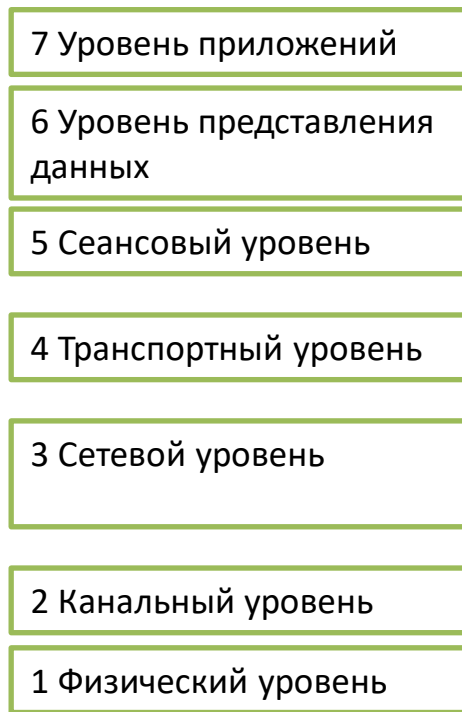
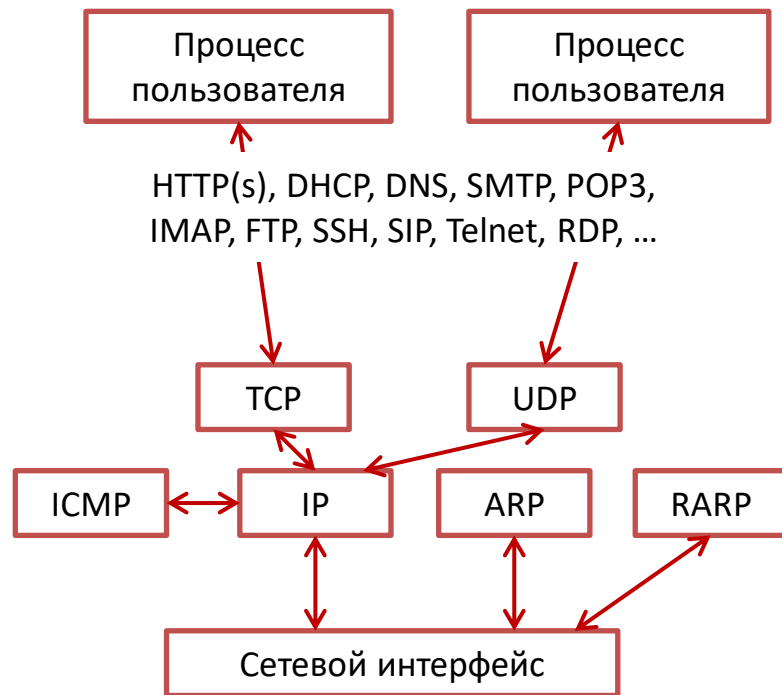
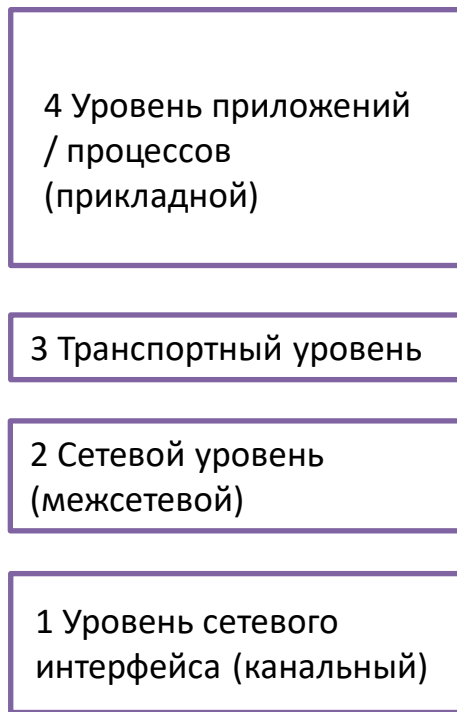


# Сетевые протоколы

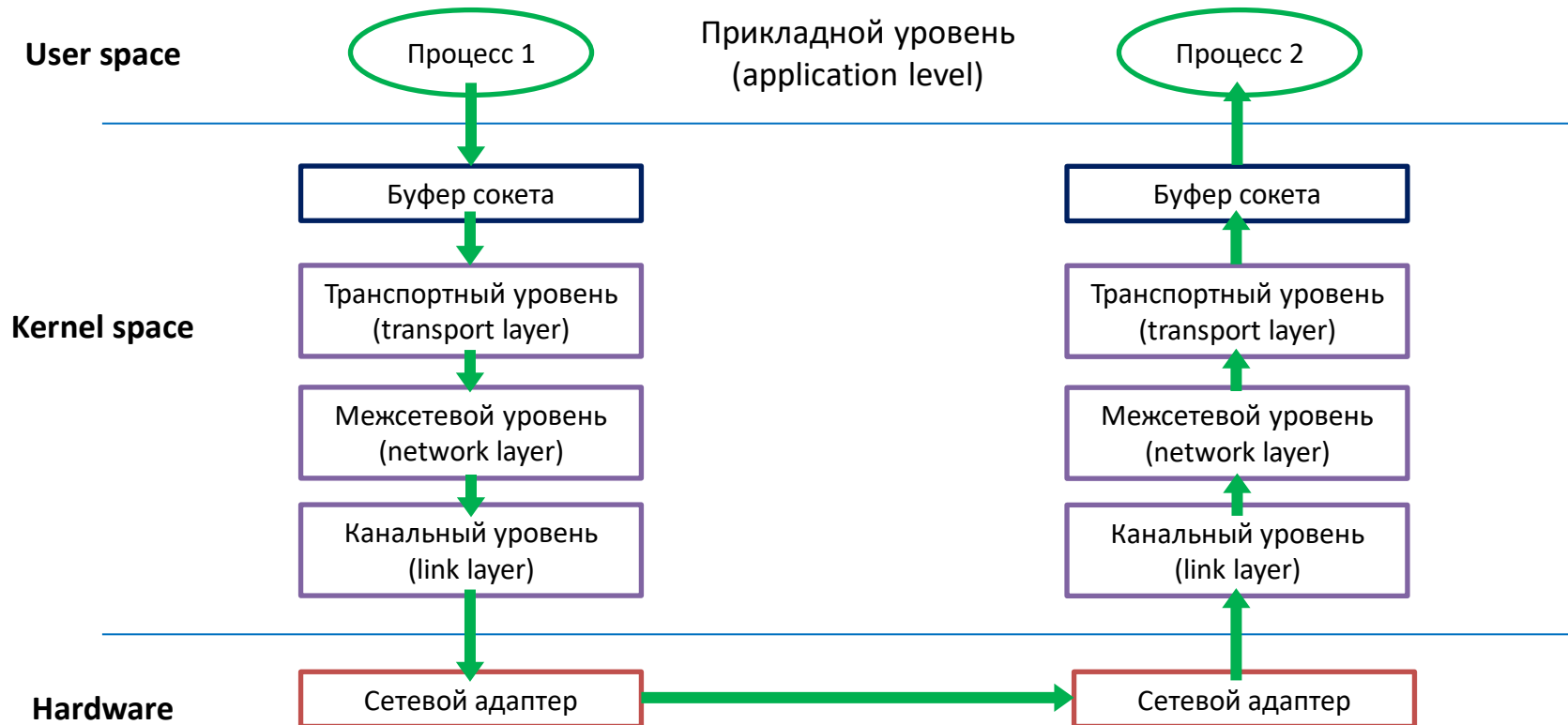
## Модель OSI/ISO



## Модель TCP/IP

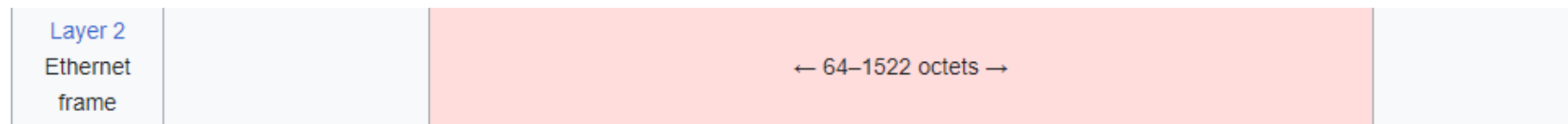
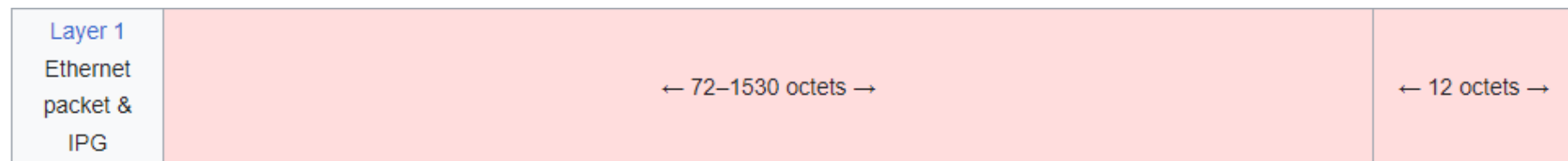


# Модель передачи данных



# Заголовок Ethernet (IEEE 802.3)

Layer	Preamble	Start frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap (IPG)
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets



# Заголовок IP v4

Отступ	Октет	0								1								2								3							
Октет	Бит	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	0	Версия				Размер заголовка				Differentiated Services Code Point				Explicit Congestion Notification				Размер пакета (полный)															
4	32	Идентификатор																Флаги		Смещение фрагмента													
8	64	Время жизни								Протокол								Контрольная сумма заголовка															
12	96	IP-адрес источника																															
16	128	IP-адрес назначения																															
20	160	Опции (если размер заголовка > 5)																															
20 или 24+	160 или 192+	Данные																															

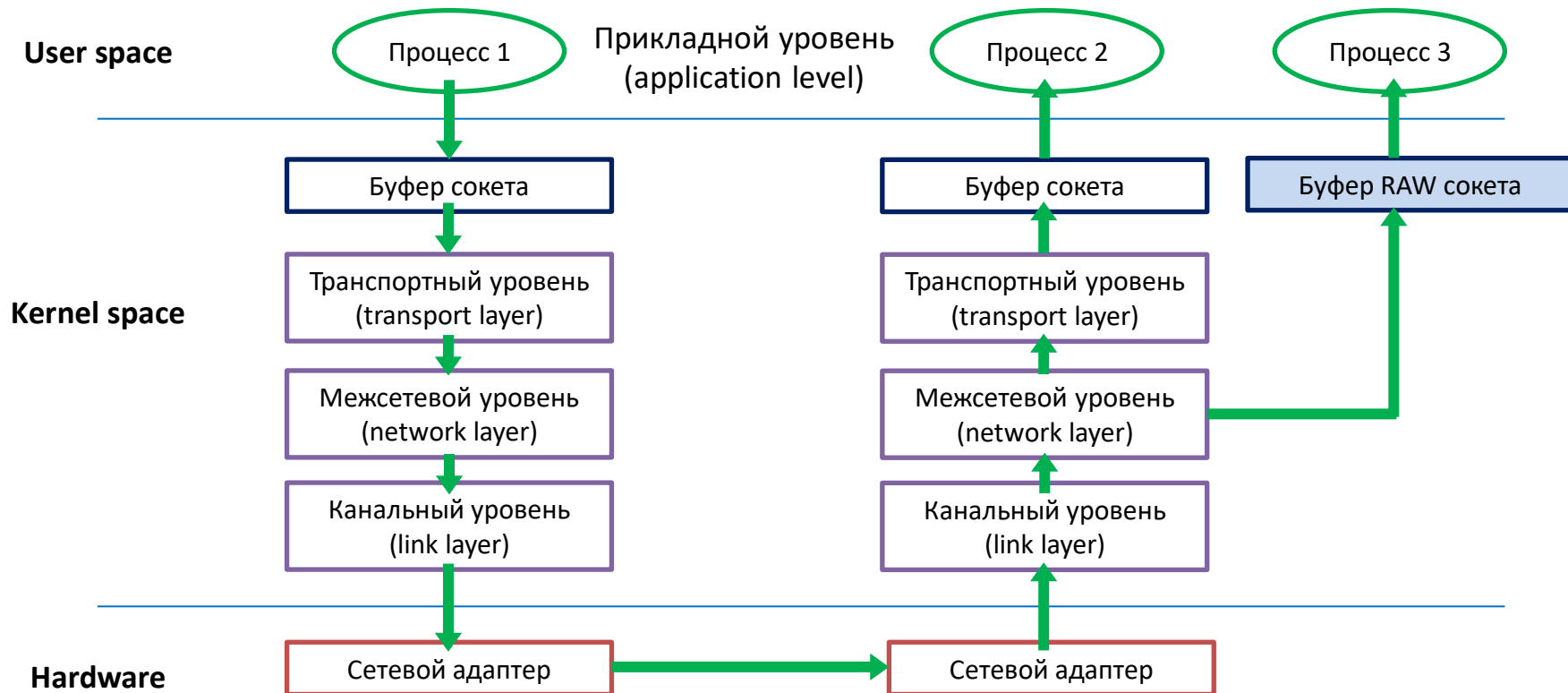
# Заголовок UDP

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

# Заголовок TCP

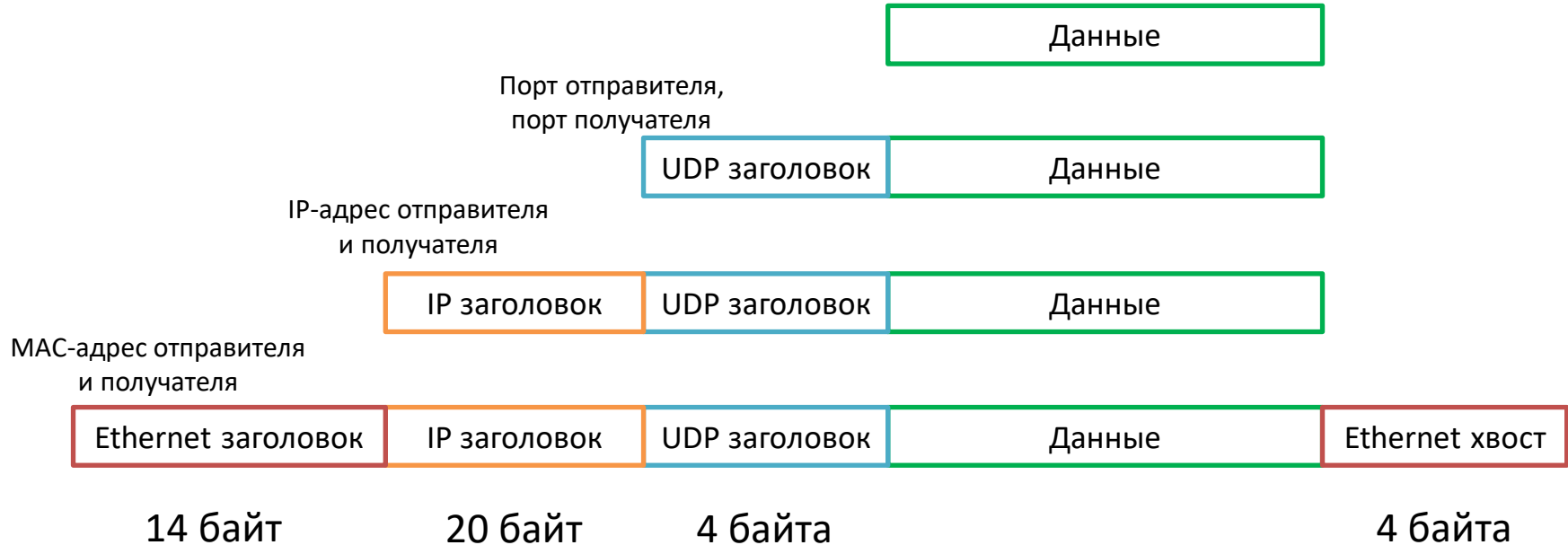
Бит	0 — 3	4 — 6	7 — 15	16 — 31
0	Порт источника, <b>Source Port</b>			Порт назначения, <b>Destination Port</b>
32	Порядковый номер, <b>Sequence Number (SN)</b>			
64	Номер подтверждения, <b>Acknowledgment Number (ACK SN)</b>			
96	Длина заголовка, <b>(Data offset)</b>	Зарезервировано	Флаги	Размер Окна, <b>Window size</b>
128	Контрольная сумма, <b>Checksum</b>			Указатель важности, <b>Urgent Point</b>
160	Опции (необязательное, но используется практически всегда)			
160/192+	Данные			

# RAW сокеты



# Сетевые протоколы

## Инкапсуляция для UDP в Ethernet





# RAW сокет

- `man 7 raw`
- **Создание сокета**  

```
#include <sys/socket.h>  
int socket(int socket_family, int socket_type, int protocol);  
  
fd = socket(AF_INET, SOCK_RAW, IPPROTO_UDP);
```
- Входящий пакет дублируется (копируется) на все RAW сокет.
- При передаче с сетевого на транспортный уровень добавляется псевдозаголовок с двумя IP адресами и длиной пакета.
- При передаче с транспортного на сетевой уровень псевдозаголовок не добавляется.

# Псевдозаголовок для расчета контрольной суммы UDP (IP v4)

Биты	0 — 7	8 — 15	16 — 23	24 — 31
0	Адрес источника			
32	Адрес получателя			
64	Нули	Протокол	Длина UDP	
96	Порт источника		Порт получателя	
128	Длина		Контрольная сумма	
160+	Данные			

# RAW сокет

- При отправке данных (sendto) к буферу нужно добавить заголовок транспортного уровня.

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

- Контрольную сумму можно указать равной 0 (игнорируется в linux).
- Сетевой порядок байтов!
- При использовании RAW сокета на транспортном уровне может не быть endpoint-а.  
Тогда формируется ICMP сообщение от транспортного уровня о недоступности получателя (port-unreachable).

# Задания

- **6.1** (1 балл). Выполнить захват пакетов (wireshark), отправляемых клиентом и сервером из задачи 5.1.

Проанализировать структуру и содержание пакетов. На примере конкретных сообщений клиента и сервера указать содержимое полей в заголовках Ethernet, IP и UDP.

# Задания

- **6.2** (2 балла). Написать программу, получающую копии пакетов, предназначенных приложению-серверу из задачи 5.1 (снифер).

Программа может формировать дампы данных, который нужно вручную проанализировать (просмотр бинарных файлов) и убедиться, что это действительно информация от программы-клиента.

Можно дополнить программу так, чтобы она расшифровывала и выводила в файл или на экран полученные сообщения (+2 балла).

# Задания

- **6.3** (4 балла). Написать программу-клиент, использующую UDP RAW сокет и отправляющую сообщения серверу.

Для этого скорректировать программу-сервер из задачи 5.1 так, чтобы она только выводила сообщения клиента и не отвечала ему.