

Что скрывается между сетью и плюсами?

Шагойко Алексей

План

Введение. Intro to High-Frequency Trading

1. Что представляет из себя биржа и торговля на ней
2. Алгоритмический и высокочастотный трейдинг
3. Как устроена HFT-инфраструктура

Network Stack. Что скрывается между плюсами и сетью?

1. Что такое socket и как он работает
2. Как улучшить получение данных
3. Что происходит внутри ядра и сетевой карты

Что такое HFT: Ручная торговля раньше и сейчас

Процесс торговли на бирже в
70-80-ых годах XX века:

Медленно

Потеря информации

Чем меньше сделка, тем она
менее выгодная

Процесс торговли на бирже
сейчас:

быстро

нет потерь информации

не зависит от размера сделки

Что такое HFT: Алгоритмический трейдинг

Алгоритмический – это общее название процесса применения запрограммированных математических алгоритмов для автоматического трейдинга.

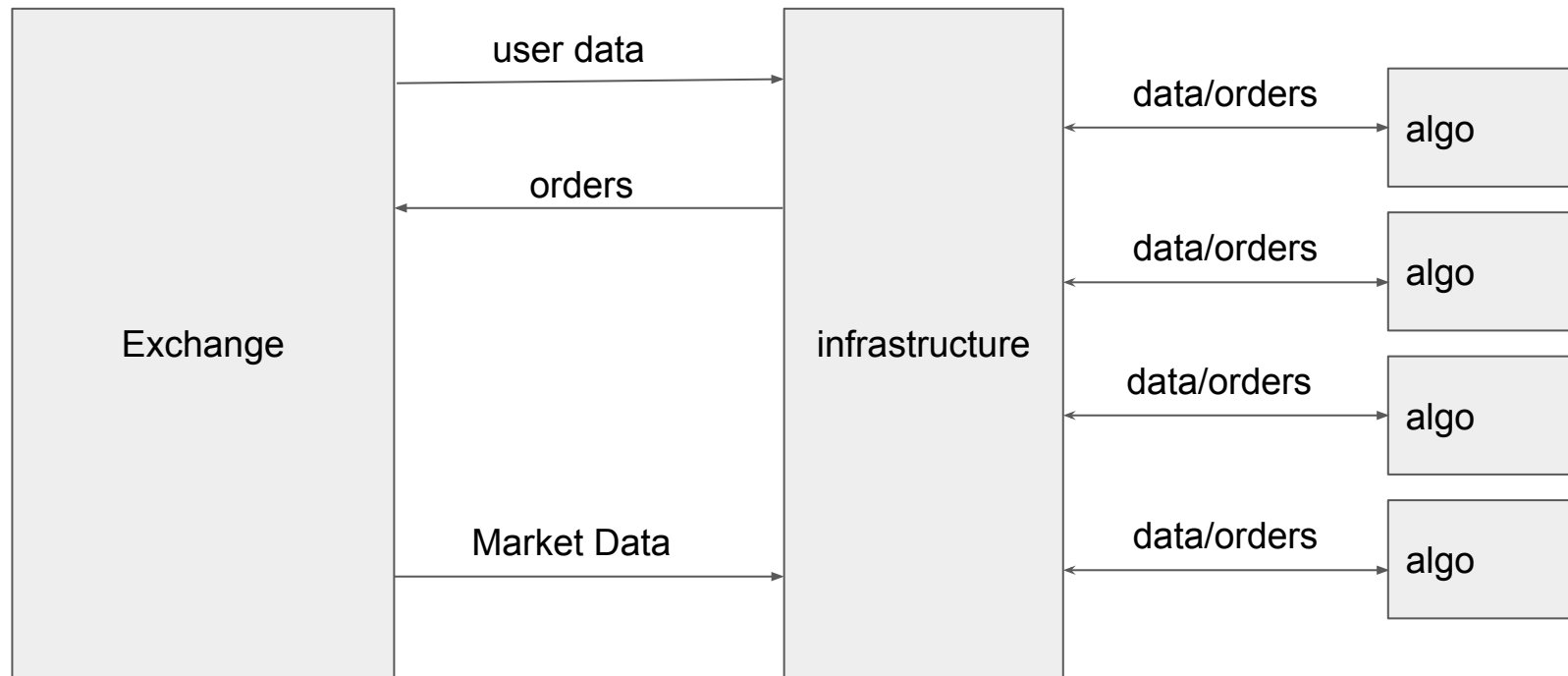
Высокочастотный трейдинг (*с англ. «HFT, High-frequency trading»*) — это основной вид автоматической торговли с использованием алгоритмов, используемый для сверхбыстрых операций с инструментами на фондовом рынке

Важно о HFT

Высокочастотный трейдинг:

- прежде всего технологическая область, к экономике не имеет практически никакого отношения.
- место обитания многих хардкорных C++-разработчиков.
- закрытая и высококонкурентная сфера, где выживают только профессиональные команды.
- одно из самых технологически емких направлений в IT.

Торговая инфраструктура

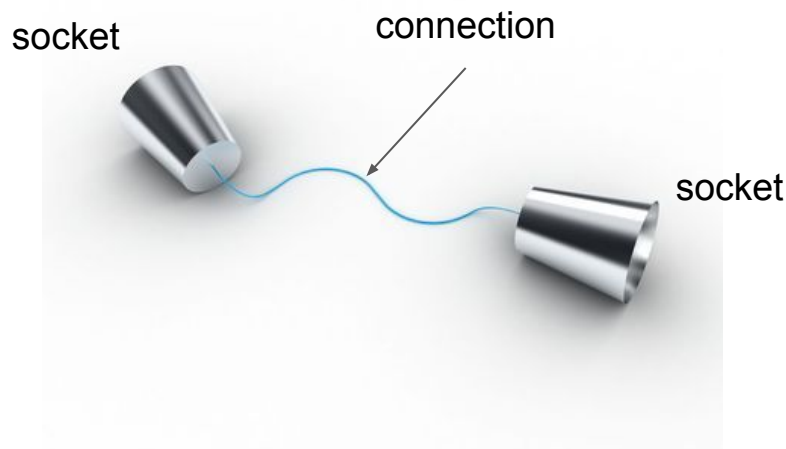


Что скрывается между сетью и плюсами?

Шагойко Алексей

Что такое socket

Интернет сокет - это **абстракция сетевого взаимодействия в операционной системе Linux**. Каждому сокету соответствует пара IP-адрес + номер порта



Что такое socket

Интернет сокет - это **абстракция сетевого взаимодействия в операционной системе Linux**. Каждому сокету соответствует пара IP-адрес + номер порта

```
int fd = socket(int domain, int type, int protocol);
```

domain - семейство протоколов, которое будет использоваться для соединения.

Например, PF_INET - IPv4, PF_INET6 - IPv6 и тп

type - тип сокета, который определяет семантика общения.

Например, SOCK_DGRAM - ненадежные, упакованные данные. SOCK_STREAM - надежный последовательный поток сообщений

protocol - зависимости от type может быть определен по разному.

В большинстве случаев 0

Пример сервер \Leftrightarrow клиент

сервер		клиент
Установка сокета socket()		Установка сокета socket()
Присвоение имени (ip/port) bind()		
Установка очереди запросов listen()		
Выбор соединения из очереди accept()		Установка соединения connect()
recv()		send()
send()		recv()

В чем проблема данной реализации?

recv/read в данном случае является блокирующей операцией. То есть поток при вызове замирает в ожидании сообщения. Нельзя получать сообщения по нескольким соединениям

Nonblock recv/read

```
int fcntl(int fd, int cmd, long arg);
```

fd - файловый дескриптор. В данном случае дескриптор сокета, для которого нужно включить неблокирующий режим

cmd - операция, которую нужно сделать с параметром (set, get)

arg - сам параметр

```
int error = fcntl(sockfd, F_SETFL, O_NONBLOCK);
```

Nonblock recv/read

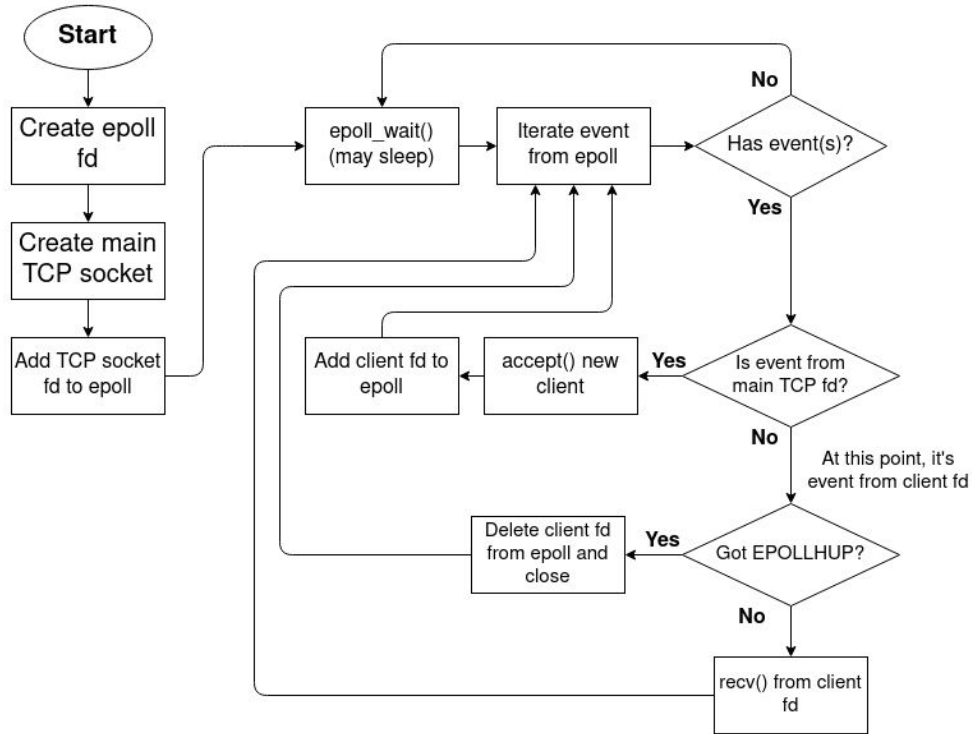
Плюсы:

- Поток не будет ждать сообщения, а сразу скажет, что прочитано 0 байт

Минусы:

- Активное ожидание (минус в зависимости от ситуации)

Epoll



Epoll

Плюсы:

- можно реагировать сразу на нужный сокет

Минусы:

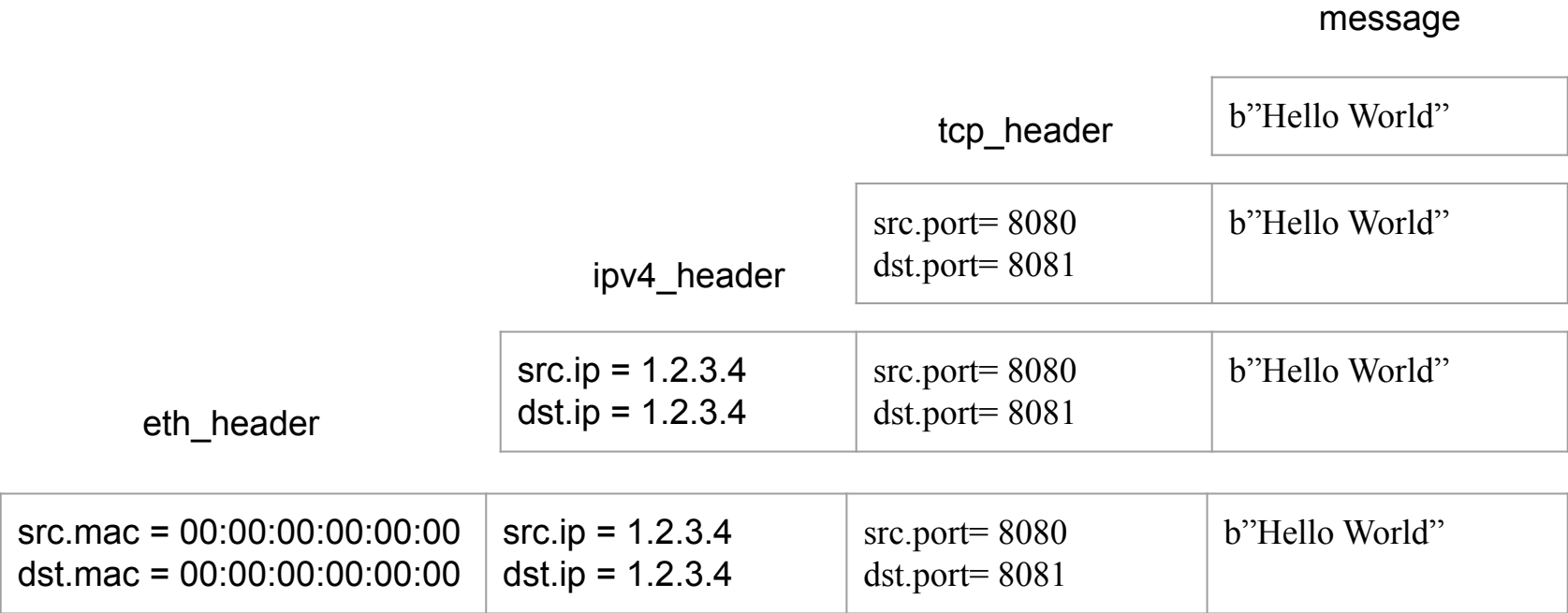
- засыпание на `epoll_wait`

Как сообщения ходят по сети?

модели OSI и TCP/IP

OSI	TCP/IP	protocols
Application	Application layer	Http, WebSocket, TLS, FIX etc
Presentation		
Session		
Transport	Transport layer	TCP, UDP
Network	Internet layer	IPv4, IPv6
Data link	Link layer	Ethernet
Physical		—

Сообщение “hello world”

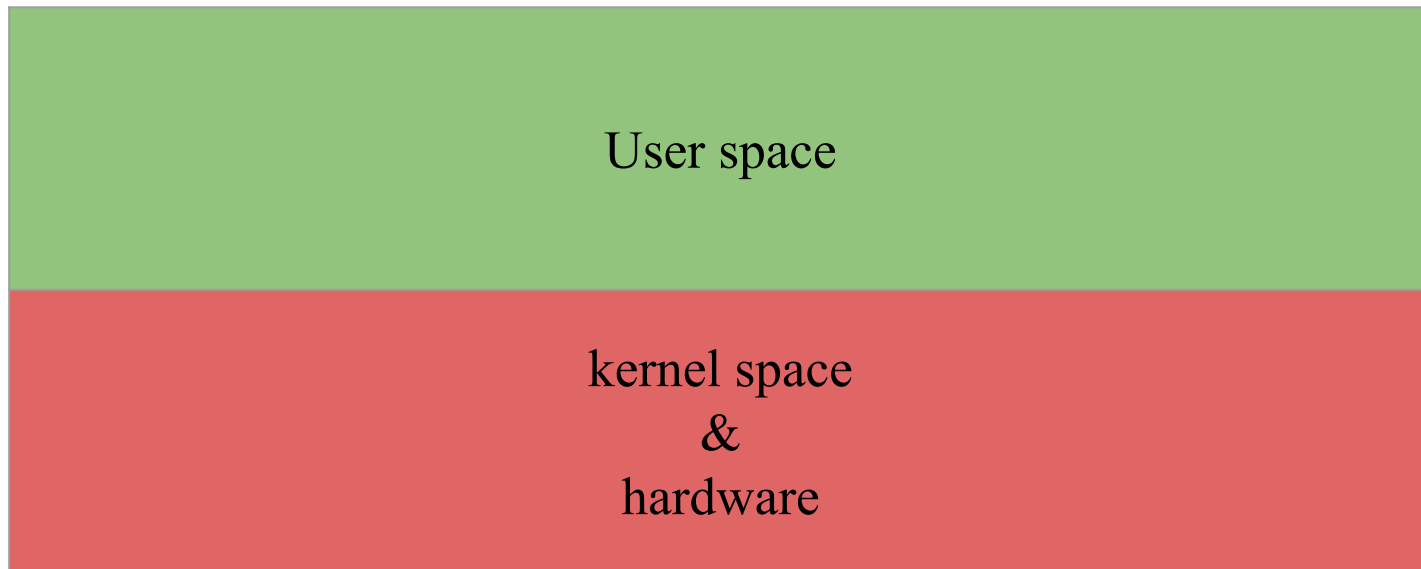


Wireshark

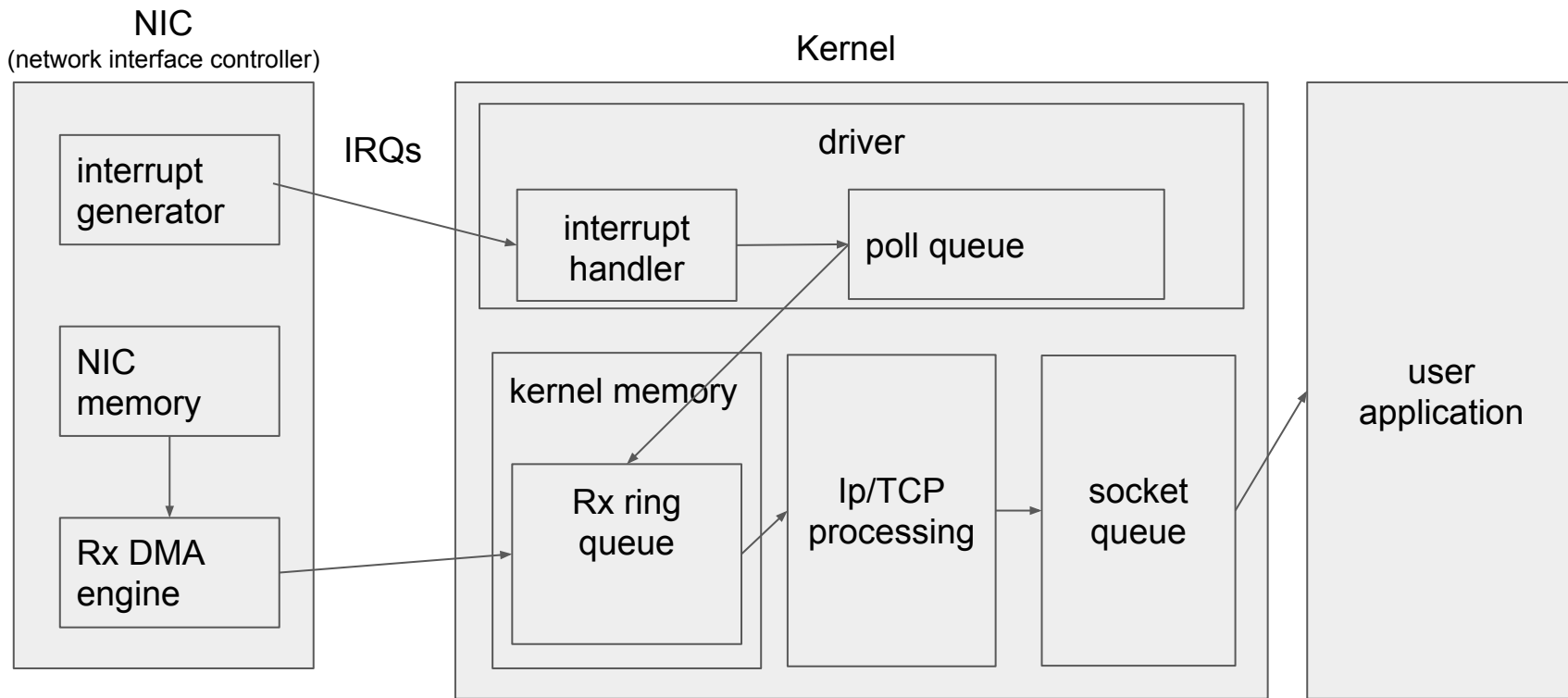
Wireshark — инструмент для захвата и анализа сетевого трафика



Погрузимся немного глубже

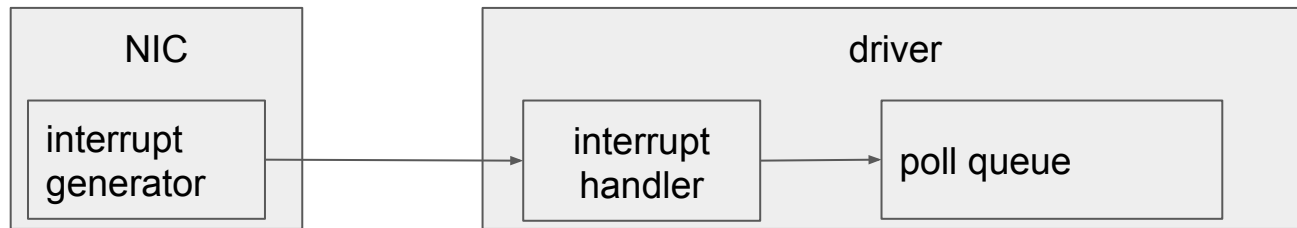


Обработка пришедшего сетевого пакета



Момент получение пакета nic'ом

1. Сетевая карта принимает сигнал.
2. Сетевая карта через DMA копирует свою память с пакетом в оперативную память.
3. IRQ (interrupt request)
 1. После чего выполняет прерывание (IRQ, что видно в **/proc/interrupts**), которое оповестит ядро о том что пакет пришел.
 2. Вызывается функция (interrupt handler), которую зарегистрировал драйвер сетевой карты при инициализации.
 3. В контексте обработчика прерывания IRQ выполняется пометка того, что пакеты пора обрабатывать

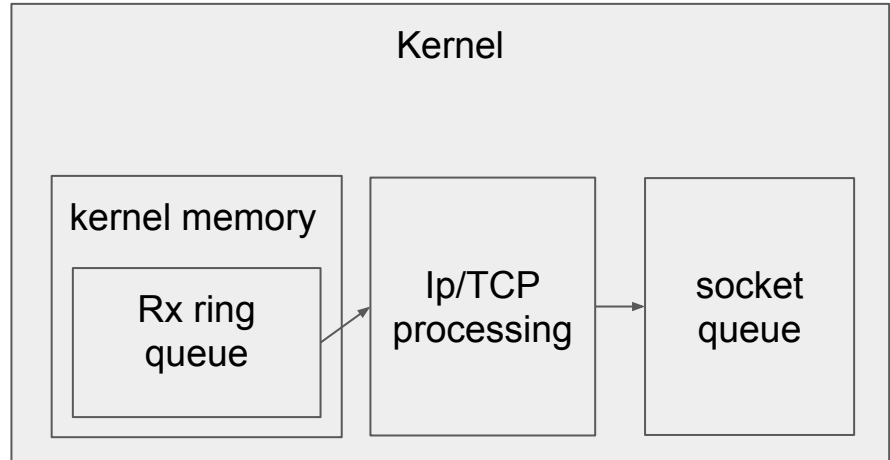


NAPI

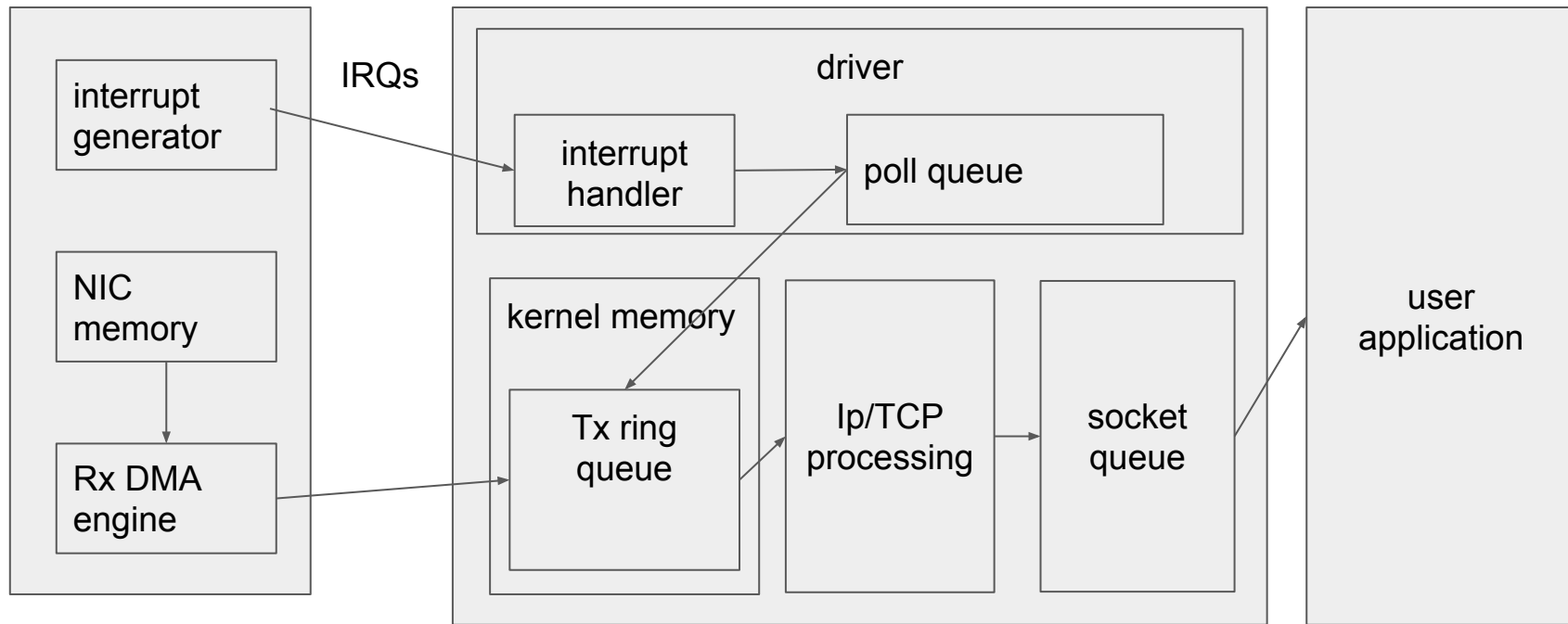
Основная цель NAPI - сократить количество прерываний, генерируемых при получении пакетов. В NAPI механизм прерываний сочетается с механизмом опроса. Чаще всего в разработке избегают использования поллинга, так как могут тратиться лишние ресурсы, когда оборудование молчит. У высоконагруженных интерфейсов такой проблемы не возникает

SoftIRQ

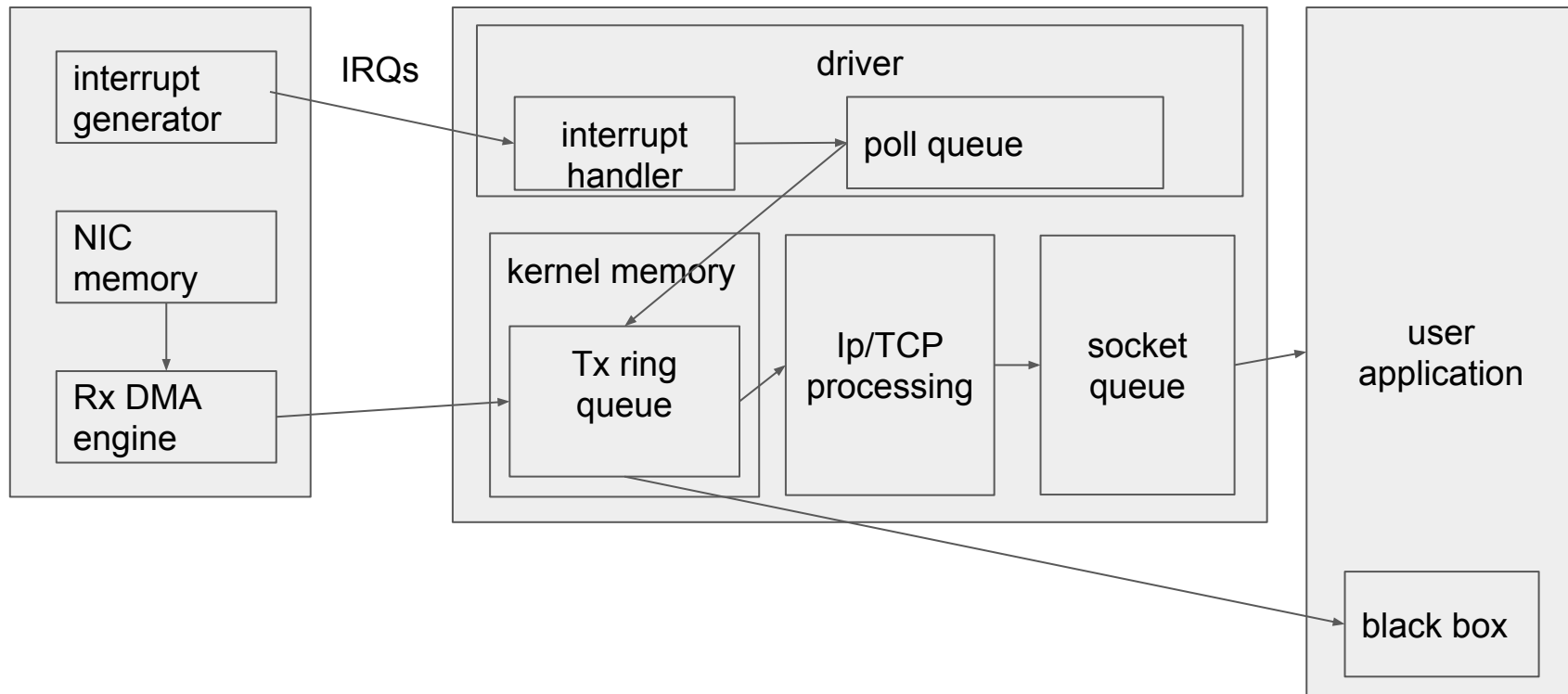
SoftIRQ - программное прерывание, в данном случае сигнализирует о появлении новых данных



Где можно срезать?



Можно ли так?



Sock Raw

SOCK_RAW - тип socket, который позволяет читать сырые данные (с заголовками низкоуровневых протоколов).

```
int fd = socket(int domain, SOCK_RAW, int protocol);
```

В качестве domain можно использовать PF_INET, PF_INET6, PF_PACKET (сырой трафик с eth_header)

Чего мы добились с SOCK_RAW?

Плюсы

1. умеем видеть данные с сырыми заголовками и получать их без парсинга в ядре

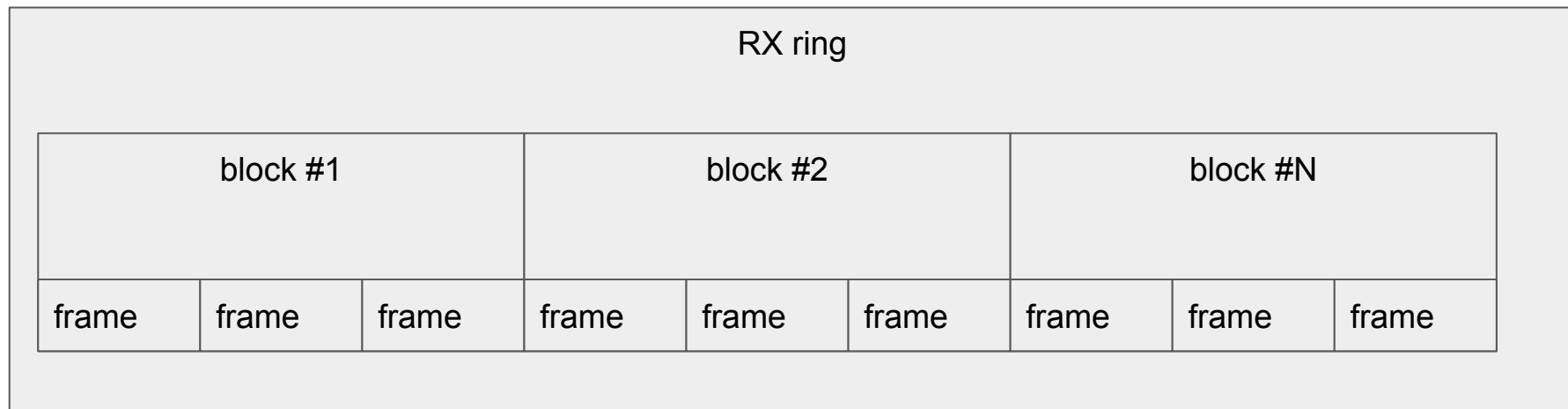
Минусы

1. Теперь хедеры нужно парсить в user space
2. все так же нужно делать syscall (recv/read) для получения данных из Rx ring

Можно ли лучше?

PACKET_RX_RING

PACKET_RX_RING - параметр сокета, который позволяет замапить участок памяти с сетевыми пакетами



block - физически не разрывной кусок памяти, который может содержать в себе целое число frame

frame - сетевой пакет

Вопрос