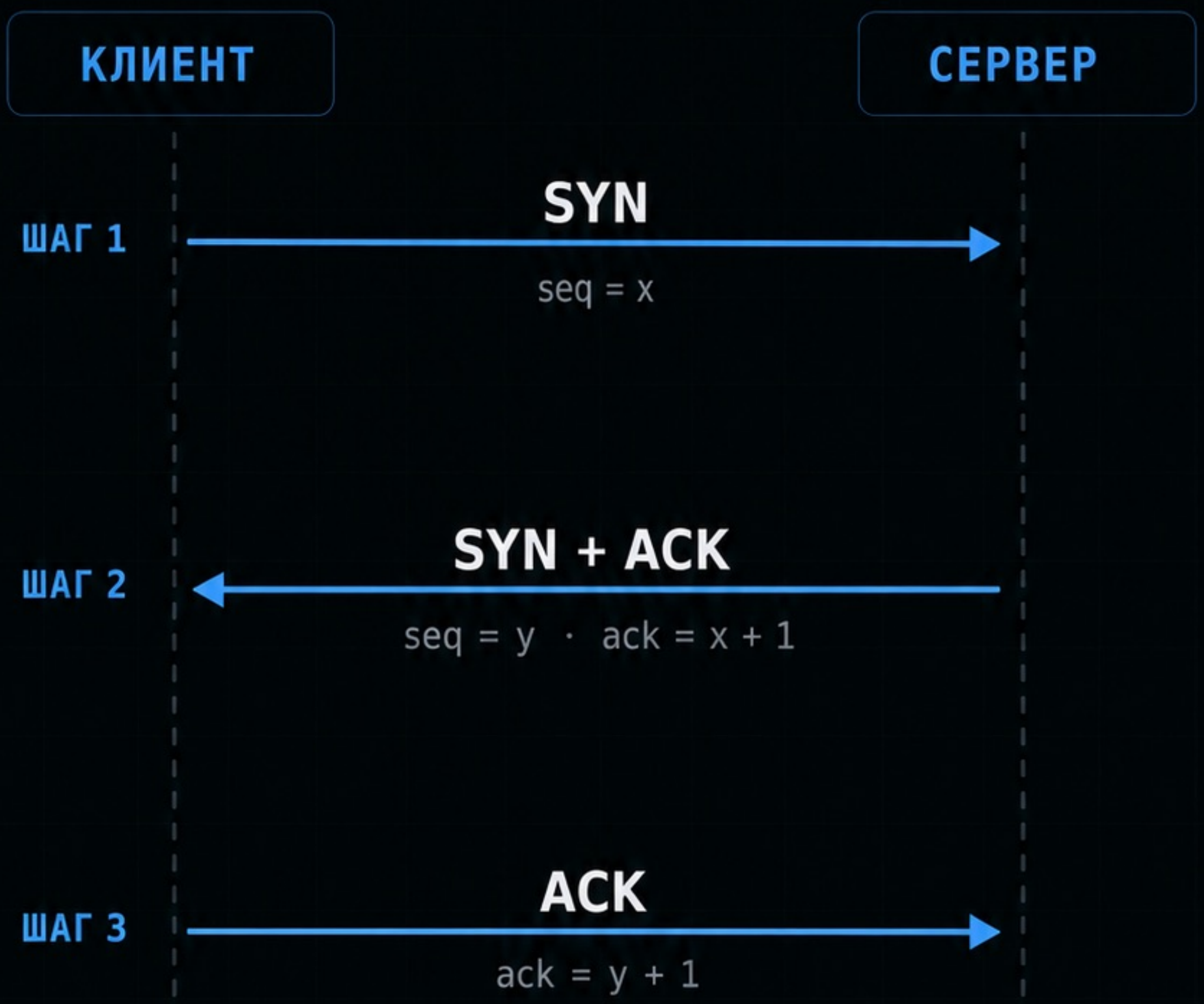
8 байт заголовок,
без состояния

TCP — гарантирует корректность.
UDP — гарантирует скорость.

КАК РАБОТАЕТ TCP

ТРЁХСТОРОННЕЕ РУКОПОЖАТИЕ

Как TCP открывает каждое соединение.

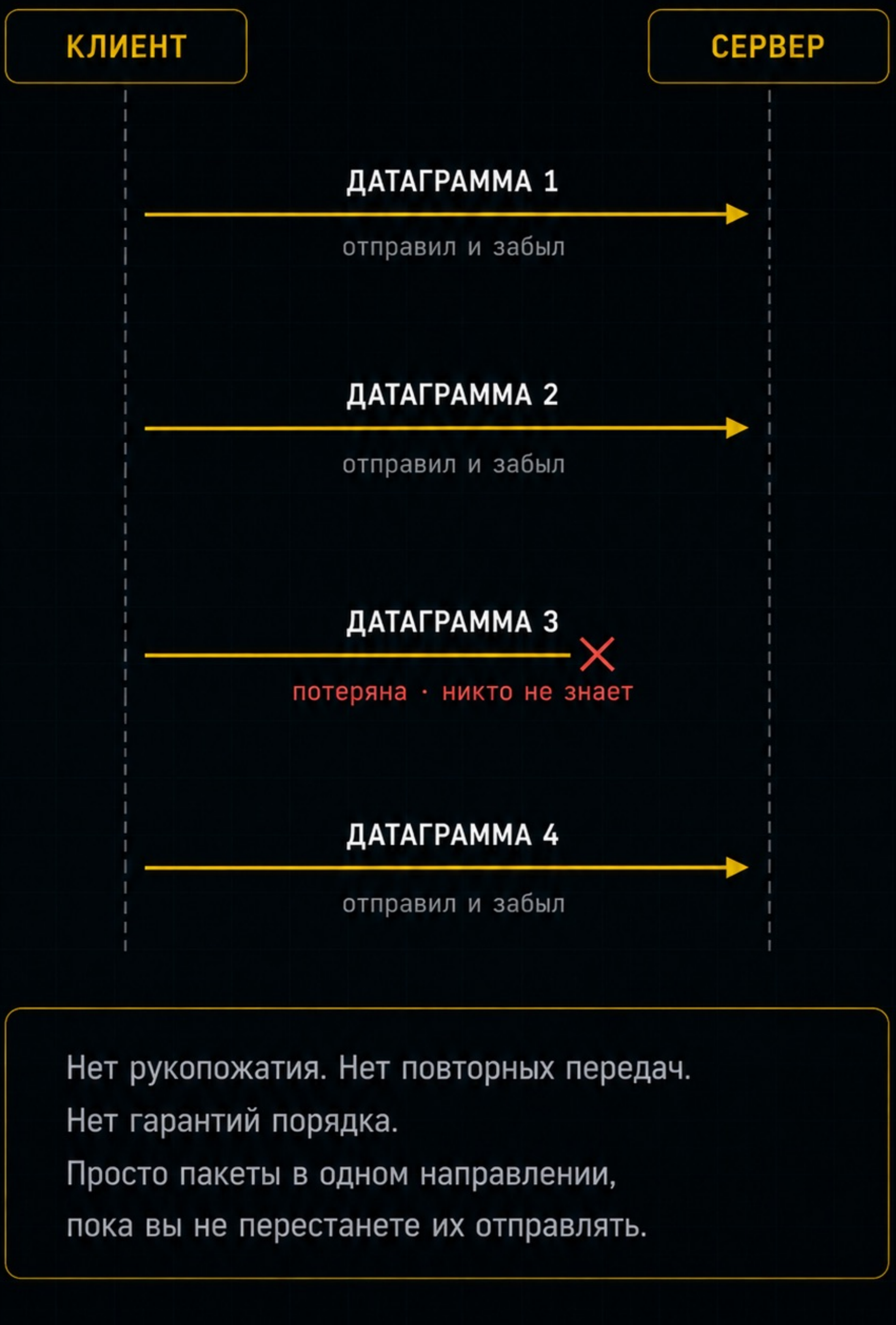


Теперь обе стороны знают, что другая достигаема, синхронизированы и готовы обмениваться упорядоченными, надёжными данными.

КАК РАБОТАЕТ UDP

ОТПРАВИЛ И ЗАБЫЛ

Как UDP отправляет данные.



Нет рукопожатия. Нет повторных передач.
Нет гарантий порядка.
Просто пакеты в одном направлении,
пока вы не перестанете их отправлять.

АНАТОМИЯ ЗАГОЛОВКА

ЧТО В КАЖДОМ ПАКЕТЕ

Метаданные обоих протоколов отправляются перед вашими данными.

ЗАГОЛОВОК TCP

20 байт

Порт источника	Порт назначения
Номер последовательности	
Номер подтверждения	
Смещение / Резерв / Флаги	Размер окна
Контрольная сумма	Указатель срочности

ЗАГОЛОВОК UDP

8 байт

Порт источника	Порт назначения
Длина	Контрольная сумма

> TCP несёт **12 дополнительных байт** на пакет перед тем, как даже начинаются ваши данные.

НАДЁЖНОСТЬ

КОГДА ПАКЕТ ТЕРЯЕТСЯ

TCP восстанавливает. UDP даже не пытается.

TCP

Вы получаете именно то, что было отправлено.

- ✓ Обнаруживает потери по пропускам в номерах последовательности
- ✓ Повторно отправляет пропущенные сегменты автоматически
- ✓ Переупорядочивает пакеты, пришедшие не по порядку
- ✓ Удаляет дубликаты
- ✓ Проверяет целостность по контрольной сумме 16 бит

UDP

Вы получаете то, что дошло.

- ✗ Потерянный пакет — потерян навсегда
- ✗ Нет упорядочивания — пакеты могут прийти в любом порядке
- ✗ Нет обнаружения дубликатов
- ✗ Нет контрольной суммы (опционально в IPv4)
- ✗ Ваше приложение должно решать всё само

Надёжность — не бесплатна.
Каждая повторная передача стоит времени и ресурсов.

СКОРОСТЬ И НАГРУЗКА

СТОИМОСТЬ ЗА ПАКЕТ

То, что вы платите, прежде чем начнут передаваться данные.

Задержка установки

Круговые поездки до первого байта данных.

TCP	UDP
1 RTT	0 RTT

Накладные расходы заголовка

Байты добавляются к каждому пакету перед полезной нагрузкой.

TCP	UDP
20 байт	8 байт

Состояние соединения

Память ядра отслеживает каждого одноранговой узла.

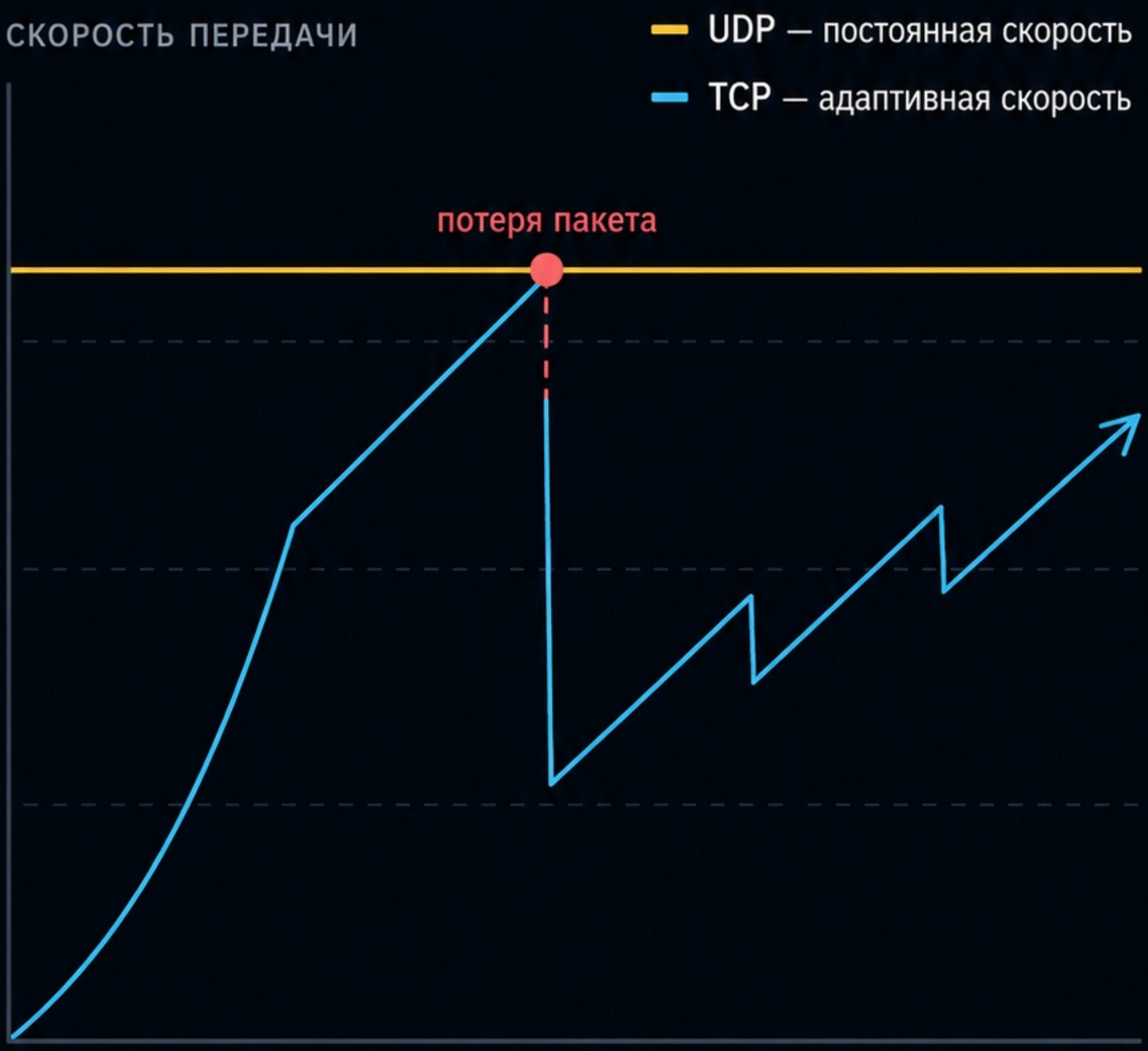
TCP	UDP
~600 Б	0 Б

TCP стоит около 12 дополнительных байт в каждом пакете. Платите заранее — экономьте потом.

ПОТОК И ПЕРЕГРУЗКА

АДАПТАЦИЯ К СЕТИ

TCP снижает скорость при потере пакетов.
UDP — нет.



TCP поддерживает здоровье сети,
автоматически снижая скорость при потере пакетов.

UDP не снижает скорость —
поэтому работает быстрее.

КОГДА TCP ПОБЕЖДАЕТ

ИСПОЛЬЗУЙ TCP, КОГДА...

Один потерянный байт может сломать приложение.

HTTPS / HTTP-2

Веб-трафик, API,
REST-сервисы

SSH

Интерактивные оболочки,
туннели

SMTP · IMAP

Передача почты,
синхронизация ящиков

FTP · SFTP

Передача файлов
с проверкой целостности

PostgreSQL · MySQL

Подключение
к базам данных

Git over HTTPS

Управление исходным
кодом, реестры пакетов

Важен каждый байт.
Выбирайте TCP, когда важны целостность,
порядок и доставка.

КОГДА UDP ПОБЕЖДАЕТ

ИСПОЛЬЗУЙ UDP, КОГДА...

Опоздавшие данные хуже, чем их полное отсутствие.
UDP не гарантирует доставку — и это его сила.

DNS

Разрешение имён,
поиск поддоменов
через sub-ms

DHCP

Назначение IP-адреса
при загрузке системы

NTP

Синхронизация времени,
короткие сообщения

VoIP · RTP

Голос и видео в реальном
времени

Online gaming

Обновление позиции
и состояния игроков

QUIC · HTTP/3

Современный веб
поверх UDP

UDP = скорость и простота.
Меньше накладных расходов, меньше задержек,
больше возможностей для реального времени.

РИСКИ БЕЗОПАСНОСТИ

РАЗНЫЕ АТАКИ, РАЗНЫЕ ПОВЕРХНОСТИ

Каждый протокол открывает свою дверь для злоумышленников.

TCP

Состояниевый • спуфинг сложнее.

- SYN flood • перегрузка таблицы соединений
- RST injection • принудительное разъединение
- Session hijacking • кража активных соединений в полёте
- Slowloris • удержание сокетов открытыми навсегда

UDP

Бессостояниевый • спуфинг тривиален.

- Amplification • усиление (DNS, NTP, Memcached, CLDAP)
- Source spoofing • нет рукопожатия для проверки
- Reflection • отражение ответа жертве (огромные ответы)
- Hard to filter • отсутствие состояния = сложно фильтровать

ОДИН СПУФИНГОВАННЫЙ DNS-ЗАПРОС
может вернуть в 50 раз больше байт, чем было отправлено.

60 байт → ~3000 байт

ЧТО ДЕЛАТЬ?

- Используйте ограничения скорости и квоты (rate limiting).
- Фильтруйте поддельные источники на границе сети (BCP 38).
- Включайте мониторинг аномалий и поведенческий анализ.
- Для UDP-сервисов отключайте неиспользуемые функции.
- Держите системы и прошивки в актуальном состоянии.

Безопасность — это слои защиты, а не один серебряный патч.