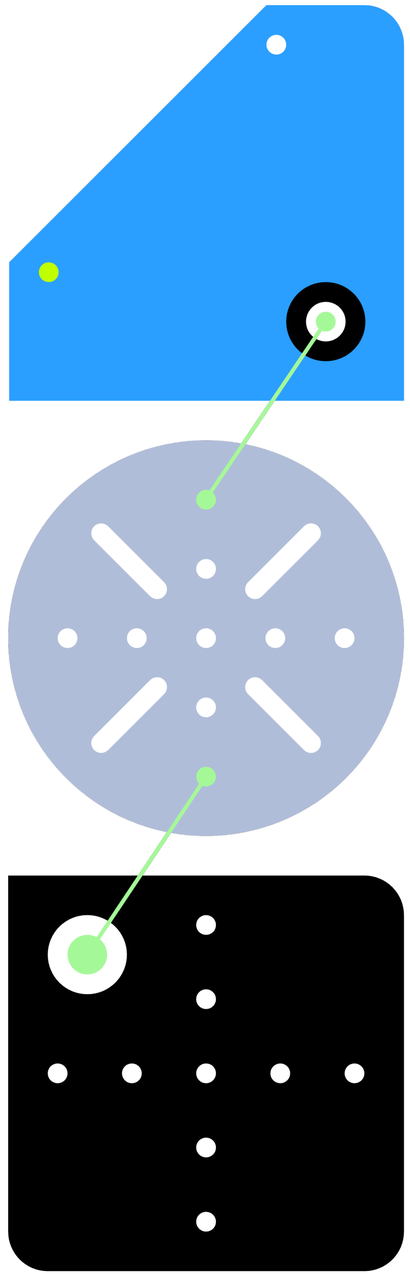
Yandex Cloud

Что происходит между Select и данными

Петр Гуринов

Команда продуктовой архитектуры,
Data Platform



Петр Гуринов

Команда продуктовой архитектуры
Data Platform, Yandex Cloud



Выпускник мехмата МГУ
и бизнес-информатики НИУ ВШЭ

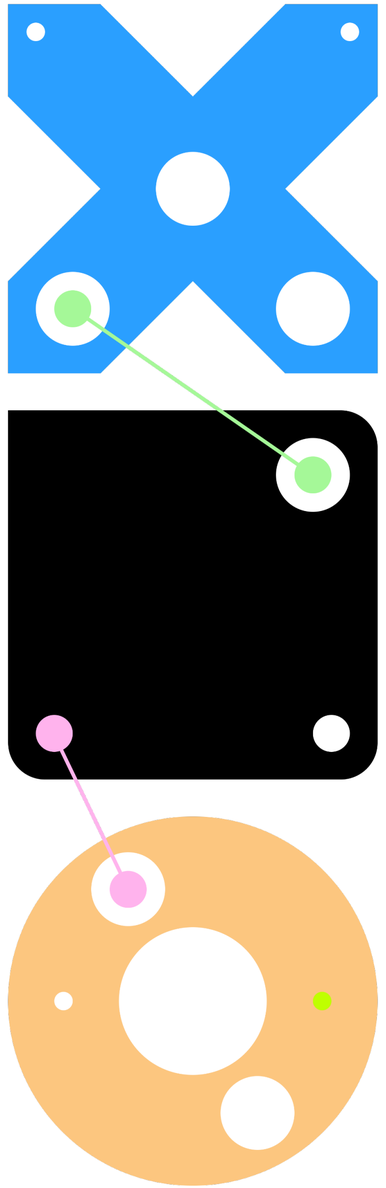
Работаю с хранилищами
данных более 5 лет

Развиваю Managed Service
for Apache Spark™ и другие
compute-движки в Data Platform

О спикере

О чём поговорим

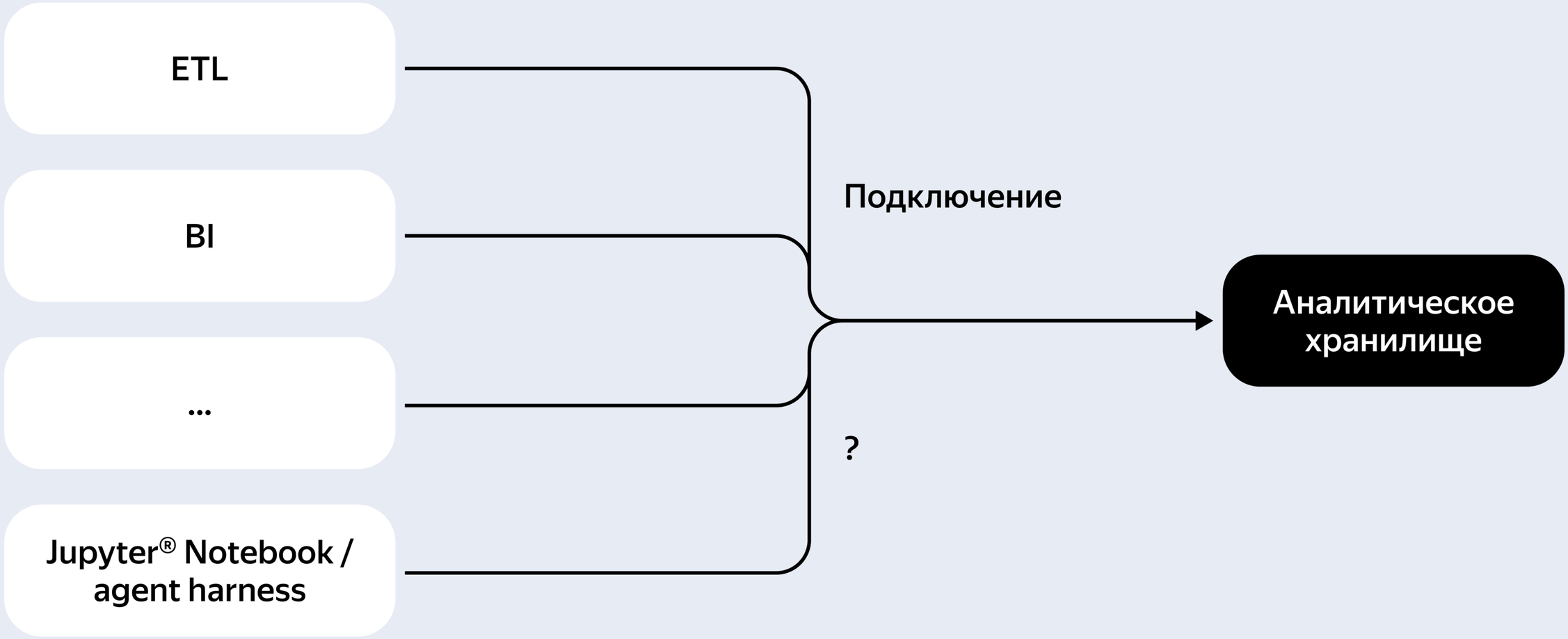
1. Введение
2. Путь данных от БД до клиента
3. Что летит по сети: Wireshark
   - В PostgreSQL
   - В ClickHouse®
4. Намерили бенчмарком
5. Итоги



Введение

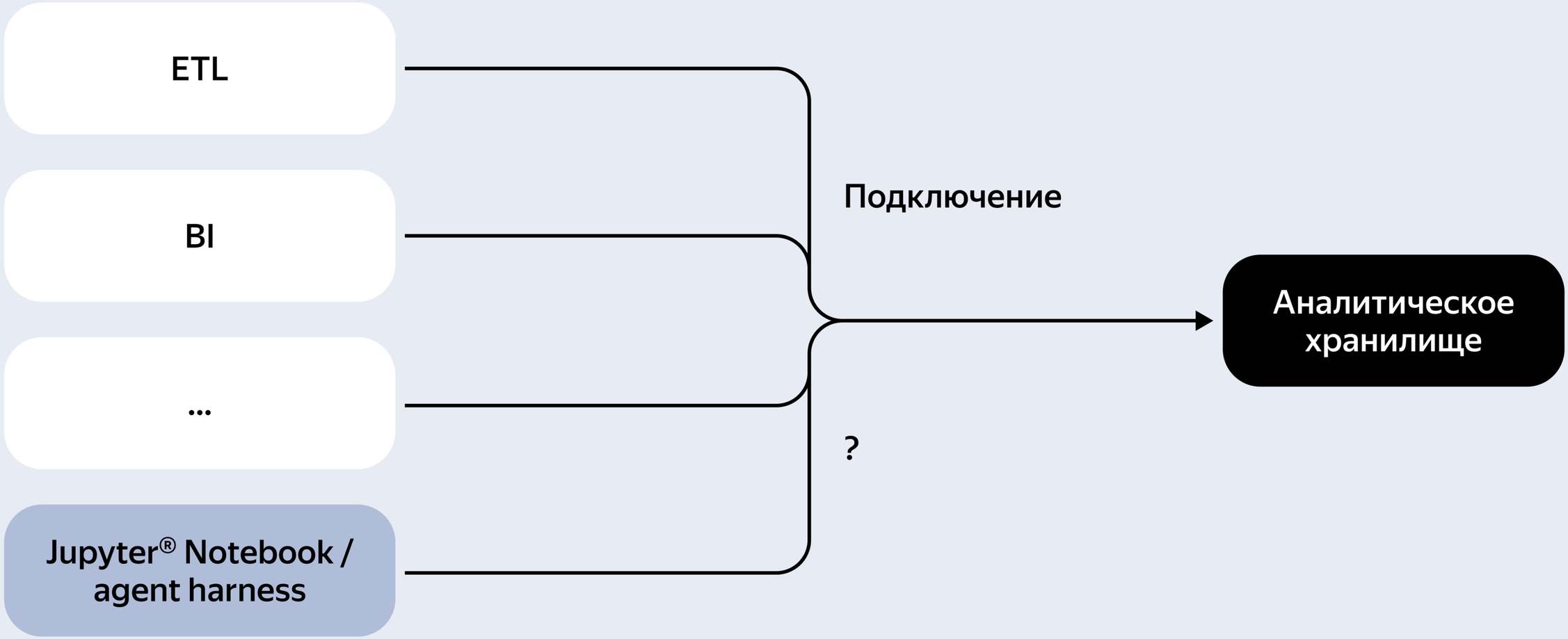
Мы проектируем компоненты

и оставляем подключение без внимания



Мы проектируем компоненты

и оставляем подключение без внимания



Архитектуру согласовали

Способ доставки — нет



Движок и формат хранения выбрали



Ресурсы кластера посчитали



Способ получения результата скопировали из документации

Архитектуру согласовали

Способ доставки — нет



Движок и формат хранения выбрали



Ресурсы кластера посчитали



Способ получения результата скопировали из документации

Хотя он определяет, какой будет:

драйвер, протокол, формат, способ сборки результата

Архитектуру согласовали

Способ доставки — нет

Движок и формат хранения выбрали



Ресурсы кластера посчитали



Способ получения результата скопировали из документации



Архитектуру согласовали

Способ доставки — нет

Движок и формат хранения выбрали



Ресурсы кластера посчитали



Способ получения результата скопировали из документации



Хотя он определяет, какой будет:

драйвер, протокол, формат, способ сборки результата

«Hold my data hostage — a case for client protocol redesign»

Способ	Время, с	Передано, ГБ	Коэф. к NetCat
Netcat	10,25	7,19	×1,0
MySQL®	101,22	7,44	×9,9
DB2	169,48	7,33	×16,5
DBMS X	189,50	6,35	×18,5
PostgreSQL	201,89	10,39	×19,7
MonetDB	209,02	8,97	×20,4
MySQL® + C	391,27	2,85	×38,2
Hive™	627,75	8,69	×61,2
MongoDB®	686,45	43,6	×67,0

×10–67

время к Netcat

Статья:
clck.ru/3WAMtb

Raasveldt, Muehleisen: Don’t hold my data hostage, Table 1. PVLDB 10(10) (2017).

PostgreSQL и ClickHouse®

PostgreSQL 18.6

- Один основной протокол pg-wire
- Много драйверов и режимов
- Построчная модель хранения и результата

ClickHouse® 26.7

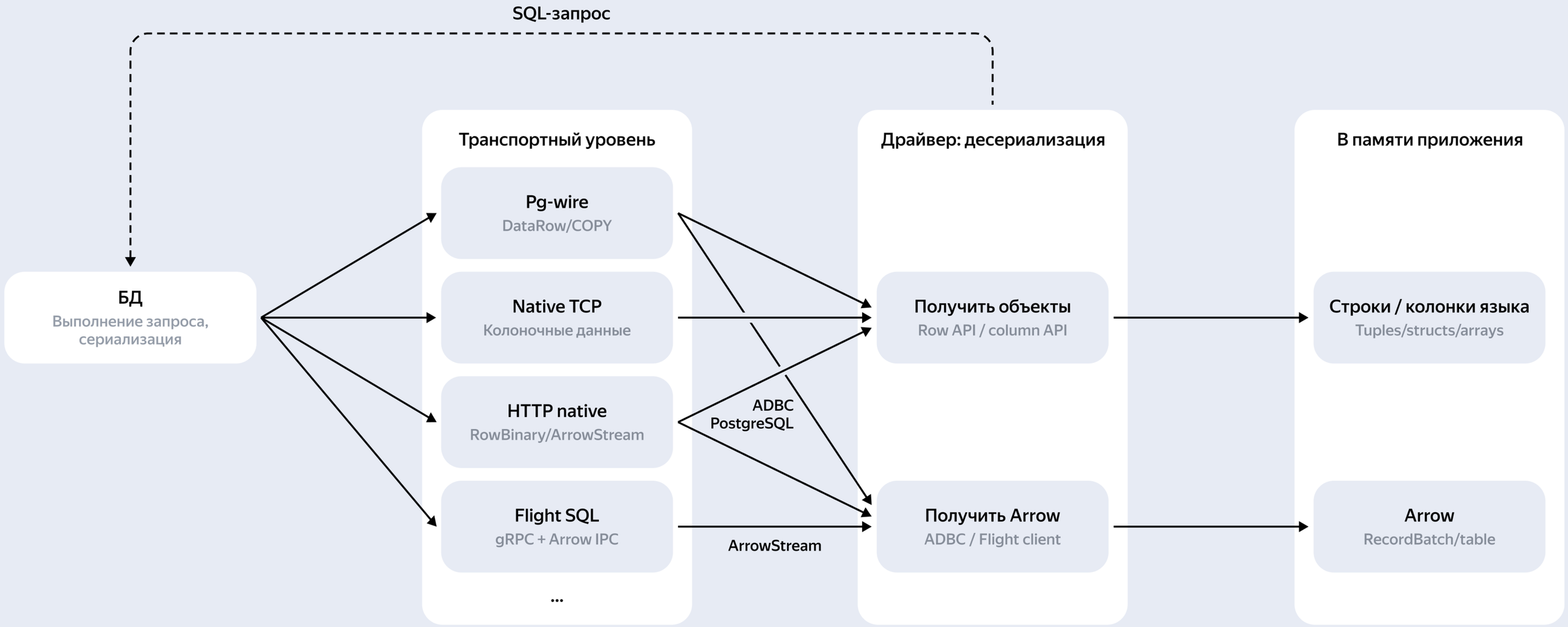
- Несколько протоколов и форматов ответа
- Колоночная модель хранения. Можно её сохранить вплоть до Arrow на клиенте

Какие решения принимает драйвер?

Где и как меняется формат данных?

Путь данных от БД до клиента

Путь запроса от БД до клиента



Одна строка определяет все уровни способа доставки

```
engine = create_engine("driver://user@host/db")
with engine.connect() as connection:
    df = pd.read_sql(sql, connection)
```

Что явно

- Клиент
- Подключение к базе
- SQL-запрос



Что скрыто

- Драйвер
- Протокол
- Формат
- Десериализация в DataFrame



Запрос ↔ результат

Участок	↓ Запрос	↑ Результат
Приложение	SQL и параметры	Структура данных или поток строк
Клиентский API	Выполнить запрос	Прочитать порцию или весь результат
Драйвер	Кодирует команды в сообщения	Декодирует и преобразует данные
Сеть	Сообщения выбранного протокола	Метаданные и данные выбранного формата
Серверный endpoint	Принимает и разбирает запрос	Сериализует результат
Движок БД	Планирует и выполняет запрос	Вычисляет значения результата

Формат ответа связывает сериализацию на сервере и декодирование в драйвере

Пять слоёв на пути данных

Слой	За что отвечает	Примеры
API	Какие вызовы доступны приложению	DB-API, JDBC, ODBC, ADBC, Go DataBase / SQL…
Драйвер	Как выполнить вызовы и собрать результат	Psycopg, pgJDBC, psqlODBC, ADBC PostgreSQL, ClickHouse®-connect…
Протокол обмена	Какие сообщения отправлять и в каком порядке	Pg-wire, ClickHouse® Native TCP, HTTP, Flight SQL…
Формат ответа	Как закодированы данные в сообщениях	Text / binary DataRow, Native, RowBinary, Arrow IPC…
Объект приложения	С чем работает код после получения данных	Rows, tuples, structs, arrays, Arrow Table, DataFrame…

API описывает вызовы. Драйвер реализует обмен и сборку результата

Arrow, ADBC, Flight и Flight SQL

	Для чего	Как используем
Arrow	Колоночное представление данных в памяти	RecordBatch, Arrow Table
Arrow IPC	Сериализация схемы и batches	Поток или файл, ClickHouse® FORMAT ArrowStream
ADBC	Клиентский API работы с БД через Arrow	Драйверы ADBC PostgreSQL и ADBC Flight SQL
Flight	RPC для передачи потоков Arrow	gRPC и Arrow IPC
Flight SQL	SQL-команды и метаданные поверх Flight	Flight SQL client и совместимый endpoint

Для Flight SQL нужна поддержка на серверной стороне

Что летит по сети

Проверим Wireshark

Для примера сгенерируем таблицу

Как участвовал в конференции

Conf varchar(32)	Starts_at timestamp	Speaker int4
SmartData	2023-09-13 11:00:00	0
SmartData	2024-09-08 10:15:00	1
SmartData	2025-10-05 10:00:00	0
SmartData	2026-09-23 10:30:00	1

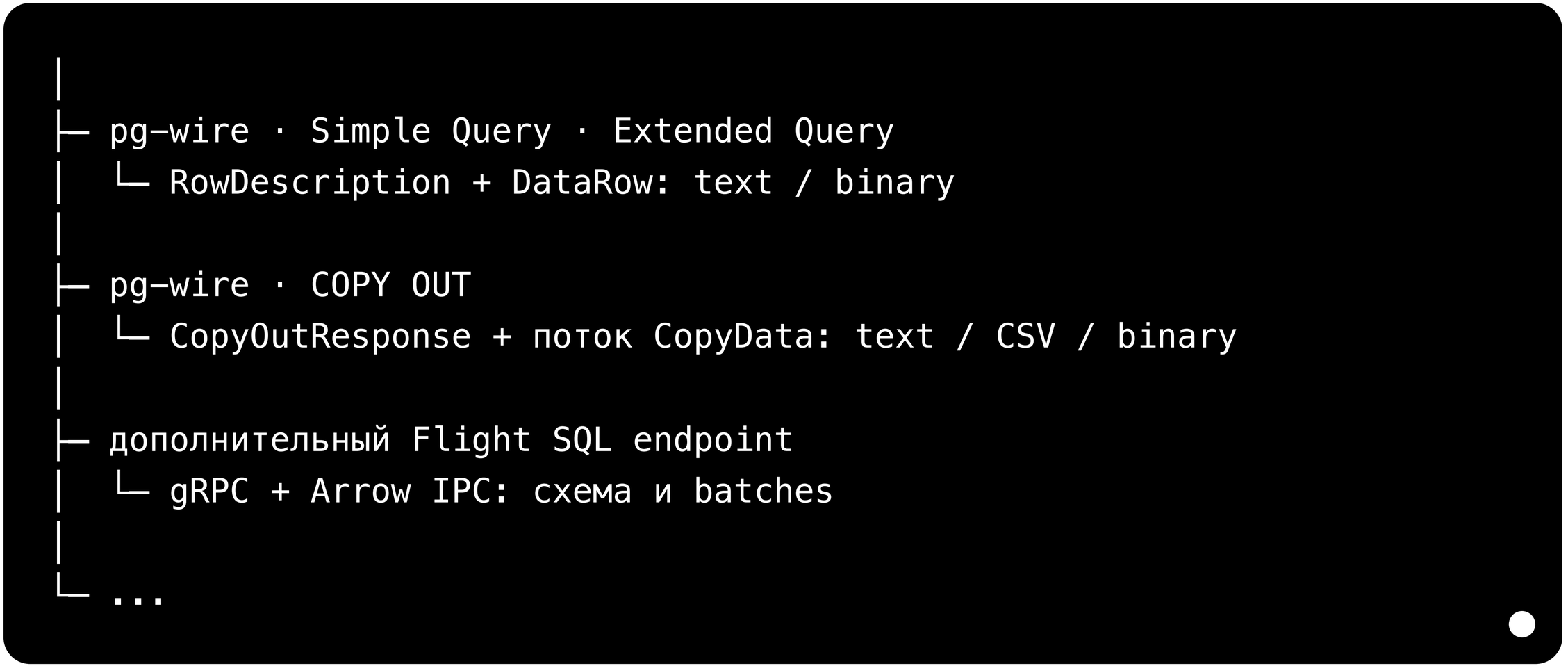
А по сети будут передаваться по-разному

SELECT conf, starts_at, speaker FROM smartdata ORDER BY starts_at.

Что летит по сети в PostgreSQL

Проверим Wireshark

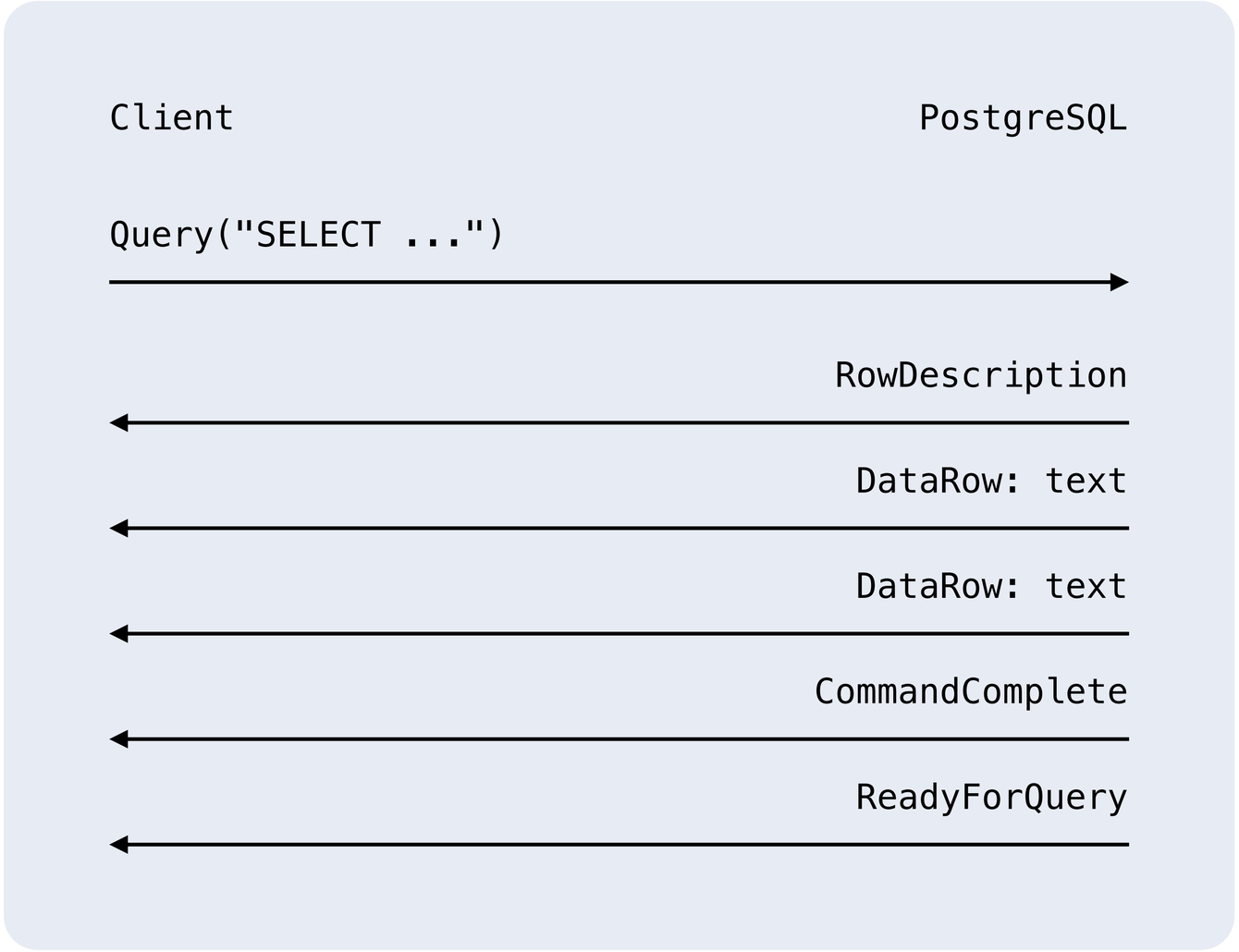
PostgreSQL → результат запроса



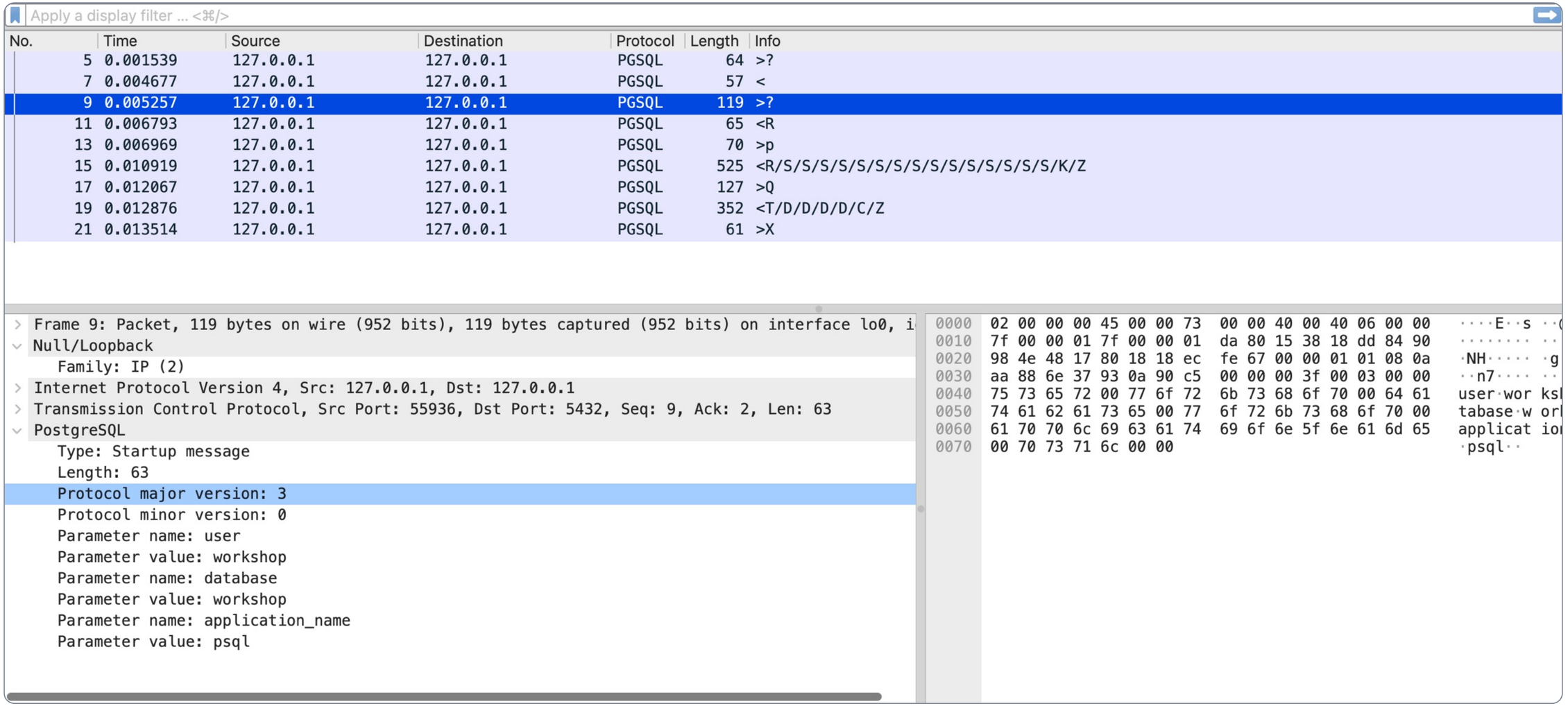
PostgreSQL → результат запроса

```
│
├─ pg-wire · Simple Query · Extended Query
│  └─ RowDescription + DataRow: text / binary
│
├─ pg-wire · COPY OUT
│  └─ CopyOutResponse + поток CopyData: text / CSV / binary
│
├─ дополнительный Flight SQL endpoint
│  └─ gRPC + Arrow IPC: схема и batches
│
└─ ...
```

PostgreSQL — simple query

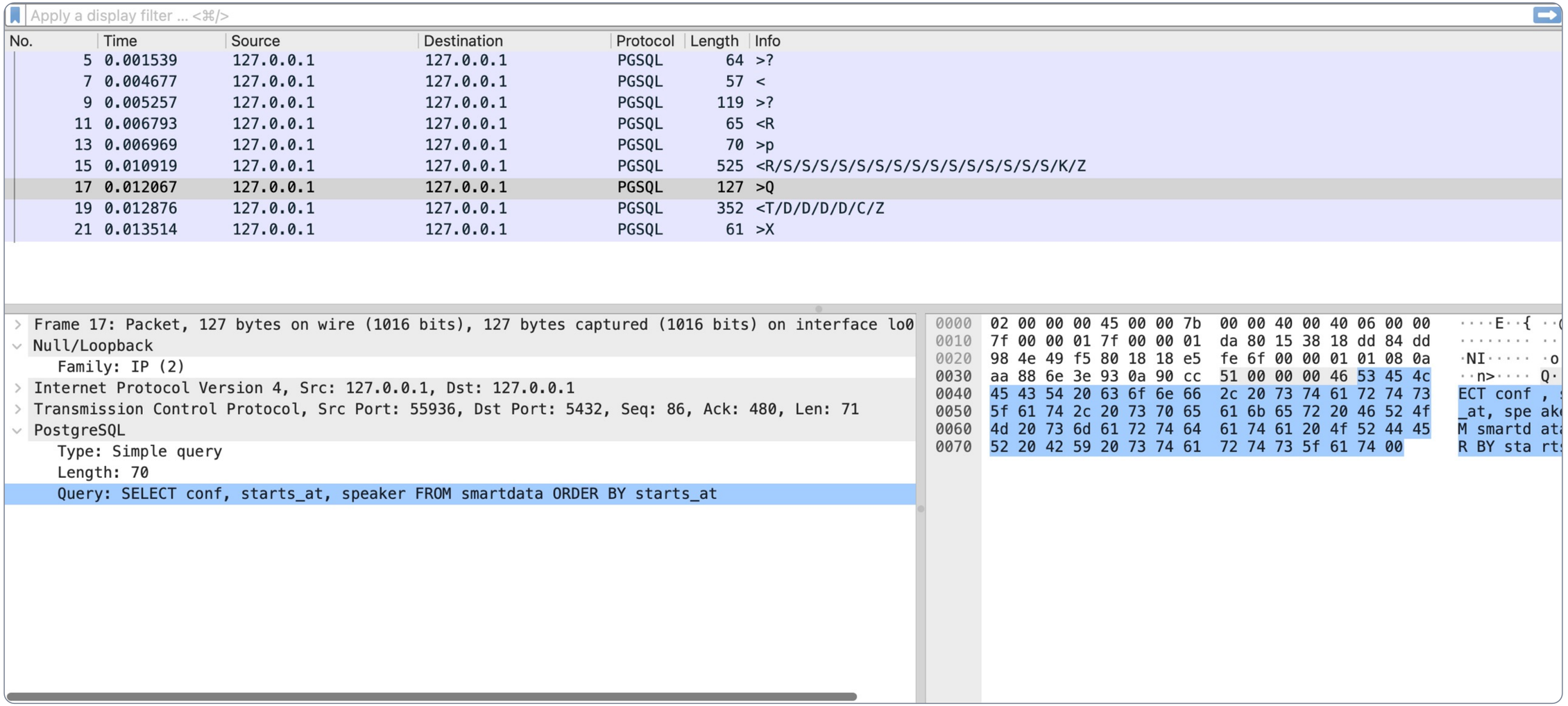


PostgreSQL — simple query



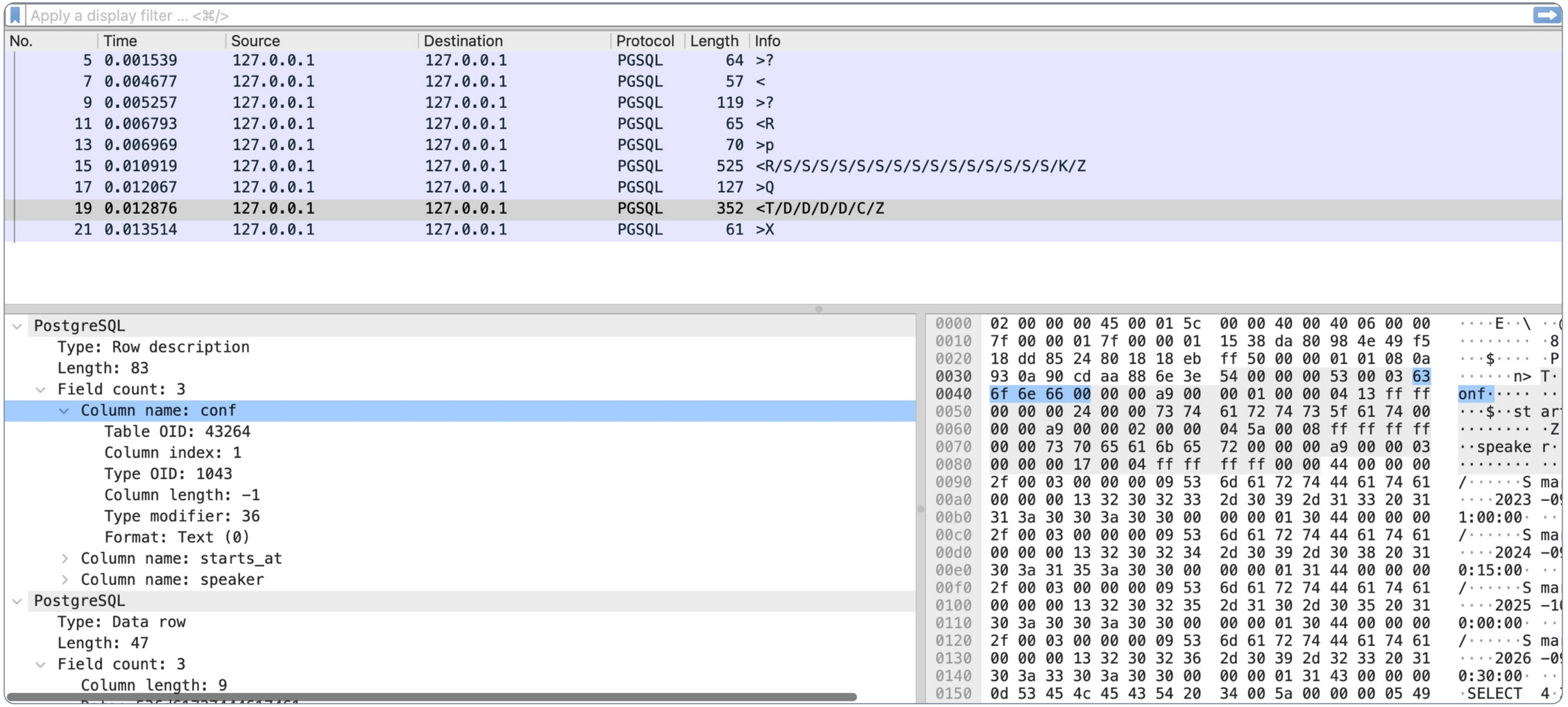
За сценарий: клиент получил 13 пак. (1515 Б), отправил 13 пак. (901 Б). Всего 2416 Б.
TCP payload: 936 Б (38,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — simple query



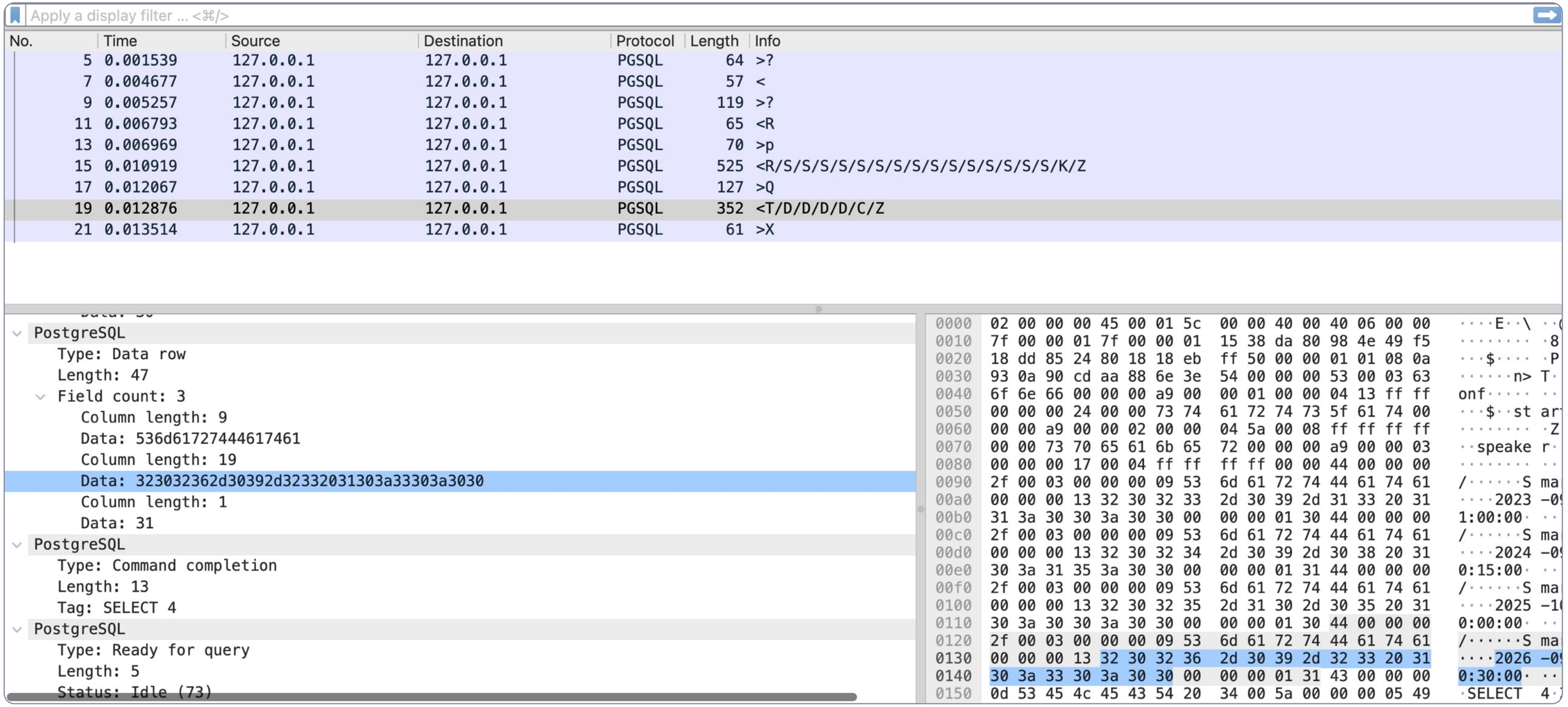
За сценарий: клиент получил 13 пак. (1515 Б), отправил 13 пак. (901 Б). Всего 2416 Б.
TCP payload: 936 Б (38,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — simple query



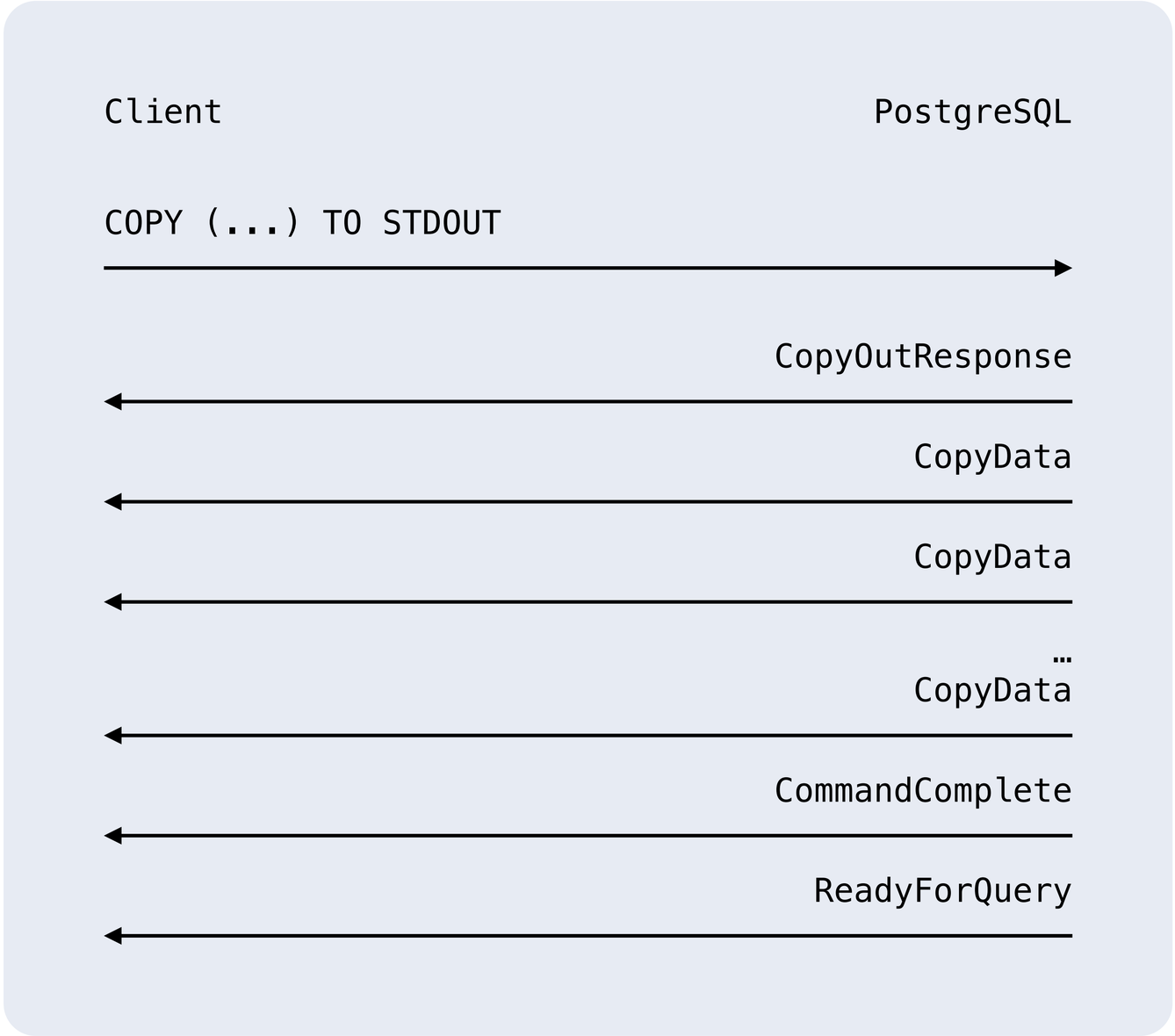
За сценарий: клиент получил 13 пак. (1515 Б), отправил 13 пак. (901 Б). Всего 2416 Б.
TCP payload: 936 Б (38,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — simple query

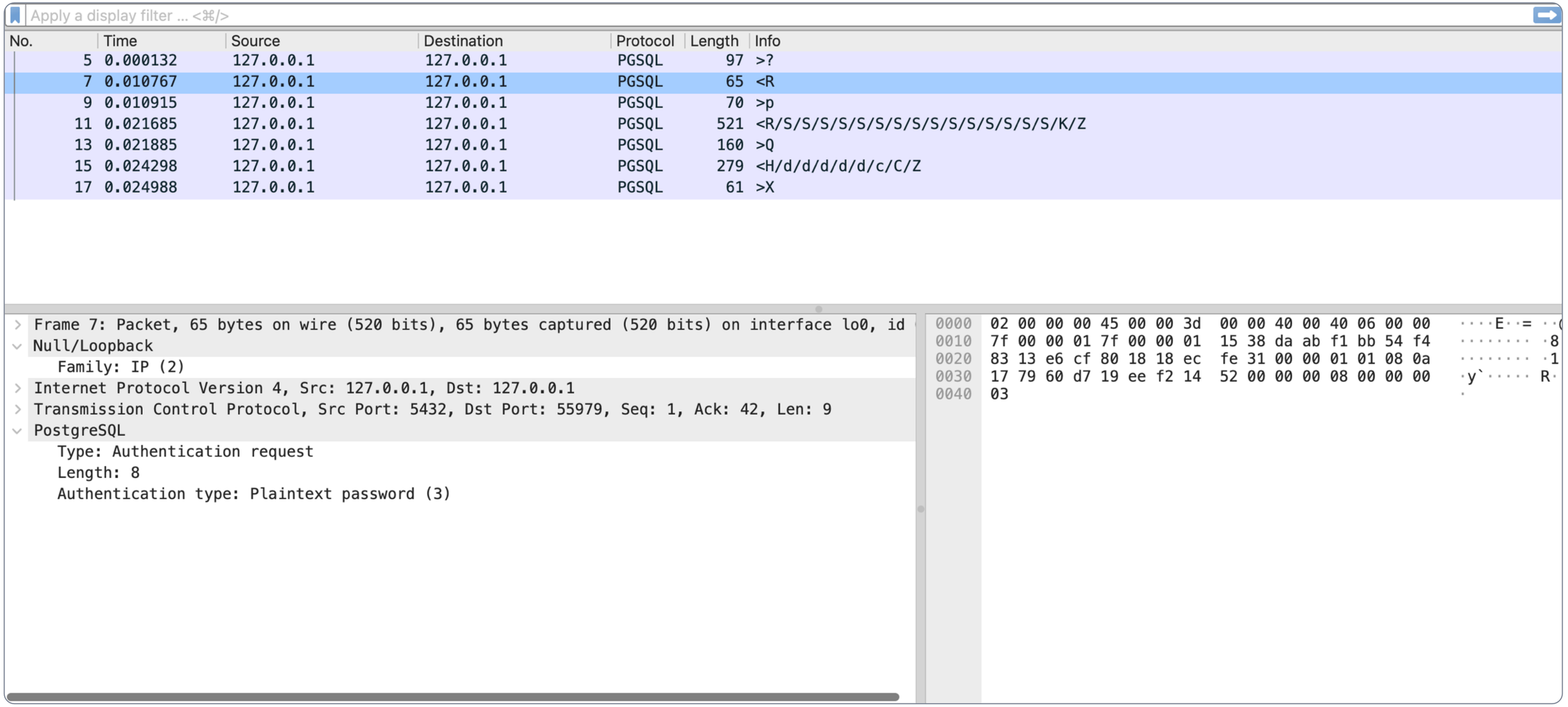


За сценарий: клиент получил 13 пак. (1515 Б), отправил 13 пак. (901 Б). Всего 2416 Б.
TCP payload: 936 Б (38,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — COPY

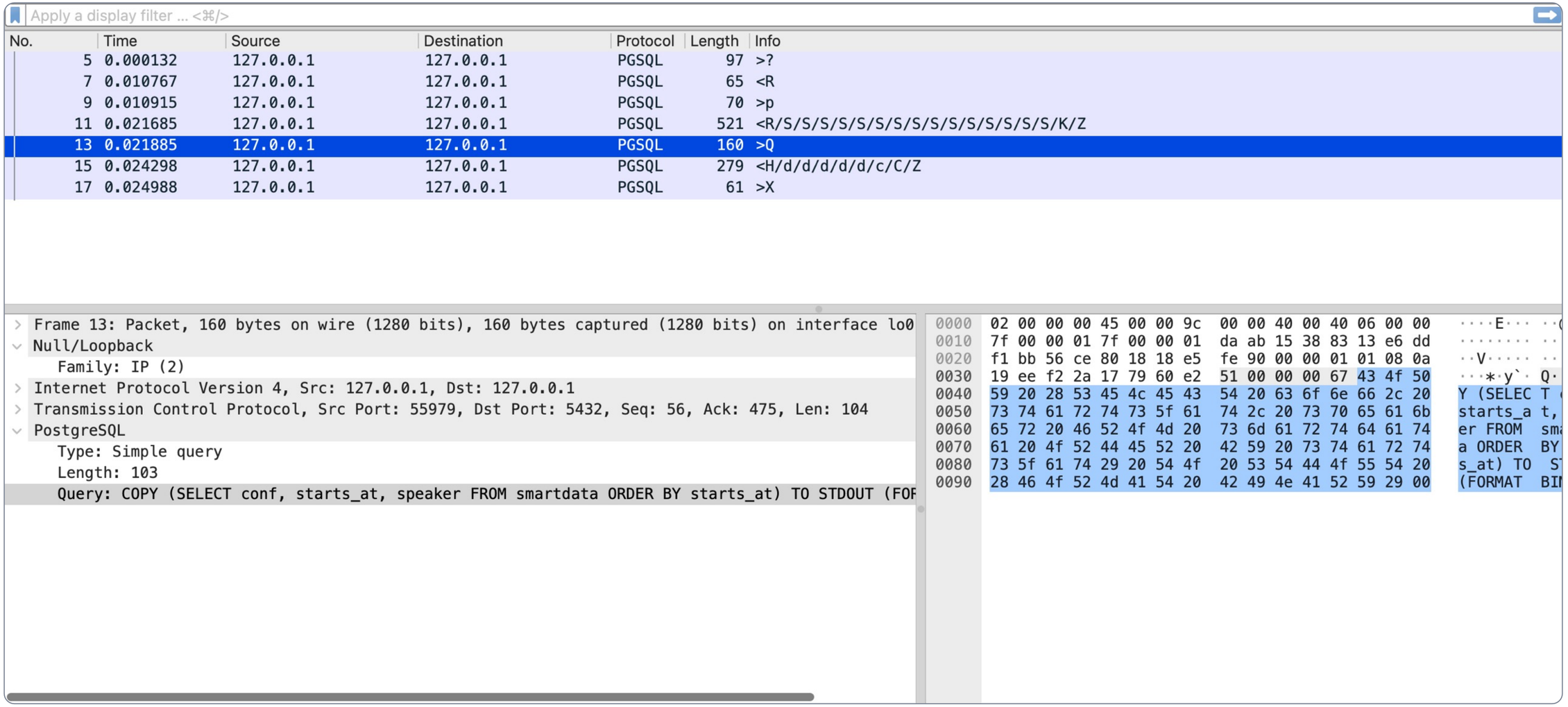


PostgreSQL — COPY



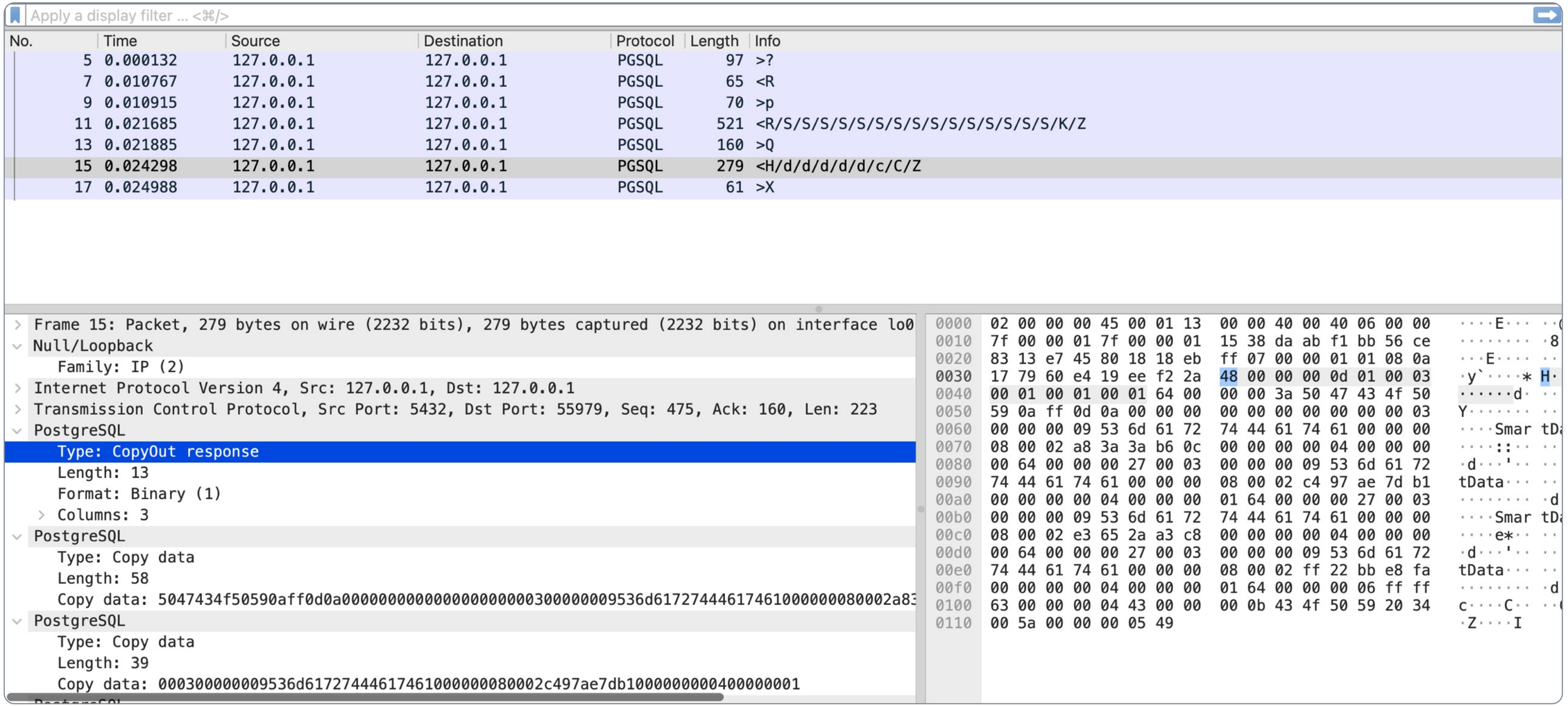
За сценарий: клиент получил 11 пак. (1325 Б), отправил 11 пак. (792 Б). Всего 2117 Б.
TCP payload: 861 Б (40,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — COPY



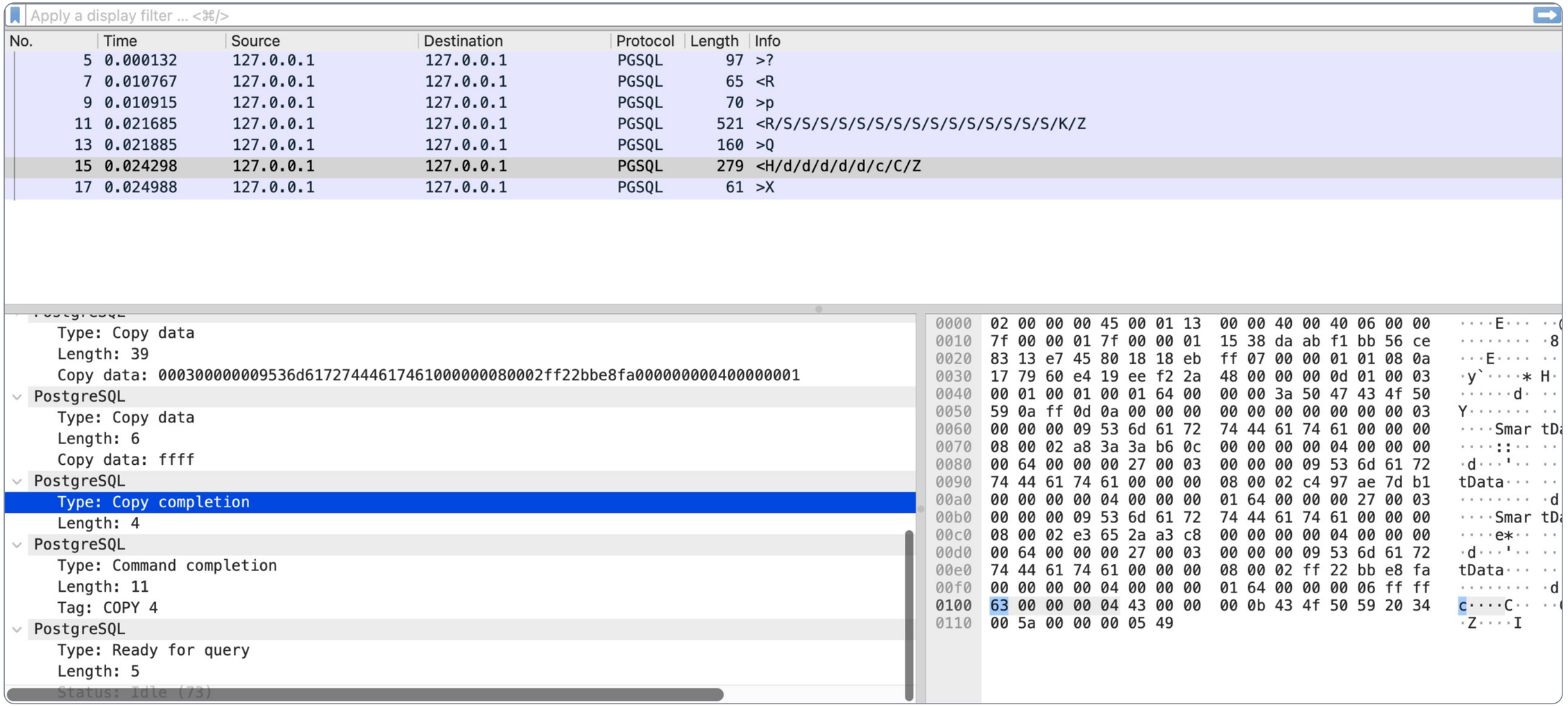
За сценарий: клиент получил 11 пак. (1325 Б), отправил 11 пак. (792 Б). Всего 2117 Б.
TCP payload: 861 Б (40,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — COPY



За сценарий: клиент получил 11 пак. (1325 Б), отправил 11 пак. (792 Б). Всего 2117 Б.
TCP payload: 861 Б (40,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — COPY



За сценарий: клиент получил 11 пак. (1325 Б), отправил 11 пак. (792 Б). Всего 2117 Б.
TCP payload: 861 Б (40,7% объёма кадров). Включает служебные сообщения.

PostgreSQL — ADBC

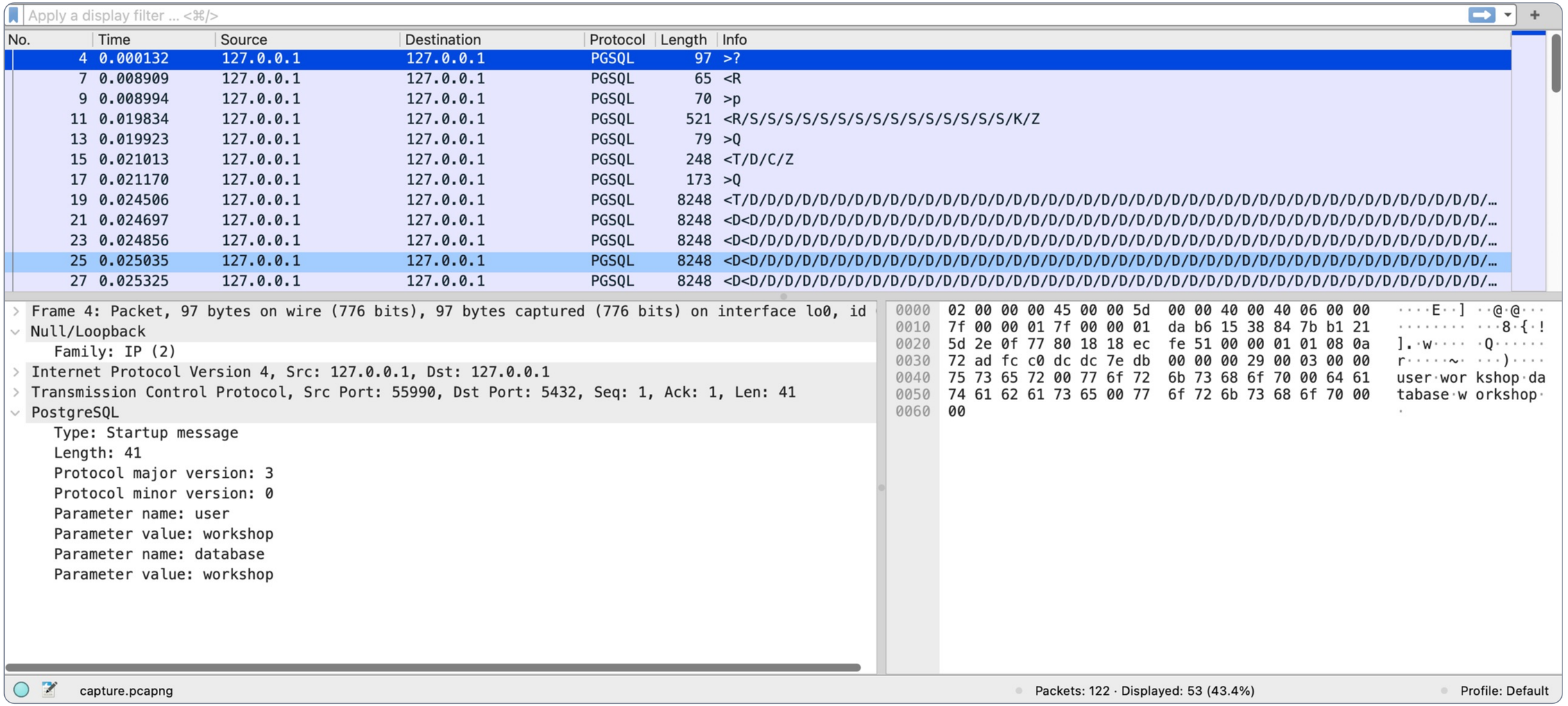
The PostgreSQL driver provides access to any database that supports
the PostgreSQL wire format. It wraps libpq, the client library for PostgreSQL.
The project owes credit to 0x0L's pgeon for the overall approach

```
import adbc_driver_postgresql.dbapi

uri = "postgresql://user:pass@localhost:5433/postgres"
with adbc_driver_postgresql.dbapi.connect(uri) as conn:
    pass
```

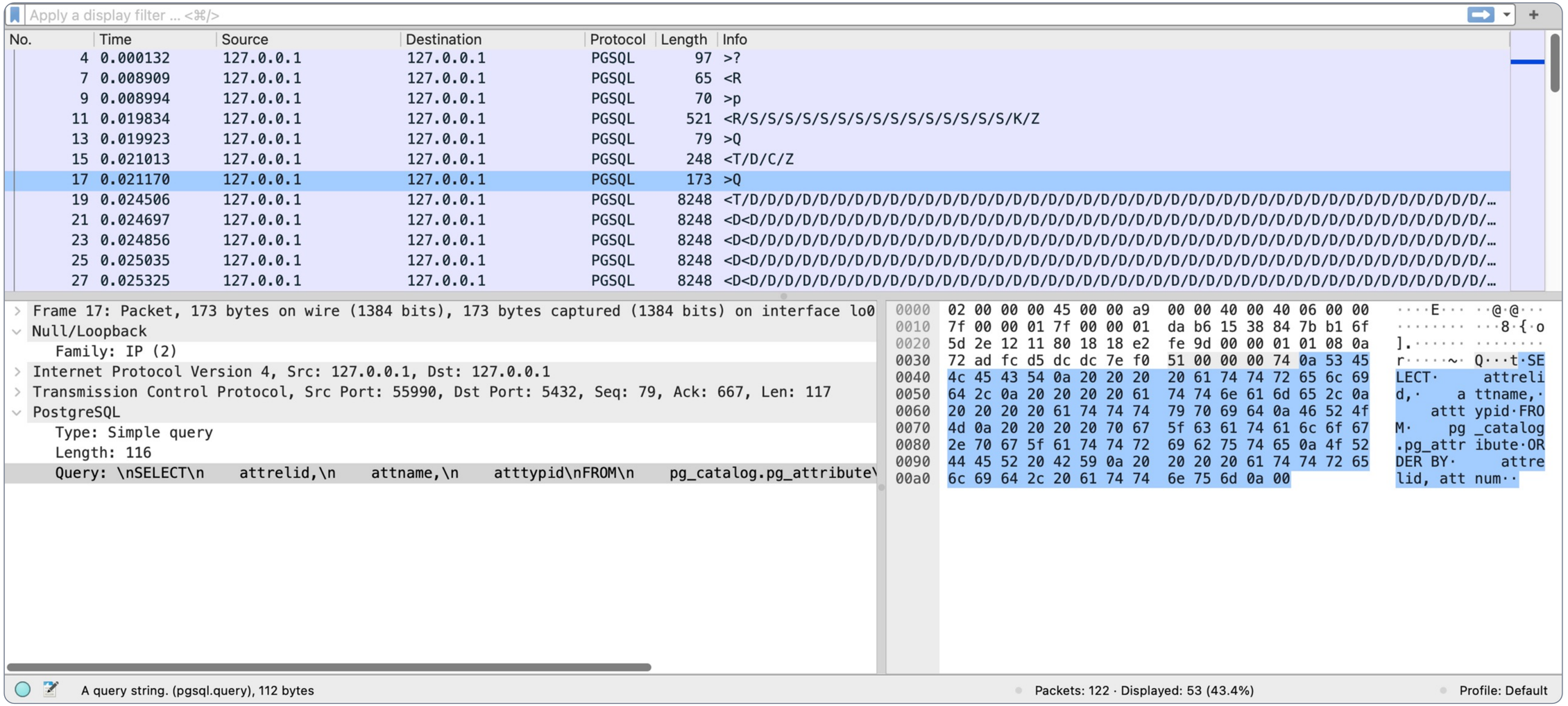
Источник

PostgreSQL — ADBC



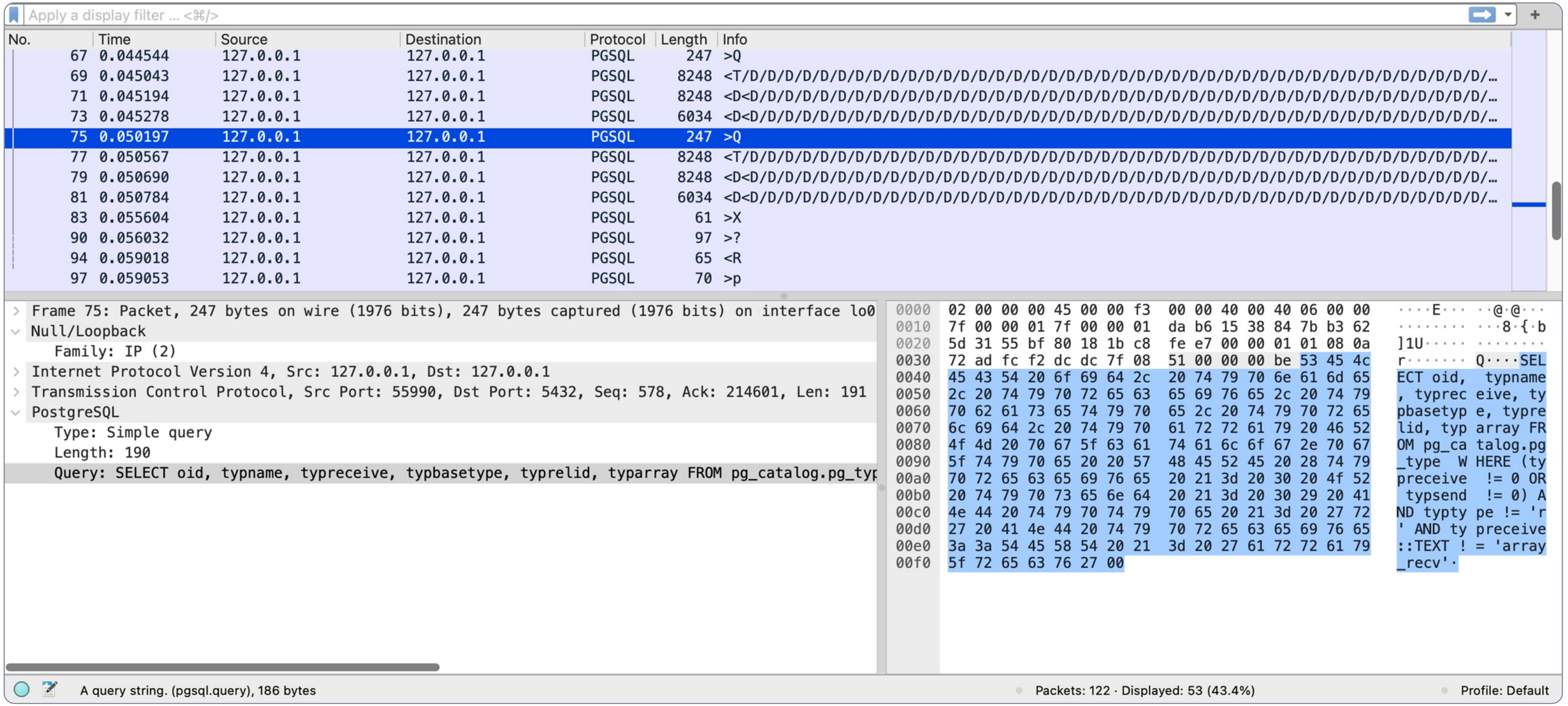
За сценарий: клиент получил 61 пак. (241 239 Б), отправил 61 пак. (4531 Б). Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL — ADBC



За сценарий: клиент получил 61 пак. (241 239 Б), отправил 61 пак. (4531 Б). Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

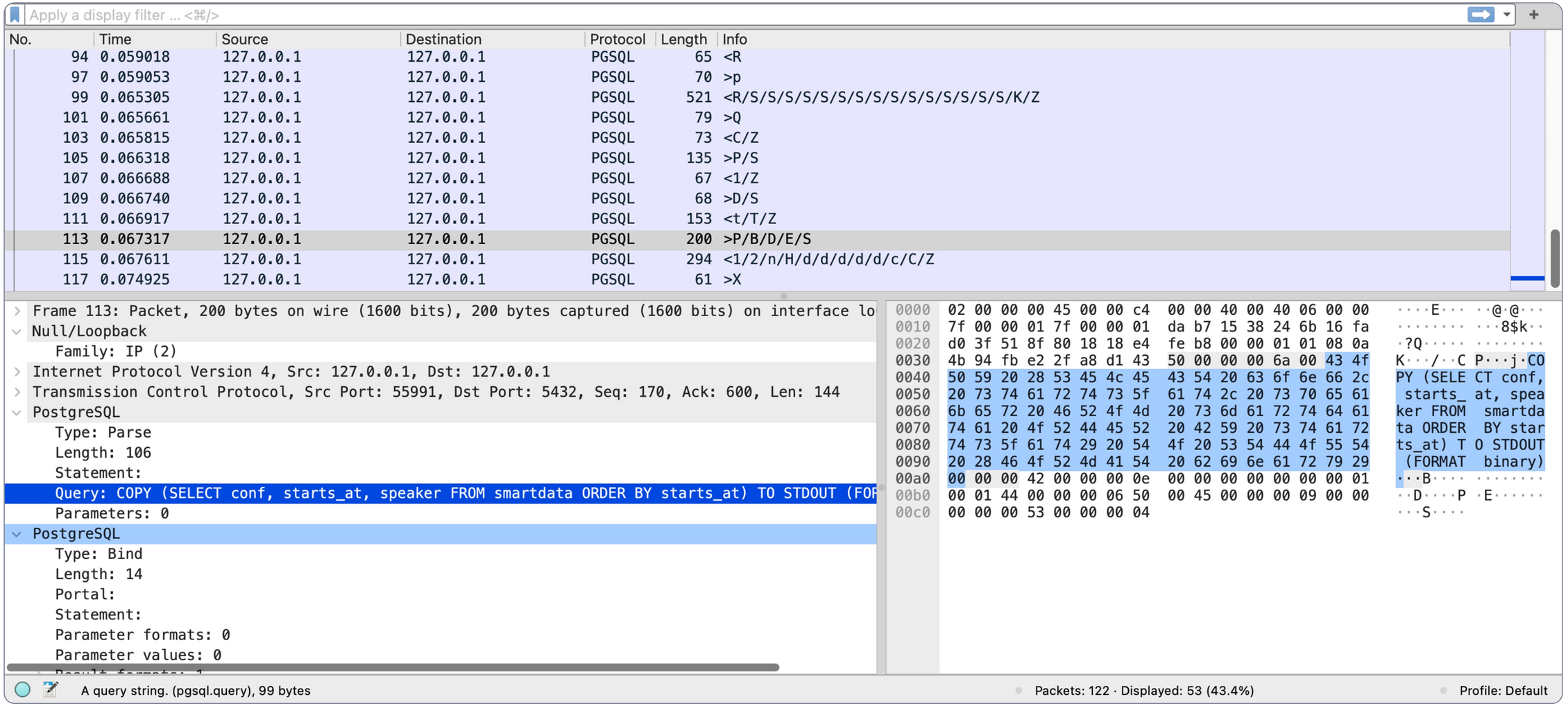
PostgreSQL — ADBC



За сценарий: клиент получил 61 пак. (241 239 Б), отправил 61 пак. (4531 Б). Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL — ADBC

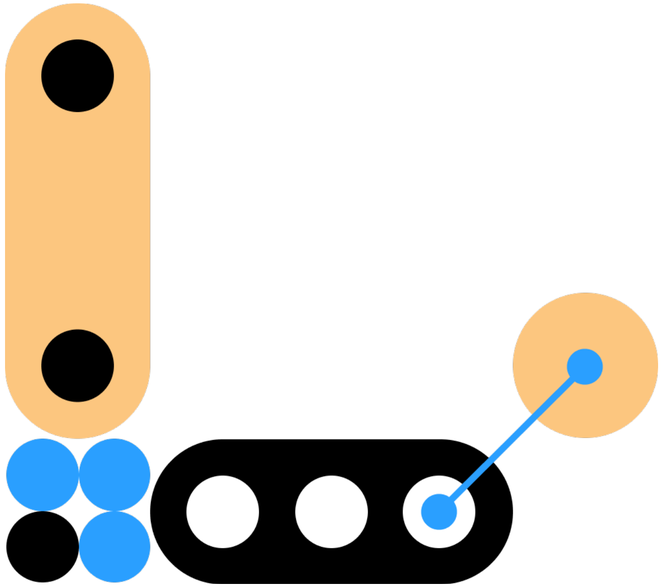
ADBC не заменил pg-wire,
драйвер вызвал COPY binary



За сценарий: клиент получил 61 пак. (241 239 Б), отправил 61 пак. (4531 Б). Всего 245 770 Б.
TCP payload: 238 890 Б (97,2% объёма кадров). Включает служебные сообщения.

PostgreSQL — Flight SQL

Переносит Arrow на отдельный серверный endpoint



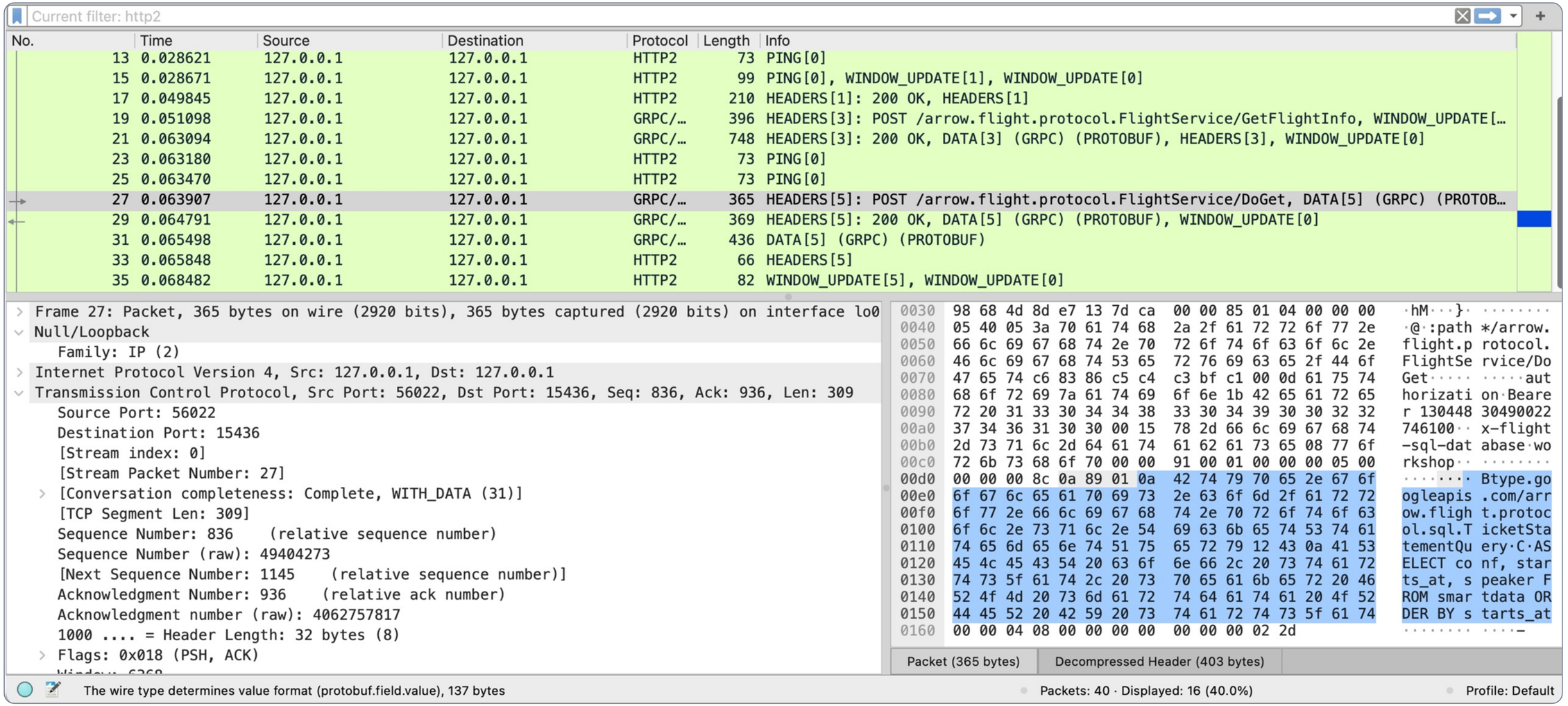
github.com/apache/arrow-flight-sql-postgresql

What is this?

- Apache Arrow Flight SQL adapter for PostgreSQL is a PostgreSQL extension that adds an Apache Arrow Flight SQL endpoint to PostgreSQL
- Apache Arrow Flight SQL is a protocol to use Apache Arrow format to interact with SQL databases. You can use Apache Arrow Flight SQL instead of the PostgreSQL wire protocol to interact with PostgreSQL by Apache Arrow Flight SQL adapter for PostgreSQL
- Apache Arrow format is designed for fast typed table data exchange. If you want to get large data by **SELECT or INSERT/UPDATE** large data, Apache Arrow Flight SQL will be faster than the PostgreSQL wire protocol

PostgreSQL — Flight SQL

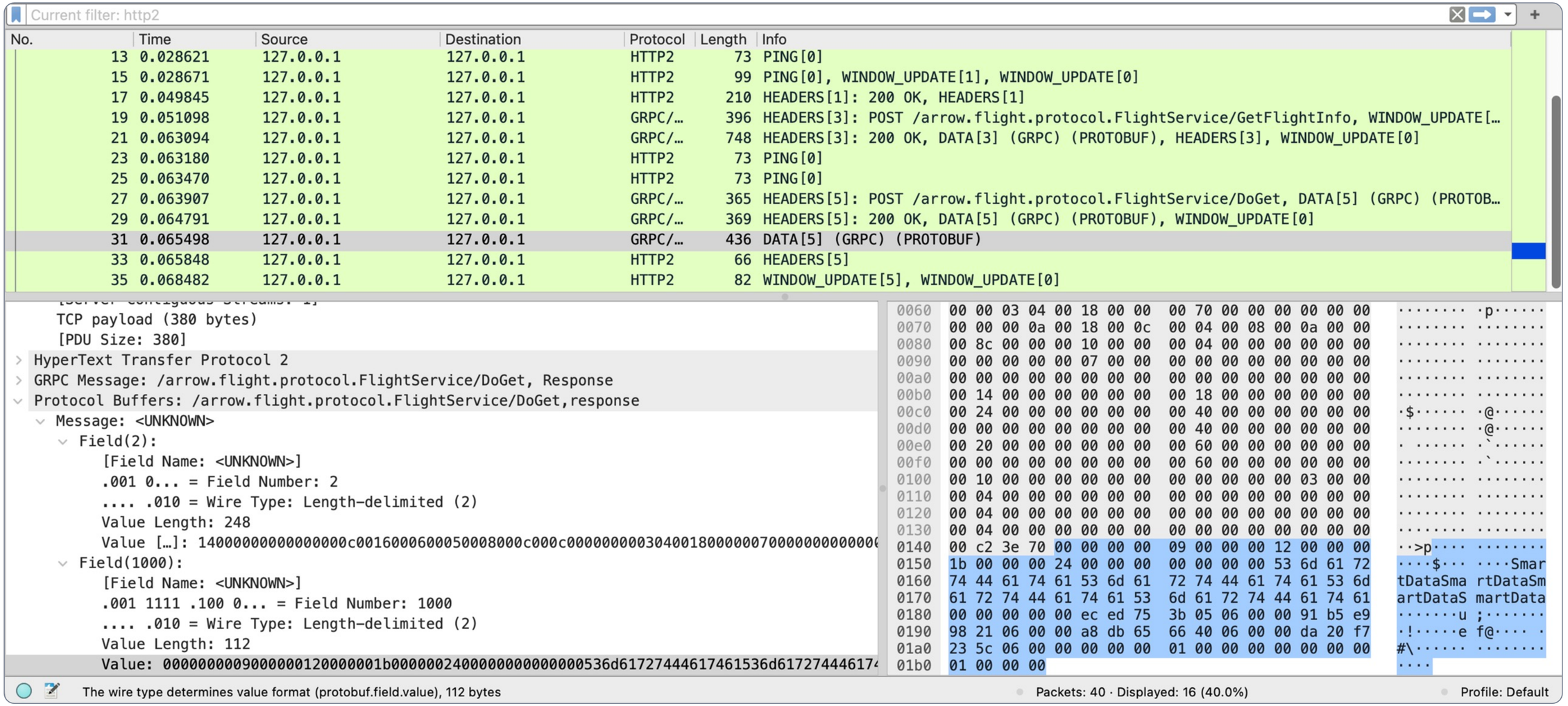
DoGet отвечает дважды: сначала схема Arrow, потом record batch



За сценарий: клиент получил 20 пак. (2770 Б), отправил 20 пак. (2302 Б). Всего 5072 Б.
TCP payload: 2808 Б (55,4% объёма кадров). Включает служебные сообщения.

PostgreSQL — Flight SQL

Буфером получаем данные



За сценарий: клиент получил 20 пак. (2770 Б), отправил 20 пак. (2302 Б). Всего 5072 Б.
TCP payload: 2808 Б (55,4% объёма кадров). Включает служебные сообщения.

Один SELECT в PostgreSQL

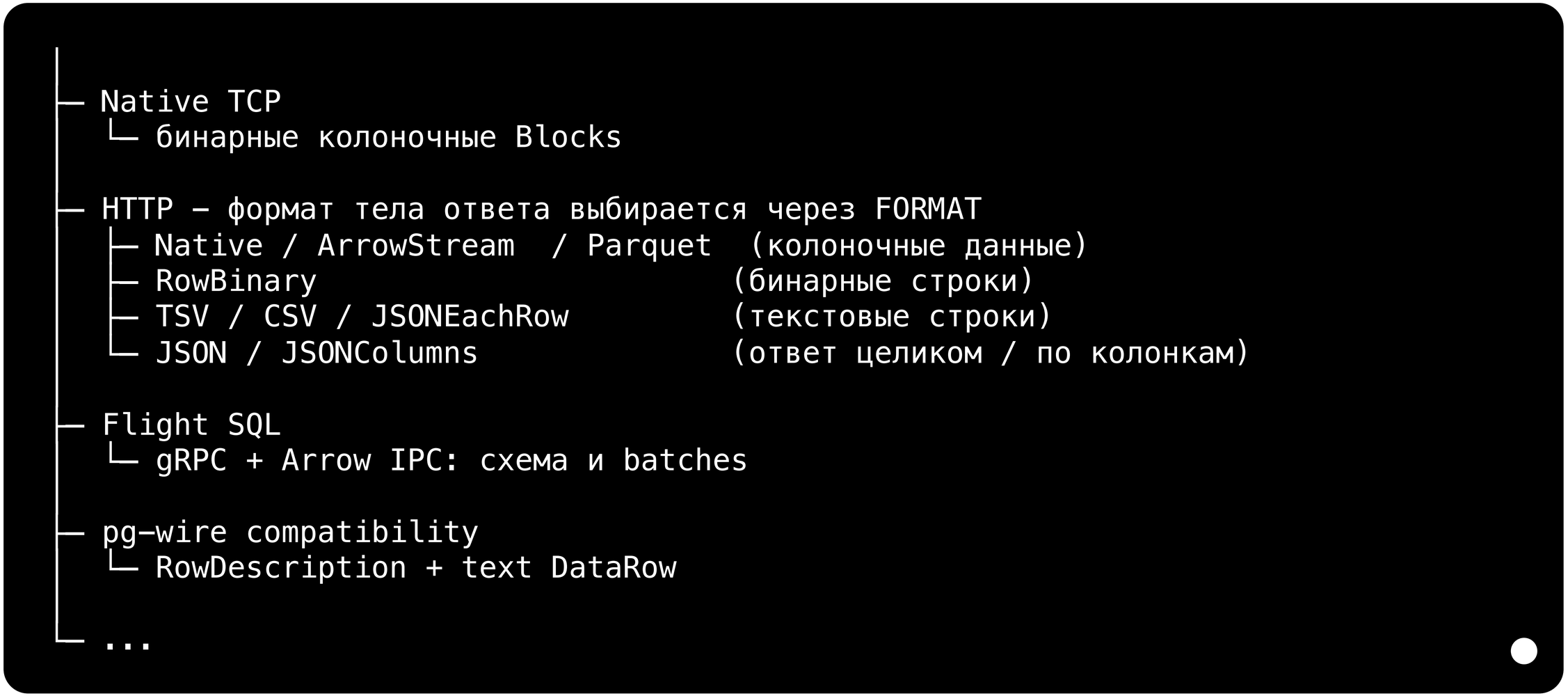
Способ	Что летит по сети	Где собирается конечный результат
Simple/extended	DataRow	Драйвер
COPY	COPY	Драйвер
ADBC PostgreSQL	COPY binary	Драйвер ADBC
Flight SQL	Arrow IPC	Сериализует результат

Протокол не всегда определяет, что именно будет передаваться по сети

Что летит по сети в ClickHouse®

Проверим Wireshark

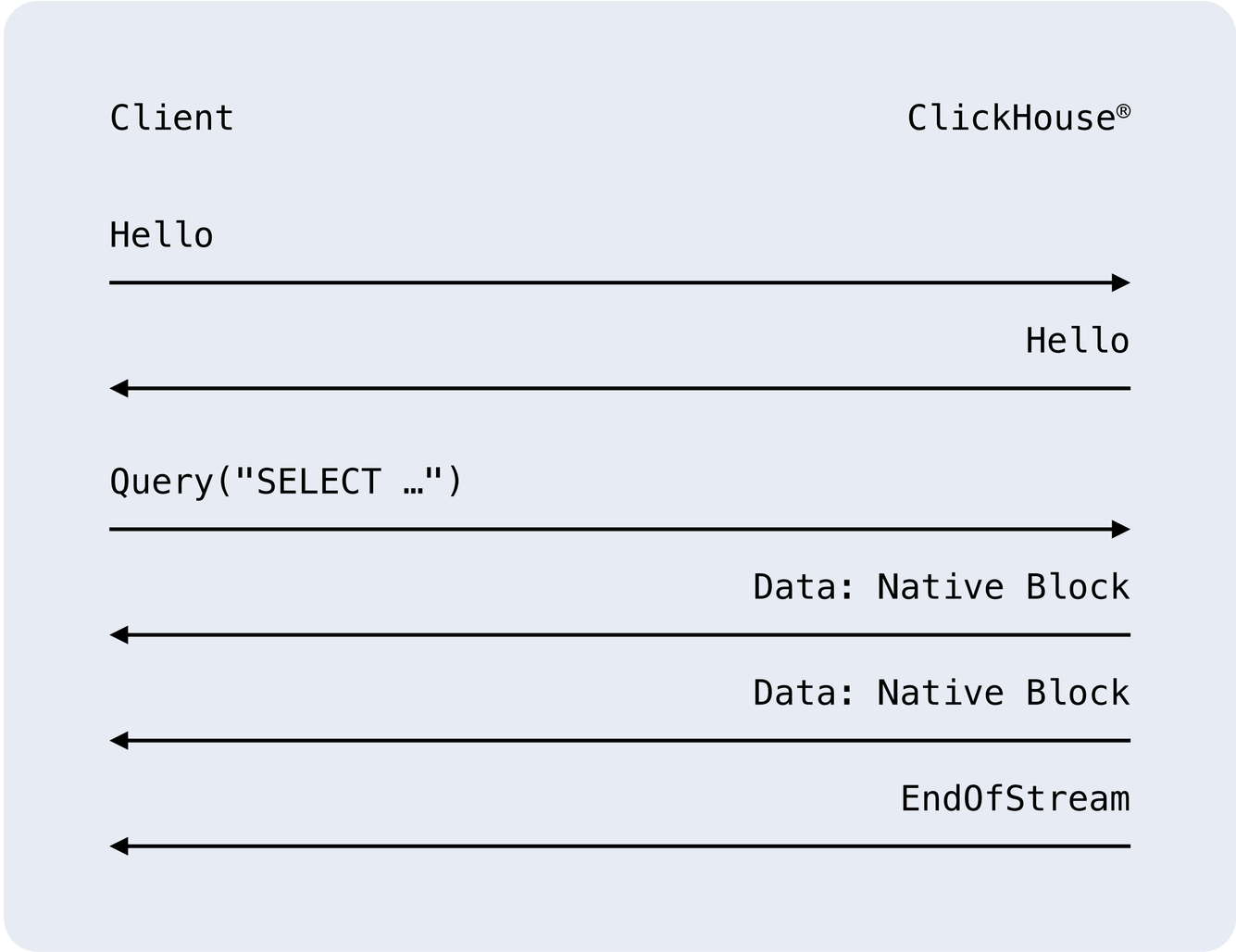
ClickHouse® → результат запроса



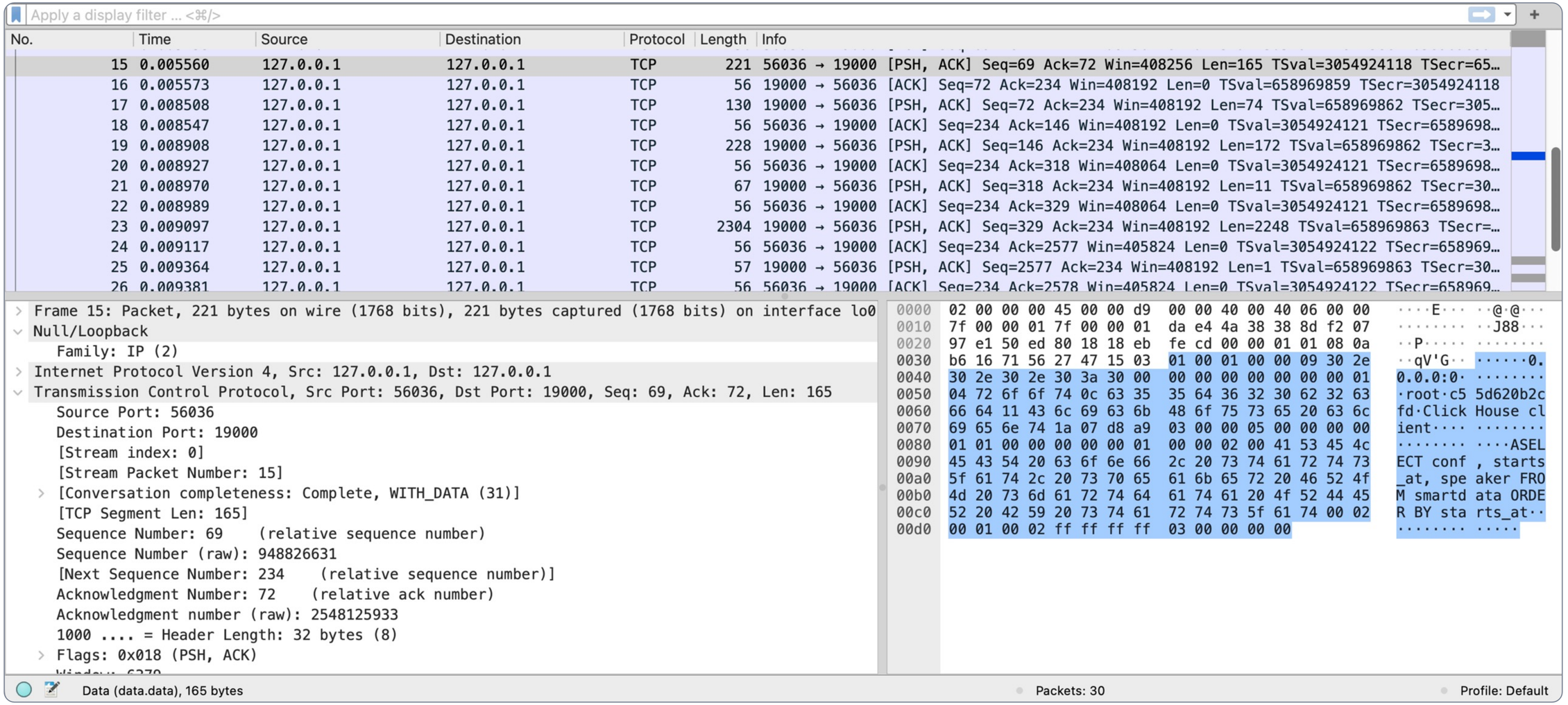
ClickHouse® → результат запроса

```
├─ Native TCP
│  └─ бинарные колоночные Blocks
│
├─ HTTP – формат тела ответа выбирается через FORMAT
│  ├─ Native / ArrowStream  / Parquet  (колоночные данные)
│  ├─ RowBinary                        (бинарные строки)
│  ├─ TSV / CSV / JSONEachRow          (текстовые строки)
│  └─ JSON / JSONColumns               (ответ целиком / по колонкам)
│
├─ Flight SQL
│  └─ gRPC + Arrow IPC: схема и batches
│
├─ pg-wire compatibility
│  └─ RowDescription + text DataRow
│
└─ ...
```

ClickHouse® — Native TCP



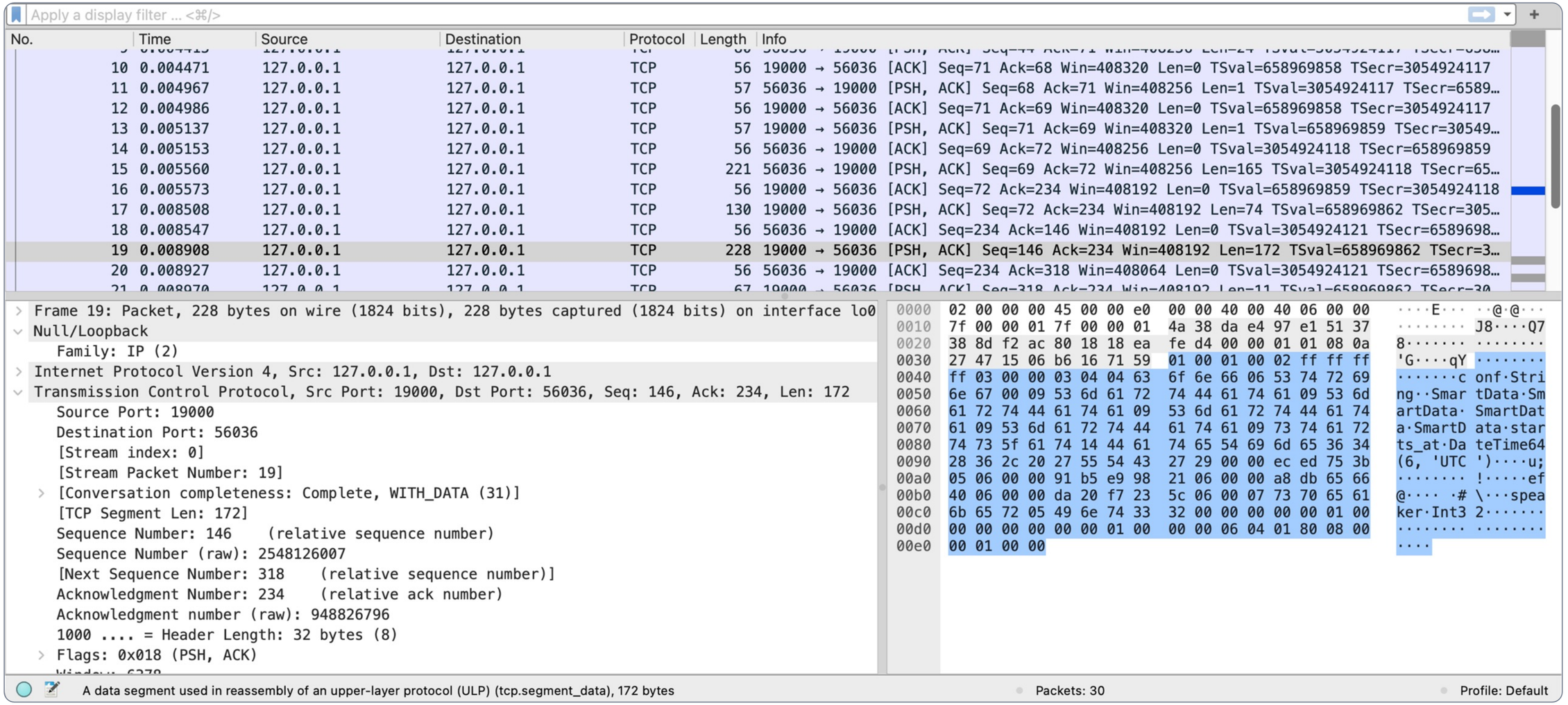
ClickHouse® — Native TCP



За сценарий: клиент получил 15 пак. (3429 Б), отправил 15 пак. (1085 Б). Всего 4514 Б.
TCP payload: 2810 Б (62,3% объёма кадров). Включает служебные сообщения.

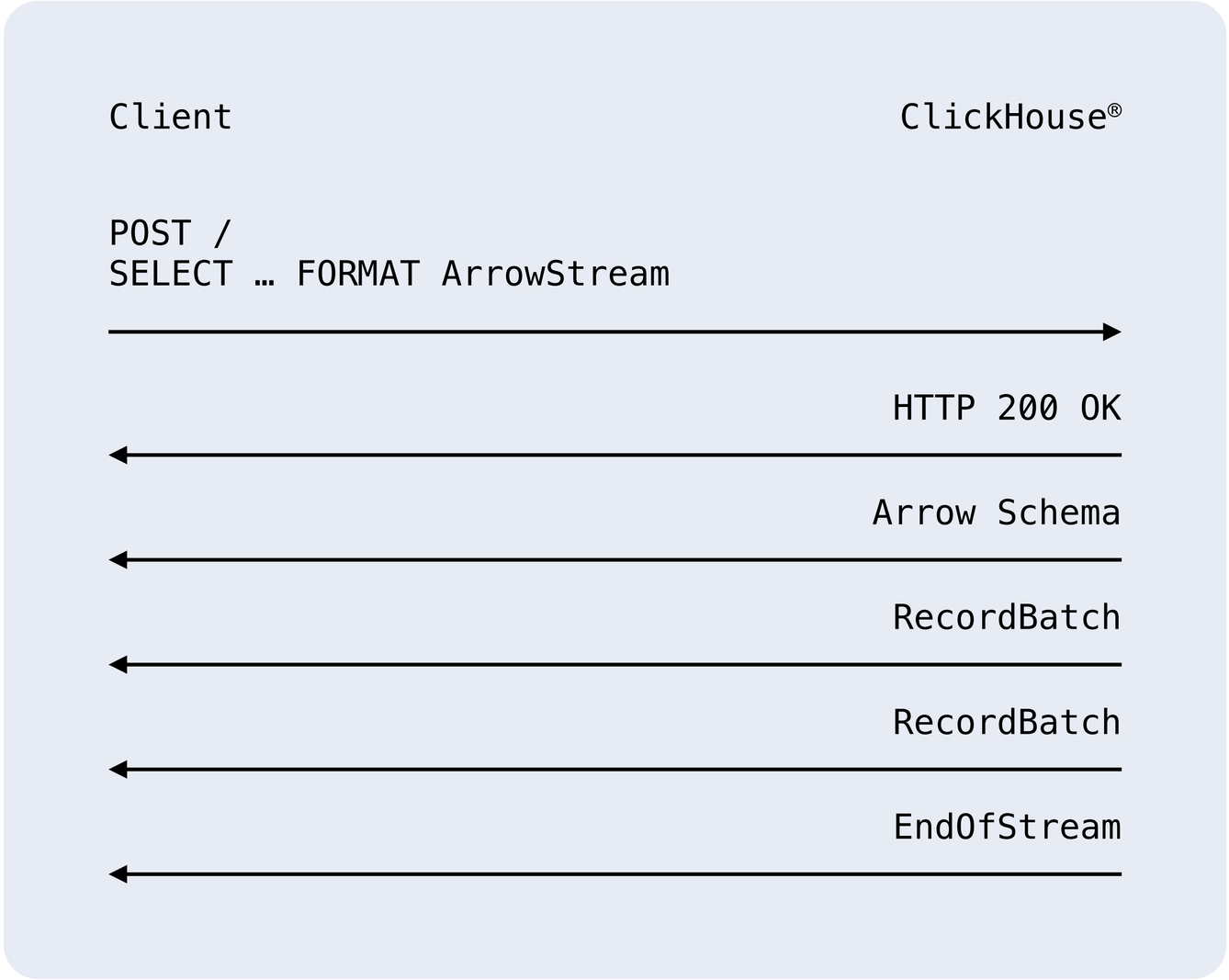
ClickHouse® — Native TCP

В payload подряд лежат четыре метки времени, следом — четыре Int32-колонки speaker

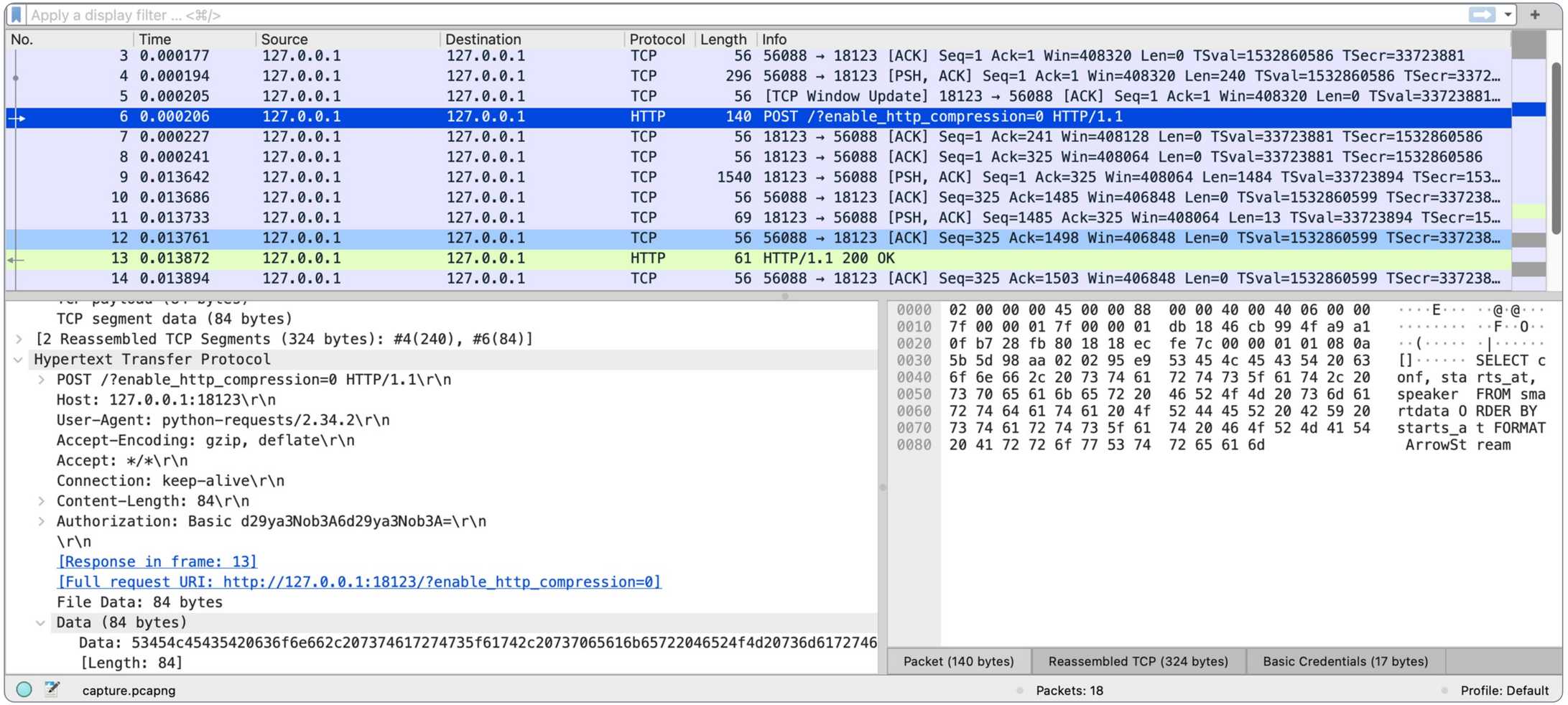


За сценарий: клиент получил 15 пак. (3429 Б), отправил 15 пак. (1085 Б). Всего 4514 Б.
TCP payload: 2810 Б (62,3% объёма кадров). Включает служебные сообщения.

ClickHouse® — HTTP — ADBC



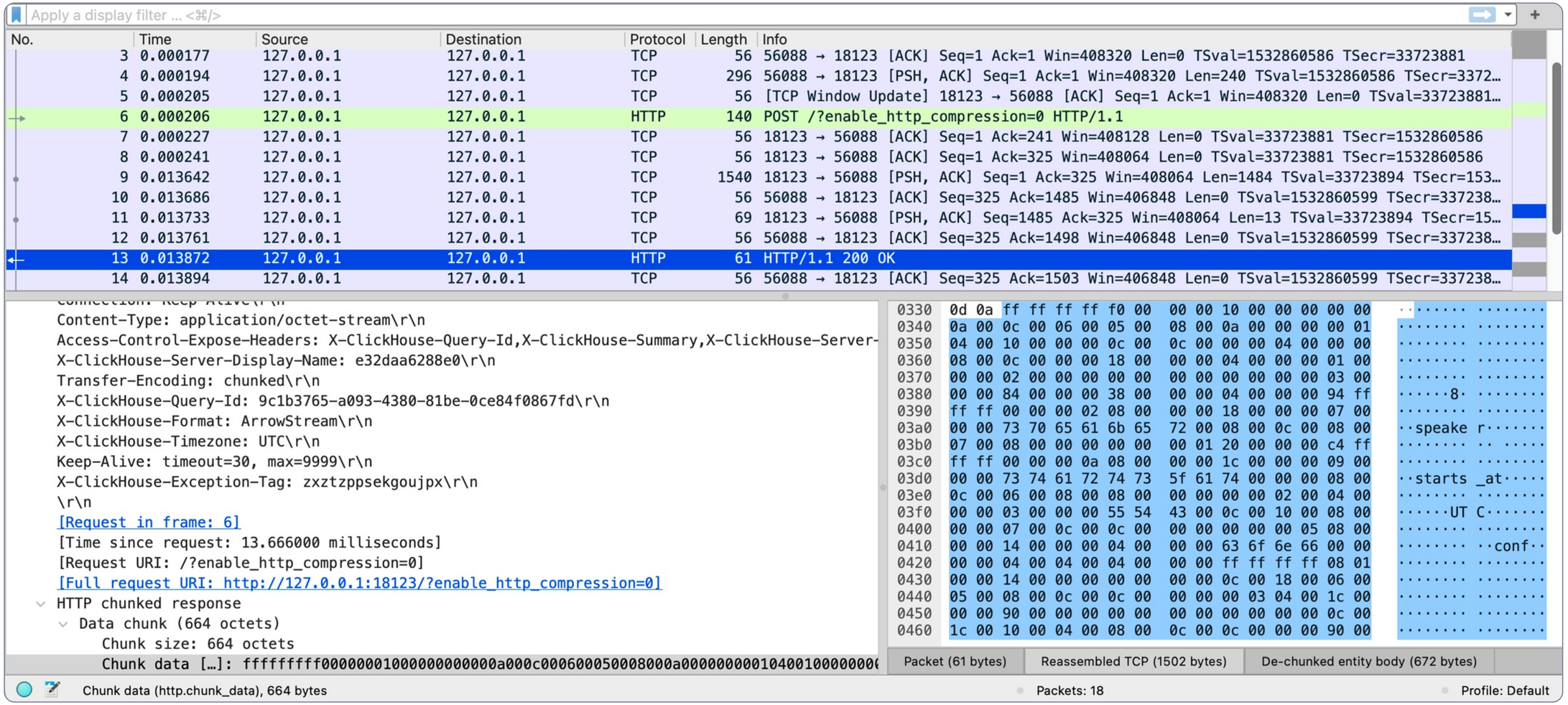
ClickHouse® HTTP — ADBC использует HTTP и запрашивает ArrowStream



За сценарий: клиент получил 9 пак. (2018 Б), отправил 9 пак. (840 Б). Всего 2858 Б.
TCP payload: 1826 Б (63,9% объёма кадров). Включает служебные сообщения.

ClickHouse® HTTP — ADBC использует HTTP и запрашивает ArrowStream

Ответ приходит как Arrow IPC



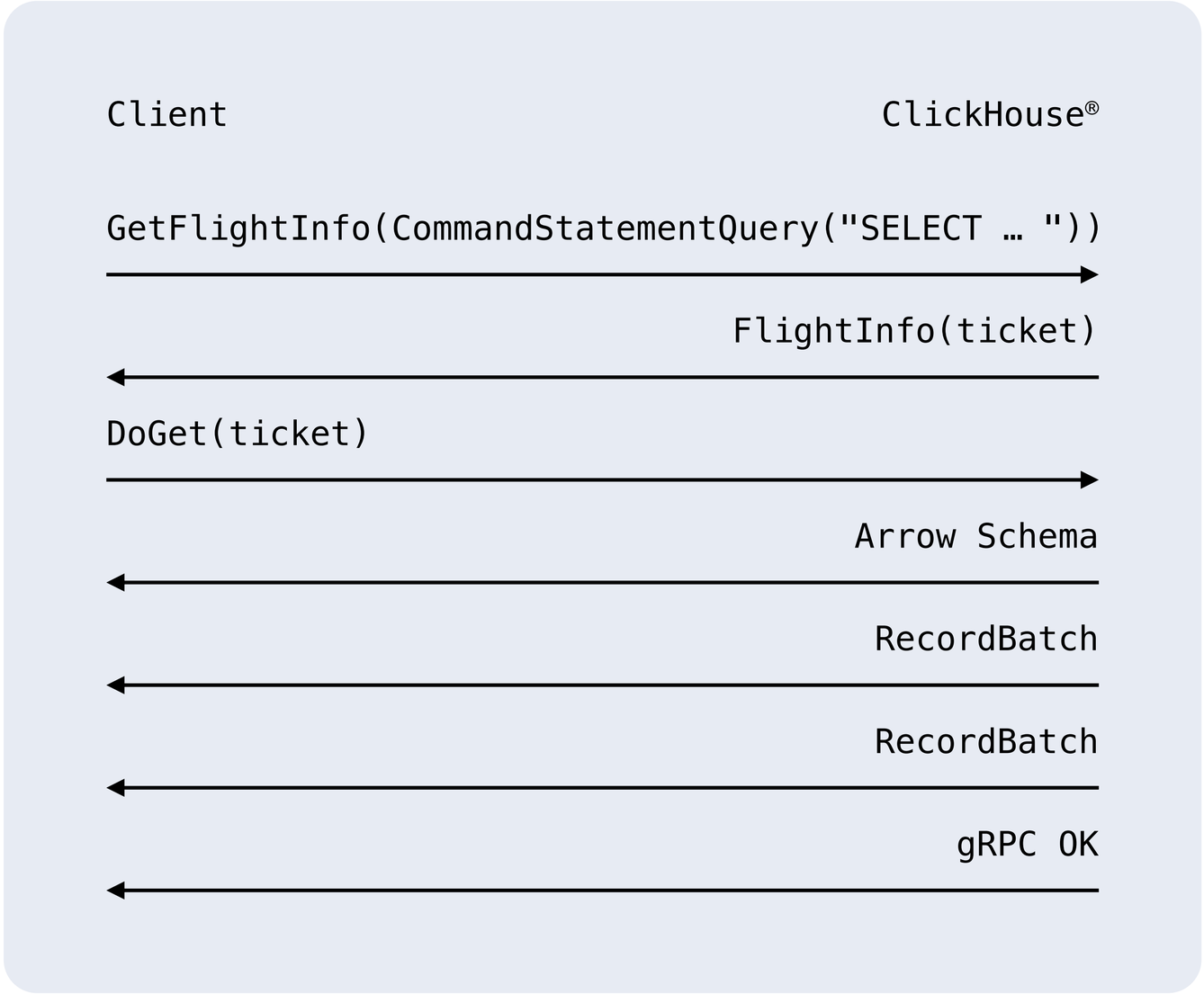
За сценарий: клиент получил 9 пак. (2018 Б), отправил 9 пак. (840 Б). Всего 2858 Б.
TCP payload: 1826 Б (63,9% объёма кадров). Включает служебные сообщения.

Драйверы ADBC. Разные пути до Arrow

Участок	ADBC PostgreSQL	ADBC ClickHouse®	ADBC Flight SQL
Клиентский API	ADBC		
Драйвер	PostgreSQL driver + libpq	ClickHouse® ADBC driver	Flight SQL driver
Серверная сторона	Обычный PostgreSQL	ClickHouse® HTTP endpoint	Flight SQL endpoint в БД или адаптере
Протокол/транспорт	Pg-wire/TCP	HTTP(S)	Flight SQL / gRPC
Формат на сети	COPY binary → CopyData	ArrowStream → Arrow IPC	Arrow IPC batches
Где появляется Arrow	Собирает клиентский драйвер	Сериализует ClickHouse®	Сериализует Flight SQL endpoint
Результат ADBC	Поток Arrow batches		

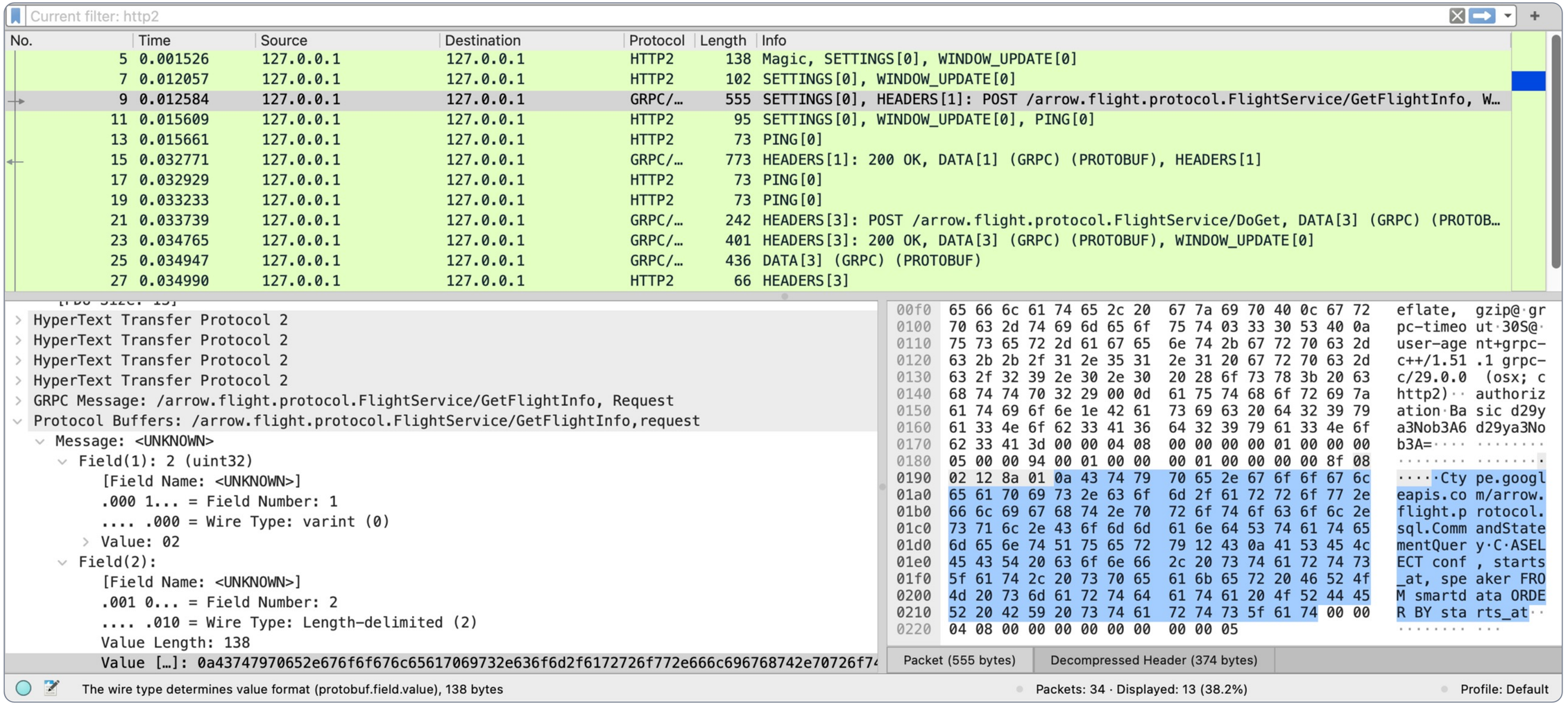
Протокол не всегда определяет, что именно будет передавать по сети

ClickHouse® — Flight SQL



ClickHouse® — Flight SQL

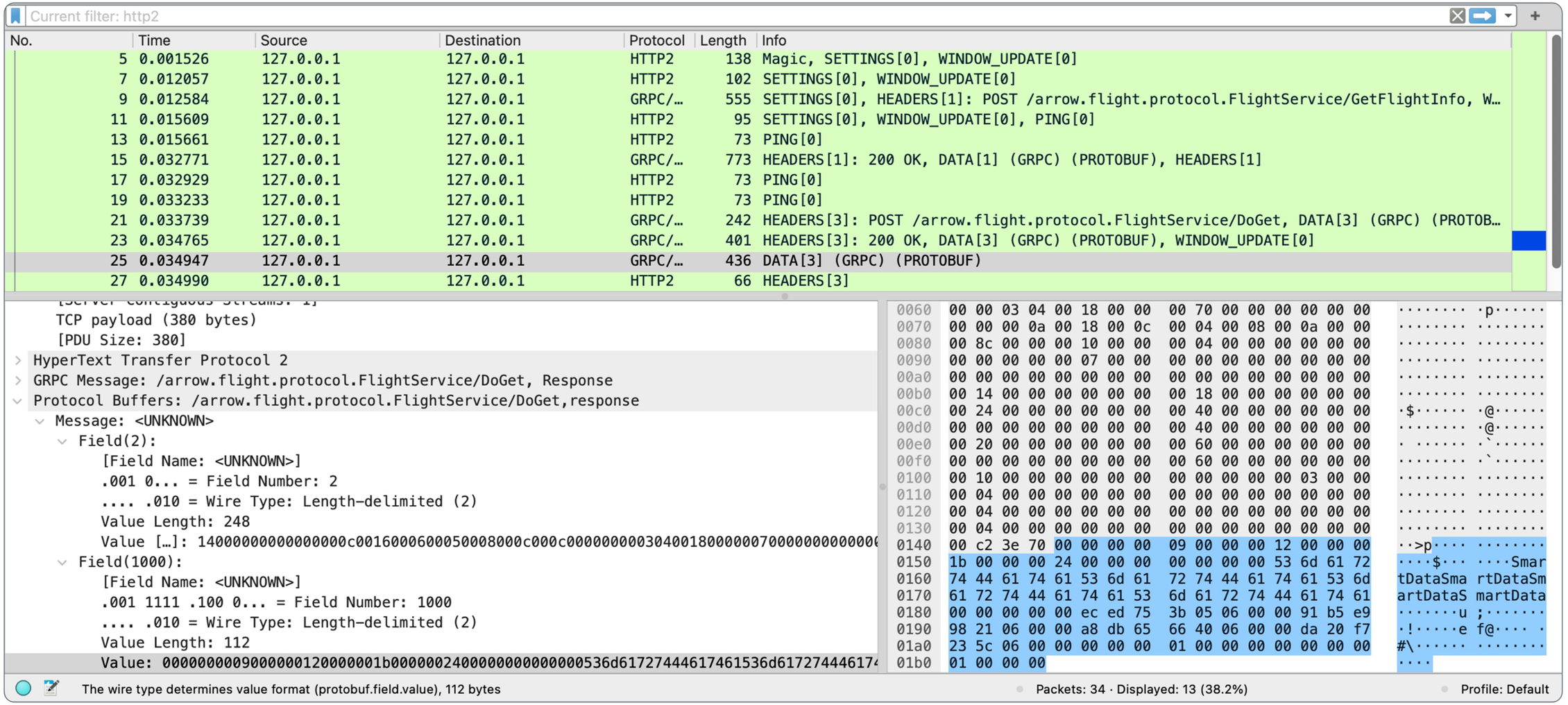
GetFlightInfo → ticket → DoGet
→ Arrow schema и batch



За сценарий: клиент получил 17 пак. (2518 Б), отправил 17 пак. (1791 Б). Всего 4309 Б.
TCP payload: 2381 Б (55,3% объёма кадров). Включает служебные сообщения.

ClickHouse® — Flight SQL

Выглядит одинаково поверх обеих баз



За сценарий: клиент получил 17 пак. (2518 Б), отправил 17 пак. (1791 Б). Всего 4309 Б.
TCP payload: 2381 Б (55,3% объёма кадров). Включает служебные сообщения.

Что намерили бенчмарком

БД и клиент работают на разных мощностях

Клиенты	Сеть	Серверы БД
32 vCPU 128 ГБ RAM Python®-драйверы	Находятся в одной VPC и одной зоне	16 vCPU 64 ГБ RAM PostgreSQL 18.6 / ClickHouse® 26.7.5.10

Датасет

Публичный ClickBench hits

Срез	Колонки	Какой паттерн эмулируем
w10	WatchID, JavaEnable, Title, EventTime, CounterID, ClientIP, RegionID, UserID, URL, Referer	Числа, строки, timestamp
w50	Первые 50 колонок DDL	Смешанные числа и строки
w105	Вся таблица	Широкая аналитическая выгрузка

12 подмножеств одного датасета

В Parquet (zstd) / CSV размер* в МБ

Строки/колонки	w10	w50	w105
10 тыс.	0,5 / 11	0,6 / 12	0,8 / 15
100 тыс.	5,2 / 104	5,9 / 121	7,7 / 147
1 млн	55 / 747	62 / 905	88 / 1178
10 млн	691 / 6609	799 / 8305	1073 / 10 993

* Размер с округлением.

PostgreSQL

Путь	Кортежи	Arrow	Pandas	Polars
psycopg simple text	Да	—	Да	Да
psycopg extended text	Да	—	Да	Да
psycopg extended binary	Да	—	Да	Да
COPY text	—	—	—	—
COPY binary	—	—	—	—
COPY csv	—	—	Да	Да
ADBC поверх COPY	—	Да	Да	Да
ADBC без COPY	—	Да	Да	Да
Connectorx binary	—	Да	Да	Да
Адаптер Flight SQL	—	Да (w10)	Да (w10)	Да (w10)

PostgreSQL

Строки/колонки	w10	w50	w105
10 тыс.	text 0,061 с / 5 МБ COPY CSV 0,097 с / 4 МБ ADBC 0,035 с / 4 МБ	text 0,178 с / 7 МБ COPY CSV 0,168 с / 6 МБ ADBC 0,077 с / 7 МБ	text 0,331 с / 11 МБ COPY CSV 0,271 с / 8 МБ ADBC 0,137 с / 11 МБ
100 тыс.	text 0,694 с / 48 МБ COPY CSV 0,812 с / 45 МБ ADBC 0,288 с / 45 МБ Flight SQL* 0,463 с / 41 МБ	text 1,84 с / 74 МБ COPY CSV 1,48 с / 60 МБ ADBC 0,600 с / 70 МБ	text 3,95 с / 111 МБ COPY CSV 2,37 с / 84 МБ ADBC 1,10 с / 106 МБ
1 млн	text 6,05 с / 406 МБ COPY CSV 6,86 с / 377 МБ ADBC 2,42 с / 373 МБ Flight SQL 4,79 с / 340 МБ	text 19,9 с / 662 МБ COPY CSV 13,1 с / 524 МБ ADBC 5,10 с / 618 МБ	text 43,3 с / 1049 МБ COPY CSV 22,2 с / 782 МБ ADBC 9,34 с / 995 МБ
10 млн	text 59,4 с / 3823 МБ COPY CSV 66,7 с / 3537 МБ ADBC 25,3 с / 3479 МБ	text 197,1 с / 6381 МБ COPY CSV 124,5 с / 5007 МБ ADBC 50,3 с / 5970 МБ	text 401,3 с / 10210 МБ COPY CSV 218,7 с / 7546 МБ ADBC 89,9 с / 9751 МБ

PostgreSQL

Строки/колонки	w10	w50	w105
10 тыс.	text 0,061 с / 5 МБ COPY CSV 0,097 с / 4 МБ ADBC 0,035 с / 4 МБ	text 0,178 с / 7 МБ COPY CSV 0,168 с / 6 МБ ADBC 0,077 с / 7 МБ	text 0,331 с / 11 МБ COPY CSV 0,271 с / 8 МБ ADBC 0,137 с / 11 МБ
100 тыс.	**text 0,694 с / 48 МБ** **COPY CSV 0,812 с / 45 МБ** **ADBC 0,288 с / 45 МБ** **Flight SQL* 0,463 с / 41 МБ**	text 1,84 с / 74 МБ COPY CSV 1,48 с / 60 МБ ADBC 0,600 с / 70 МБ	text 3,95 с / 111 МБ COPY CSV 2,37 с / 84 МБ ADBC 1,10 с / 106 МБ
1 млн	text 6,05 с / 406 МБ COPY CSV 6,86 с / 377 МБ ADBC 2,42 с / 373 МБ Flight SQL 4,79 с / 340 МБ	**text 19,9 с / 662 МБ** **COPY CSV 13,1 с / 524 МБ** **ADBC 5,10 с / 618 МБ**	text 43,3 с / 1049 МБ COPY CSV 22,2 с / 782 МБ ADBC 9,34 с / 995 МБ
10 млн	text 59,4 с / 3823 МБ COPY CSV 66,7 с / 3537 МБ ADBC 25,3 с / 3479 МБ	text 197,1 с / 6381 МБ COPY CSV 124,5 с / 5007 МБ ADBC 50,3 с / 5970 МБ	**text 401,3 с / 10210 МБ** **COPY CSV 218,7 с / 7546 МБ** **ADBC 89,9 с / 9751 МБ**

PostgreSQL

Строки/колонки	w10	w50	w105
10 тыс.	text 100% COPY CSV 158% ADBC 58%	text 100% COPY CSV 95% ADBC 44%	text 100% COPY CSV 82% ADBC 41%
100 тыс.	text 100% COPY CSV 117% ADBC 42% Flight SQL* 67%	text 100% COPY CSV 80% ADBC 33%	text 100% COPY CSV 60% ADBC 28%
1 млн	text 100% COPY CSV 113% ADBC 40% Flight SQL 79%	text 100% COPY CSV 66% ADBC 26%	text 100% COPY CSV 51% ADBC 22%
10 млн	text 100% COPY CSV 112% ADBC 43%	text 100% COPY CSV 63% ADBC 26%	text 100% COPY CSV 55% ADBC 22%

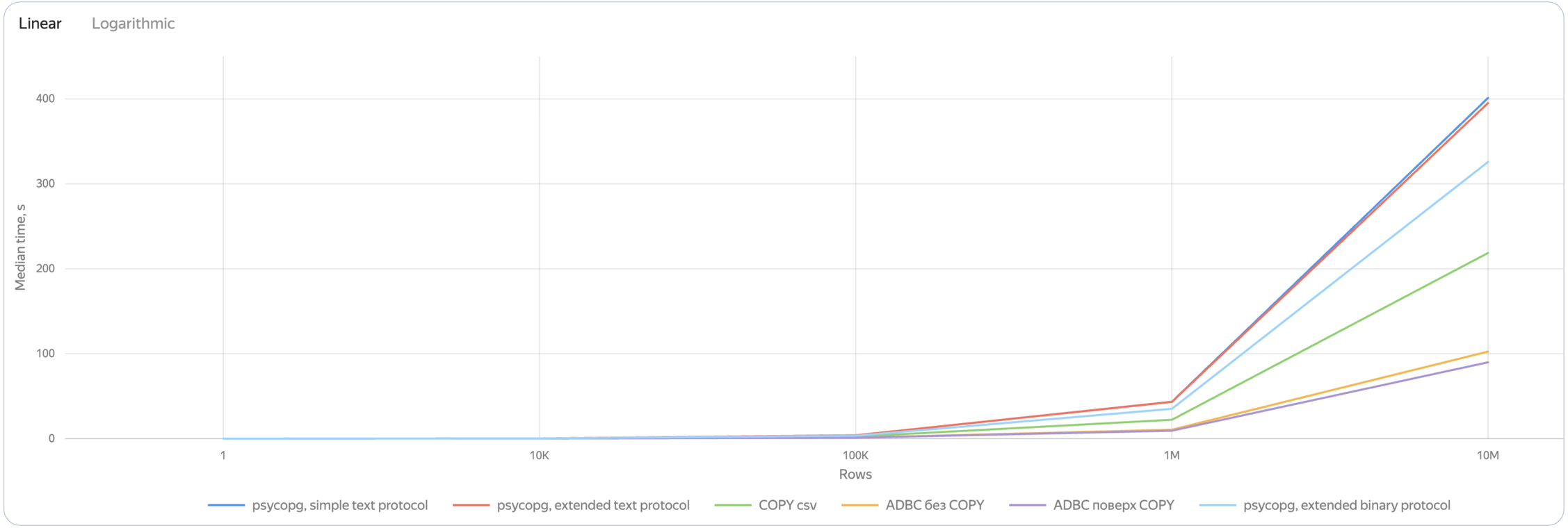
Время в % от text (100%). Меньше — быстрее. Все пути до pandas DataFrame.

PostgreSQL

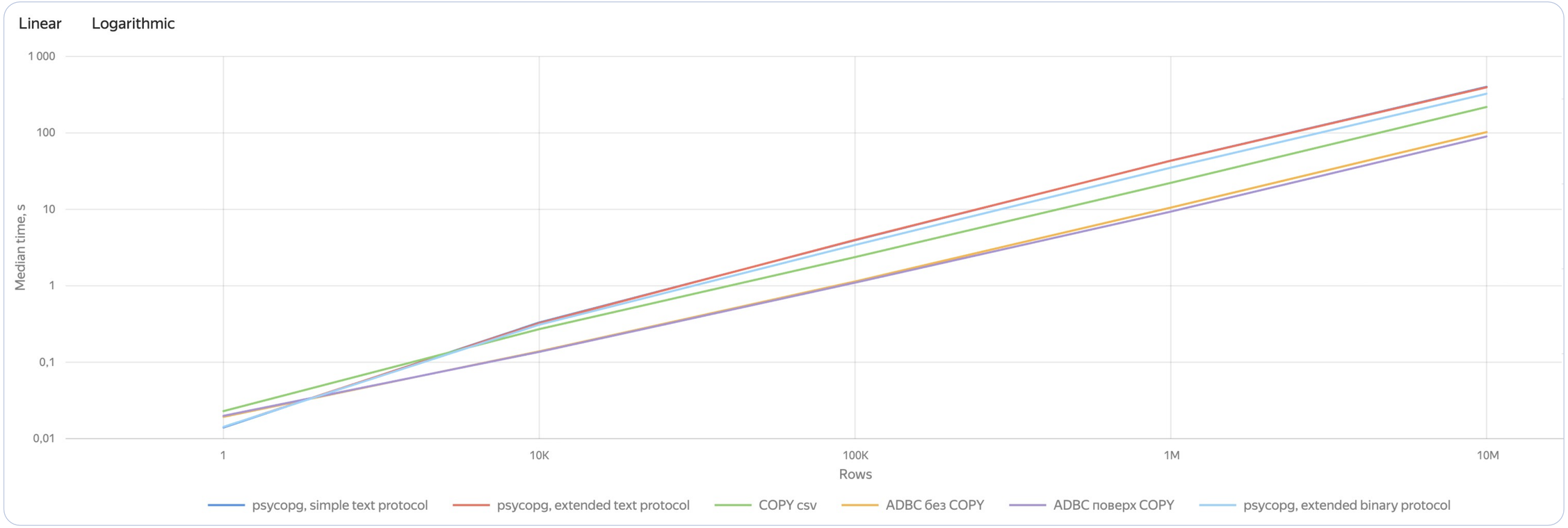
Строки/колонки	w10	w50	w105
10 тыс.	text 100% COPY CSV 158% ADBC 58%	text 100% COPY CSV 95% ADBC 44%	text 100% COPY CSV 82% ADBC 41%
100 тыс.	**text 100%** **COPY CSV 117%** **ADBC 42%** **Flight SQL* 67%**	text 100% COPY CSV 80% ADBC 33%	text 100% COPY CSV 60% ADBC 28%
1 млн	text 100% COPY CSV 113% ADBC 40% Flight SQL 79%	**text 100%** **COPY CSV 66%** **ADBC 26%**	text 100% COPY CSV 51% ADBC 22%
10 млн	text 100% COPY CSV 112% ADBC 43%	text 100% COPY CSV 63% ADBC 26%	**text 100%** **COPY CSV 55%** **ADBC 22%**

Время в % от text (100%). Меньше — быстрее. Все пути до pandas DataFrame.

PostgreSQL


Linear
Logarithmic
400
300
200
100
0
Median time, s
1
10K
100K
1M
10M
Rows
psycopg, simple text protocol
psycopg, extended text protocol
COPY csv
ADBC без COPY
ADBC поверх COPY
psycopg, extended binary protocol

PostgreSQL


Linear
Logarithmic
1 000
100
10
1
0,1
0,01
Median time, s
1
10K
100K
1M
10M
Rows
psycopg, simple text protocol
psycopg, extended text protocol
COPY csv
ADBC без COPY
ADBC поверх COPY
psycopg, extended binary protocol

ClickHouse®

Путь	Кортежи	Arrow	Pandas	Polars
Native TCP, ClickHouse®-driver	Да	—	Да	Да
HTTP Native, ClickHouse®-connect	Да	—	Да	Да
HTTP ArrowStream	—	Да	Да	Да
HTTP Parquet	—	Да	Да	Да
HTTP RowBinary	—	—	—	—
HTTP CSV	—	—	Да	Да
HTTP JSONEachRow	Да	Да	Да	Да
ADBC ClickHouse® по HTTP	—	Да	Да	Да
Flight SQL	—	Да (w10)	Да (w10)	Да (w10)
pg-wire эмуляция, psycopg	Да	—	Да	Да

ClickHouse®

Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 0,049 с / 4 МБ HTTP Native 0,045 с / 4 МБ ArrowStream 0,034 с / 4 МБ	Native TCP 0,144 с / 5 МБ HTTP Native 0,090 с / 5 МБ ArrowStream 0,060 с / 5 МБ	Native TCP 0,246 с / 6 МБ HTTP Native 0,190 с / 7 МБ ArrowStream 0,092 с / 7 МБ
100 тыс.	Native TCP 0,411 с / 40 МБ HTTP Native 0,387 с / 40 МБ ArrowStream 0,173 с / 41 МБ Flight SQL 0,244 с / 41 МБ	Native TCP 1,21 с / 49 МБ HTTP Native 0,718 с / 50 МБ ArrowStream 0,289 с / 52 МБ	Native TCP 2,21 с / 60 МБ HTTP Native 1,65 с / 66 МБ ArrowStream 0,477 с / 74 МБ
1 млн	Native TCP 3,87 с / 328 МБ HTTP Native 3,27 с / 328 МБ ArrowStream 1,27 с / 336 МБ Flight SQL 1,87 с / 337 МБ	Native TCP 11,7 с / 407 МБ HTTP Native 6,16 с / 420 МБ ArrowStream 2,03 с / 444 МБ	Native TCP 20,2 с / 537 МБ HTTP Native 15,2 с / 602 МБ ArrowStream 3,39 с / 684 МБ
10 млн	Native TCP 38,8 с / 3029 МБ HTTP Native 32,9 с / 3028 МБ ArrowStream 14,5 с / 3108 МБ Flight SQL 23,37 с / 3112 МБ	Native TCP 115,5 с / 3859 МБ HTTP Native 62,3 с / 3985 МБ ArrowStream 22,5 с / 4228 МБ	Native TCP 212,3 с / 5076 МБ HTTP Native 152,9 с / 5824 МБ ArrowStream 35,6 с / 6640 МБ

ClickHouse®

Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 0,049 с / 4 МБ HTTP Native 0,045 с / 4 МБ ArrowStream 0,034 с / 4 МБ	Native TCP 0,144 с / 5 МБ HTTP Native 0,090 с / 5 МБ ArrowStream 0,060 с / 5 МБ	Native TCP 0,246 с / 6 МБ HTTP Native 0,190 с / 7 МБ ArrowStream 0,092 с / 7 МБ
100 тыс.	**Native TCP 0,411 с / 40 МБ** **HTTP Native 0,387 с / 40 МБ** **ArrowStream 0,173 с / 41 МБ** **Flight SQL 0,244 с / 41 МБ**	Native TCP 1,21 с / 49 МБ HTTP Native 0,718 с / 50 МБ ArrowStream 0,289 с / 52 МБ	Native TCP 2,21 с / 60 МБ HTTP Native 1,65 с / 66 МБ ArrowStream 0,477 с / 74 МБ
1 млн	Native TCP 3,87 с / 328 МБ HTTP Native 3,27 с / 328 МБ ArrowStream 1,27 с / 336 МБ Flight SQL 1,87 с / 337 МБ	**Native TCP 11,7 с / 407 МБ** **HTTP Native 6,16 с / 420 МБ** **ArrowStream 2,03 с / 444 МБ**	Native TCP 20,2 с / 537 МБ HTTP Native 15,2 с / 602 МБ ArrowStream 3,39 с / 684 МБ
10 млн	Native TCP 38,8 с / 3029 МБ HTTP Native 32,9 с / 3028 МБ ArrowStream 14,5 с / 3108 МБ Flight SQL 23,37 с / 3112 МБ	Native TCP 115,5 с / 3859 МБ HTTP Native 62,3 с / 3985 МБ ArrowStream 22,5 с / 4228 МБ	**Native TCP 212,3 с /5076 МБ** **HTTP Native 152,9 с /5824 МБ** **ArrowStream 35,6 с /6640 МБ**

ClickHouse®

Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 100% HTTP Native 92% ArrowStream 70%	Native TCP 100% HTTP Native 63% ArrowStream 42%	Native TCP 100% HTTP Native 77% ArrowStream 37%
100 тыс.	Native TCP 100% HTTP Native 94% ArrowStream 42% Flight SQL 59%	Native TCP 100% HTTP Native 59% ArrowStream 24%	Native TCP 100% HTTP Native 75% ArrowStream 22%
1 млн	Native TCP 100% HTTP Native 84% ArrowStream 33% Flight SQL 48%	Native TCP 100% HTTP Native 53% ArrowStream 17%	Native TCP 100% HTTP Native 75% ArrowStream 17%
10 млн	Native TCP 100% HTTP Native 85% ArrowStream 37% Flight SQL 60%	Native TCP 100% HTTP Native 54% ArrowStream 19%	Native TCP 100% HTTP Native 72% ArrowStream 17%

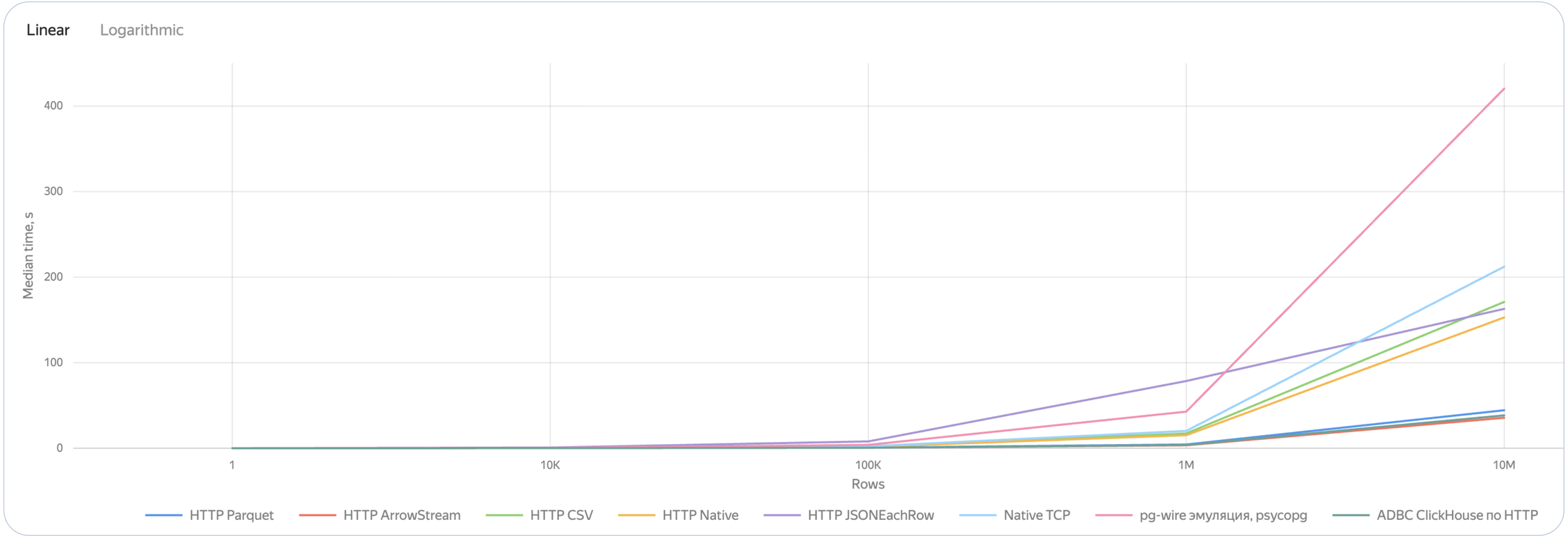
Время в % от Native TCP (100%). Меньше — быстрее. Все пути до pandas DataFrame.

ClickHouse®

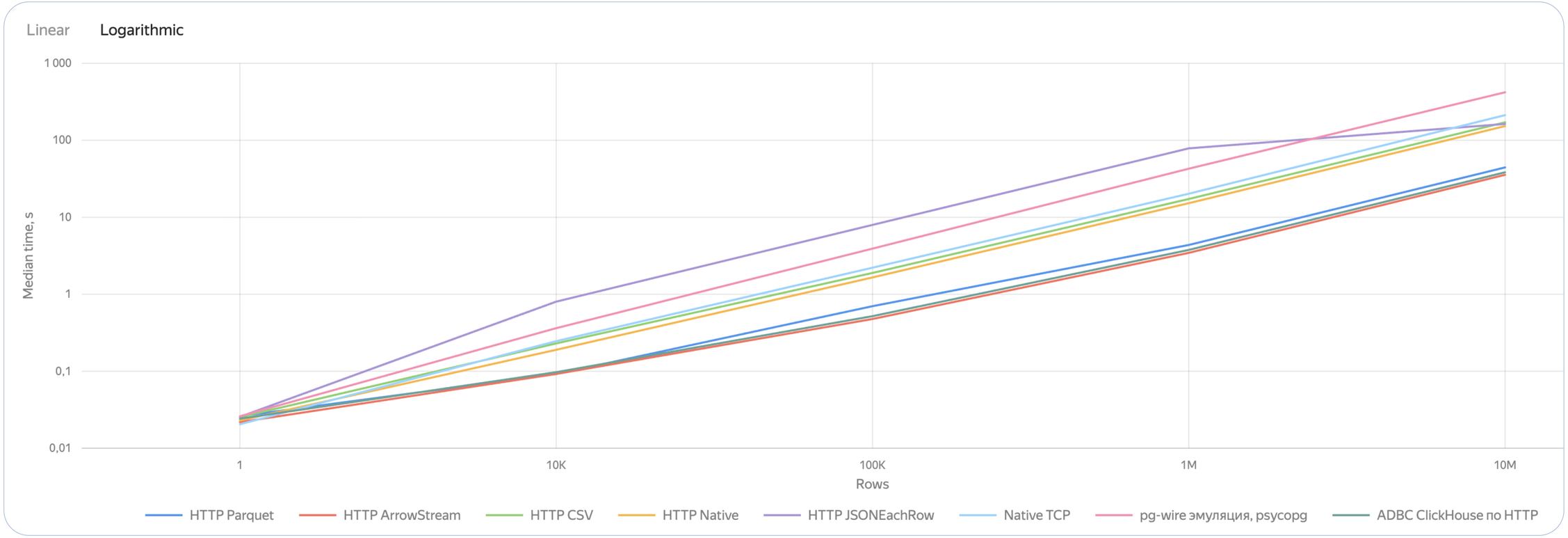
Строки/колонки	w10	w50	w105
10 тыс.	Native TCP 100% HTTP Native 92% ArrowStream 70%	Native TCP 100% HTTP Native 63% ArrowStream 42%	Native TCP 100% HTTP Native 77% ArrowStream 37%
100 тыс.	**Native TCP 100%** **HTTP Native 94%** **ArrowStream 42%** **Flight SQL 59%**	Native TCP 100% HTTP Native 59% ArrowStream 24%	Native TCP 100% HTTP Native 75% ArrowStream 22%
1 млн	Native TCP 100% HTTP Native 84% ArrowStream 33% Flight SQL 48%	**Native TCP 100%** **HTTP Native 53%** **ArrowStream 17%**	Native TCP 100% HTTP Native 75% ArrowStream 17%
10 млн	Native TCP 100% HTTP Native 85% ArrowStream 37% Flight SQL 60%	Native TCP 100% HTTP Native 54% ArrowStream 19%	**Native TCP 100%** **HTTP Native 72%** **ArrowStream 17%**

Время в % от Native TCP (100%). Меньше — быстрее. Все пути до pandas DataFrame.

ClickHouse®

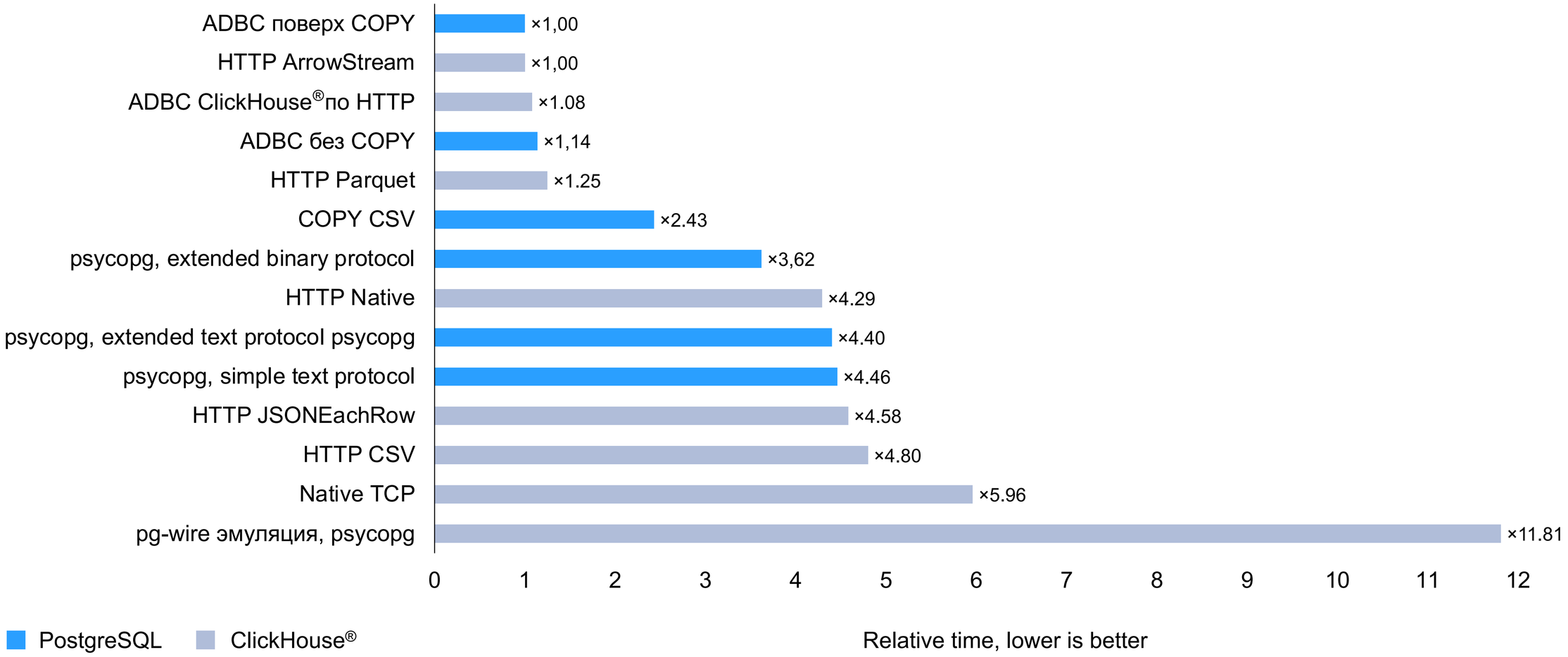

Linear
Logarithmic
400
300
200
100
0
Median time, s
1
10K
100K
1M
10M
Rows
HTTP Parquet
HTTP ArrowStream
HTTP CSV
HTTP Native
HTTP JSONEachRow
Native TCP
pg-wire эмуляция, psycopg
ADBC ClickHouse по HTTP

ClickHouse®


Linear
Logarithmic
1 000
100
10
1
0,1
0,01
Median time, s
1
10K
100K
1M
10M
Rows
HTTP Parquet
HTTP ArrowStream
HTTP CSV
HTTP Native
HTTP JSONEachRow
Native TCP
pg-wire эмуляция, psycopg
ADBC ClickHouse по HTTP

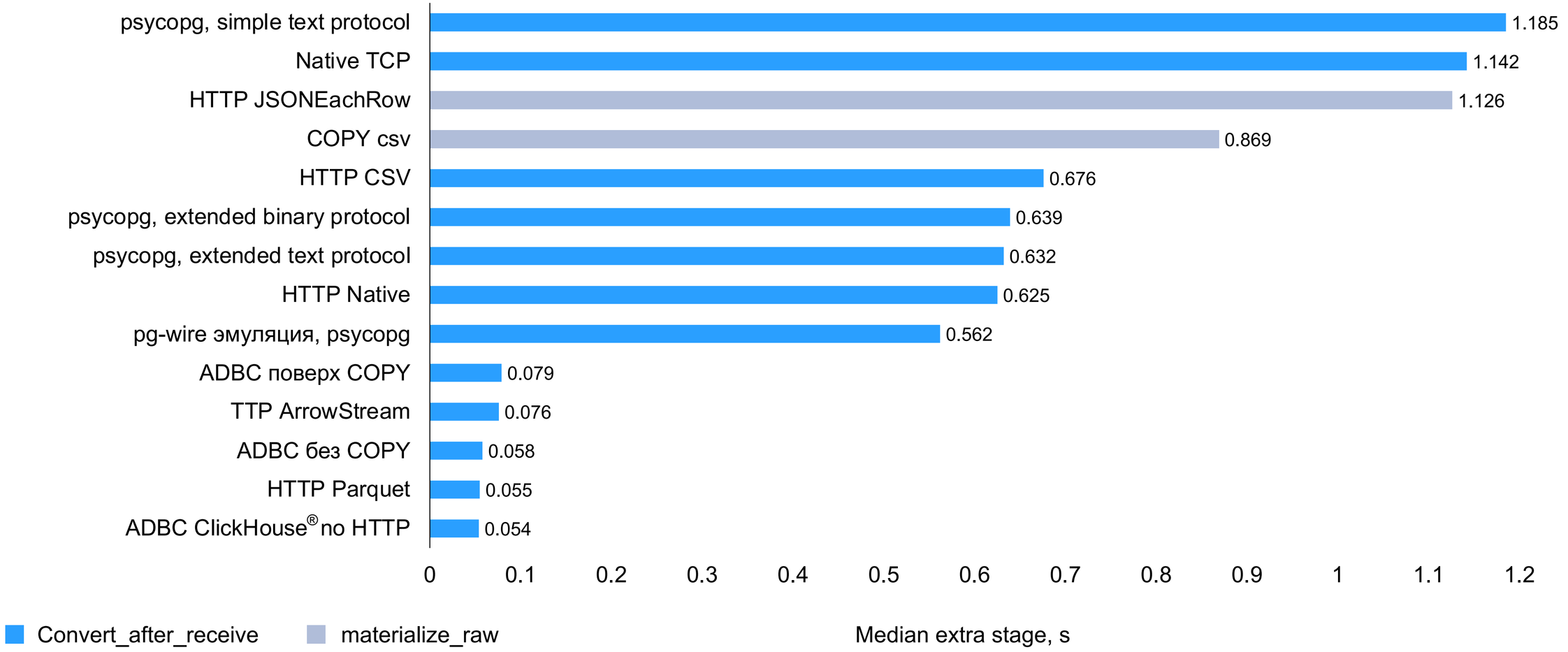
Сравним производительность

Access method rating at largest selected dataset

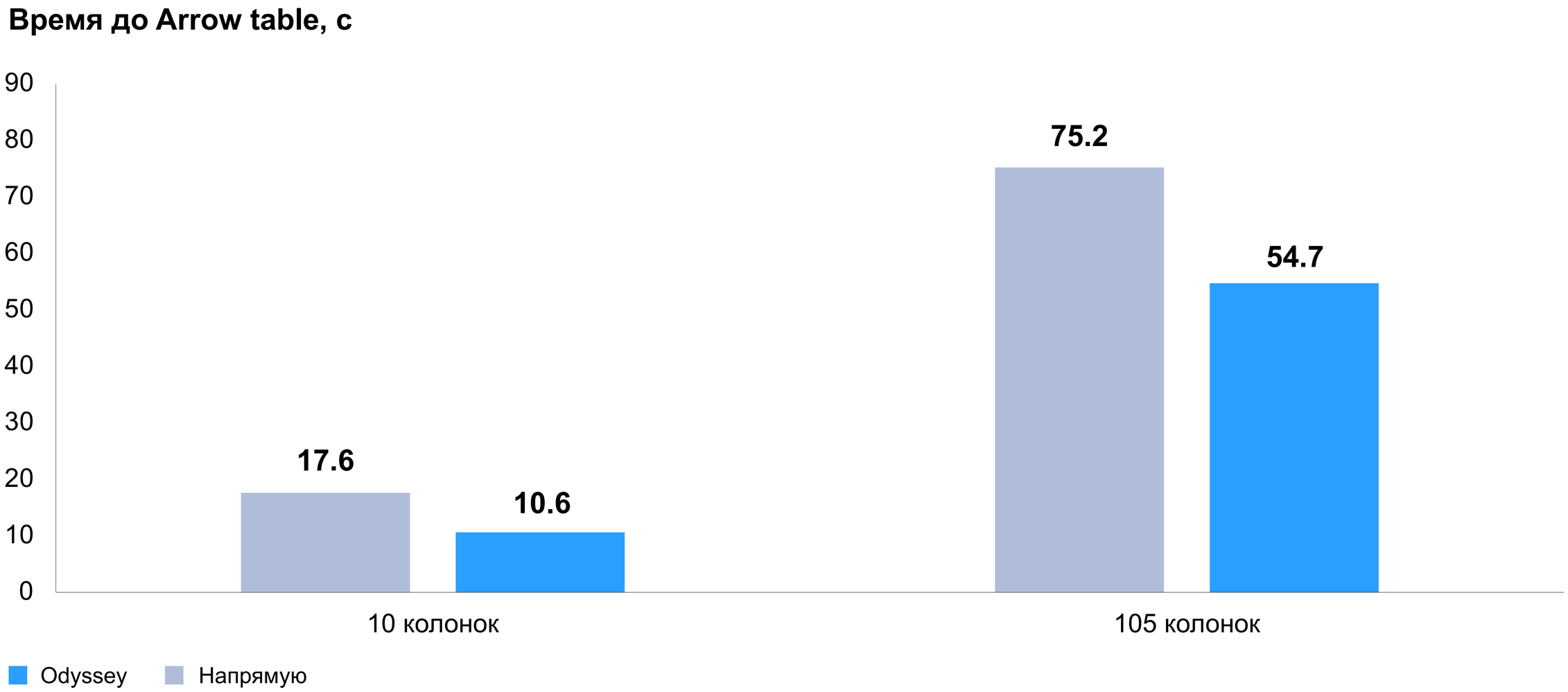


Сериализация

Extra stage time



Пуллеры и время получения результата



Итоги

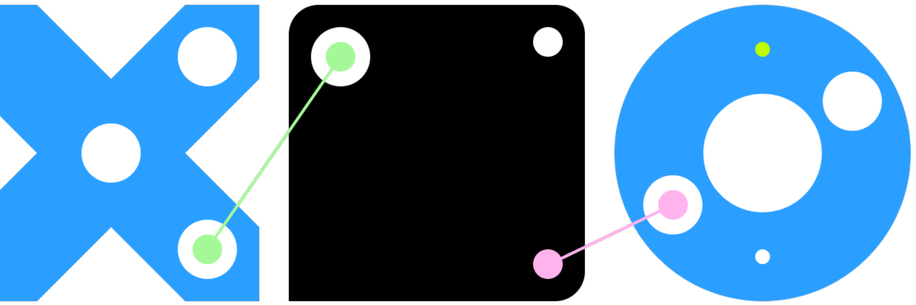
Нативные навыки ADBC

Инструмент	Что именно	С какой версии
Pandas	Поддерживает соединение ADBC	2.2.0
Polars	Поддерживает соединение ADBC	Все
DuckDB	ADBC-драйвер внутри libduckdb. Расширение adbc_scanner — DuckDB как ADBC-клиент	
dbt	В dbt v2 (на Rust) ADBC стал единым слоем подключения вместо самописной логики адаптеров	dbt Core v2
Power BI и Microsoft Fabric	Встроенные ODBC-драйверы коннекторов заменяются на ADBC	По умолчанию с августа 2026 года, ODBC убирают из сервиса к концу 2026-го
Apache Airflow®	Провайдер apache-airflow-providers-apache-arrow с AdbcHook	0.1.0, релиз 2026-09-14
Библиотеки	Python®, R, Java™, Go, C#, Rust и т. д.	Релиз ADBC 24

Где просили, но ADBC не завезли

Инструмент	Тикет	Состояние
DBeaver	dbeaver/dbeaver#15535	Открыт с 2022-02-17, метка New driver
DataGrip	YouTrack DBE-16742	Submitted с 2022-11-11, не разрёшен
Trino	trinodb/trino#18038, #24586	Открыты с 2023-го и 2024 года
SQLAlchemy™	sqlalchemy#11742	Закрыт: мейнтейнер считает, что снаружи ADBC должен жить колоночный диалект, SQLAlchemy™ — построчный
Apache Spark™	arrow-adbc#1801	Закрыт как not-planned-ставка на Spark Connect
Excel	—	В Excel остаётся ODBC

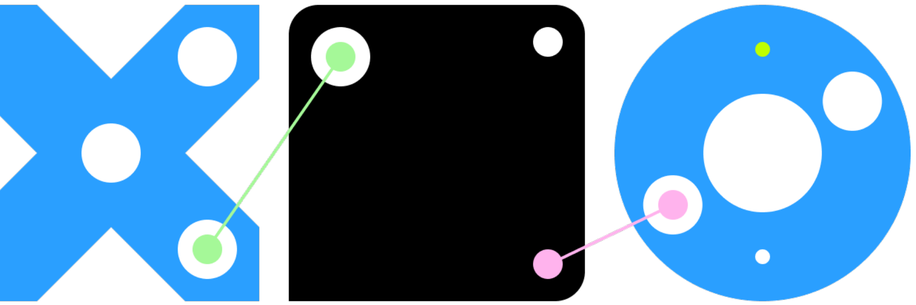
Вернёмся к «Hold my data hostage…»



Raasveldt, Muehleisen: Don’t hold my data hostage, Table 1. PVLDB 10(10) (2017).

Разные операции. Netcat льёт готовый CSV в /dev/null, SQL-клиент строит типизированный результат. Отношение ×10 — про передачу данных по сети, не про готовый объект

Вернёмся к «Hold my data hostage…»



Raasveldt, Muehleisen: Don’t hold my data hostage, Table 1. PVLDB 10(10) (2017).

Разные операции. Netcat льёт готовый CSV в /dev/null, SQL-клиент строит типизированный результат. Отношение ×10 — про передачу данных по сети, не про готовый объект

Обоих авторов этой статьи вы знаете: Марк Раасвельдт и Ханнес Мюлайзен — это основатели **DuckDB**

DuckDB выпустили свой протокол **Quack** в мае 2026 года.
Quack — не Arrow.
Своя сериализация поверх HTTP

Что важно при сравнении

Весь путь до результата

Время включает протокол, драйвер и сборку

Да-да, мы проверяли сценарий с Python®



Метаданные и служебные запросы

Учитывайте их время, особенно при чтении небольших результатов

А что с записью?

В этих прогонах измеряли чтение. Скорость и **корректность** записи нужно проверять отдельно



Как воспроизвести



datalens.yandex/
bdpws4a4b718u



github.com/petergurinov/
between-sql-and-data



storage.yandexcloud.net/
between-sql-and-data-bench-
files/manifest.json

Спасибо
за внимание!



Петр Гуринов

Команда продуктовой архитектуры,
Data Platform

Cloud