Компьютерные сети

Коротко о том,
как там всё устроено

Кафедра Алгоритмов и технологий программирования © 2017-2018

Уровни организации сети

Уровень TCP/IP	Уровень OSI	Примеры
Уровень процессов	Прикладной (Application)	HTTP, FTP, SSH, Telnet
	Представления (Presentation)	ASCII, GZIP, binary
	Сеансовый (Session)	NetBIOS, SSL
Транспортный уровень	Транспорта (Transport)	TCP, UDP
Уровень Internet	Сети (Network)	IPv4, IPv6, IPX, AppleTalk
Уровень сетевого интерфейса	Канала (Data)	PPP, IEEE 802.2 (Ethernet)
	Физический (Physical)	USB, I2C, IEEE 802.11 (WiFi), IEEE 802.3 (Ethernet)

Ethernet на уровне канала

MAC получа-теля	MAC отпра-вителя	Флаги опций	Длина фрейма	Данные	Контр-ольная сумма
6 байт	6 байт	4 байта	2 байта	от 46 до 1500 (параметр MTU) байт	4 байта

IPv4 (уровень транспорта)

MAC получа-теля	MAC отпра-вителя	Флаги опций	Длина фрейма	Данные	Контр-ольная сумма
6 байт	6 байт	4 байта	2 байта	от 46 до 1500 (параметр MTU) байт	4 байта

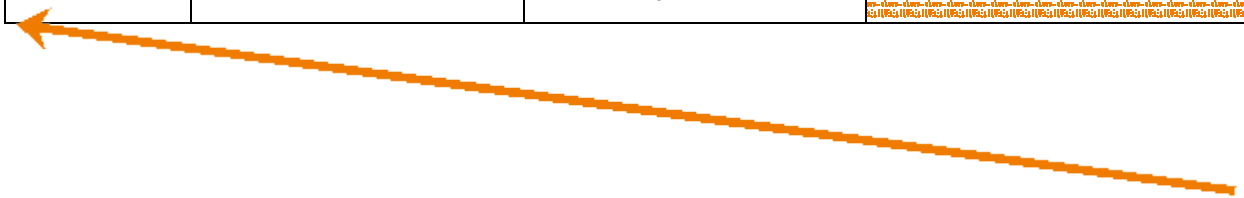
Заголовок IPv4 пакета				
Байты	**0**	**1**	**2**	**3**
0..3	Верс.+размер заголовка	Тип службы	Размер пакета	
4..7	ID группы пакетов		Опции	
8..11	**TTL**	**Номер протокола**	Контрольная сумма заголовка	
12..15	**IP-адрес отправителя**			
16..19	**IP-адрес получателя**			
20..24	Дополнительные опции			

Данные

UDP-пакеты (datagram)

Заголовок UDP				
Байты	**0**	**1**	**2**	**4**
0..4	**Порт отправителя**		Порт назначения	
5..8	Длина пакета		Контрольная сумма	

Данные



Заголовок IPv4 пакета				
Байты	**0**	**1**	**2**	**3**
0..3	Верс.+размер заголовка	Тип службы	Размер пакета	
4..7	IP		Смещение	

Номера портов

- 0 - не используется
- 20, 21 - FTP
- 22 - SSH
- 25 - SMTP
- 80 - HTTP

- **1025...65535 - исходящие в WinXP и старых UNIX**
- **32768...65535 - исходящие в Linux**
- **49152...65535 - исходящие в *BSD и WinVista+**

TCP-пакеты

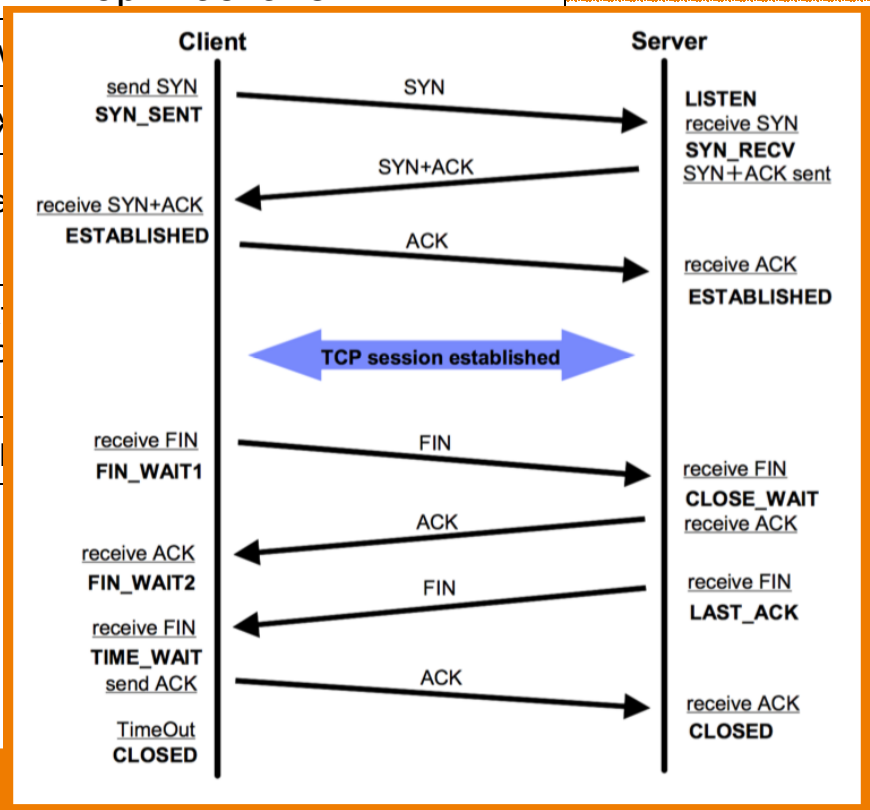
Заголовок TCP													
Байты	**0**			**1**								**2**	**4**
0..3	Порт отправления											Порт назначения	
4..7	Порядковый номер пакета												
8..11	Порядковый номер подтверждаемого пакета (ACK)												
12..15	Размер заголов-ка	000	NS	CWR	ECE	URG	**ACK**	PSH	RST	**SYN**	**FIN**	Размер окна (буфера для приема данных, ожидаемых при ответе)	
16..19	Контрольная сумма заголовка и данных											Указатель на порядковый номер пакета, в котором заканчивается блок приоритетных данных (URG)	
20...	Дополнительные опции												

Дан-ные

Заголовок IPv4 пакета				
Байты	**0**	**1**	**2**	**3**
0..3	Верс.+размер заголовка	Тип службы	Размер пакета	
4..7	IP		Смещение	

TCP-пакеты

Заголовок TCP													
Байты	**0**			**1**									**2**	**4**
0..3	Порт отправления												Порт назначения	
4..7	Порядковый ном													
8..11	Порядковый номер подтверж													
12..15	Размер заголов-ка	000		NS	CWR	ECE	URG	**ACK**	PSH	RST	**SYN**	**FIN**	Разме	
16..19	Контрольная сумма заголовка и данных												Указат которо	
20...	Дополнительны													



TCP v.s. UDP

TCP

- Полнодуплексная (двунаправленная) передача данных
- Пакеты выстраиваются в непрерывный **упорядоченный** поток данных
- Дополнительный overhead на согласование

UDP

- Отправка данных в одну сторону
- Компактные заголовки
- Без подтверждений и выстраивания данных в упорядоченный поток

Использование TCP v.s. UDP

TCP

- Полнодуплексная (двунаправленная) передача данных
- Пакеты выстраиваются в непрерывный **упорядоченный** поток данных

Клиент-серверное взаимодействие

UDP

- Отправка данных в одну сторону
- Компактные заголовки

Передача коротких сообщений (DNS)

Передача больших объемов данных, когда порядок пакетов не имеет значения

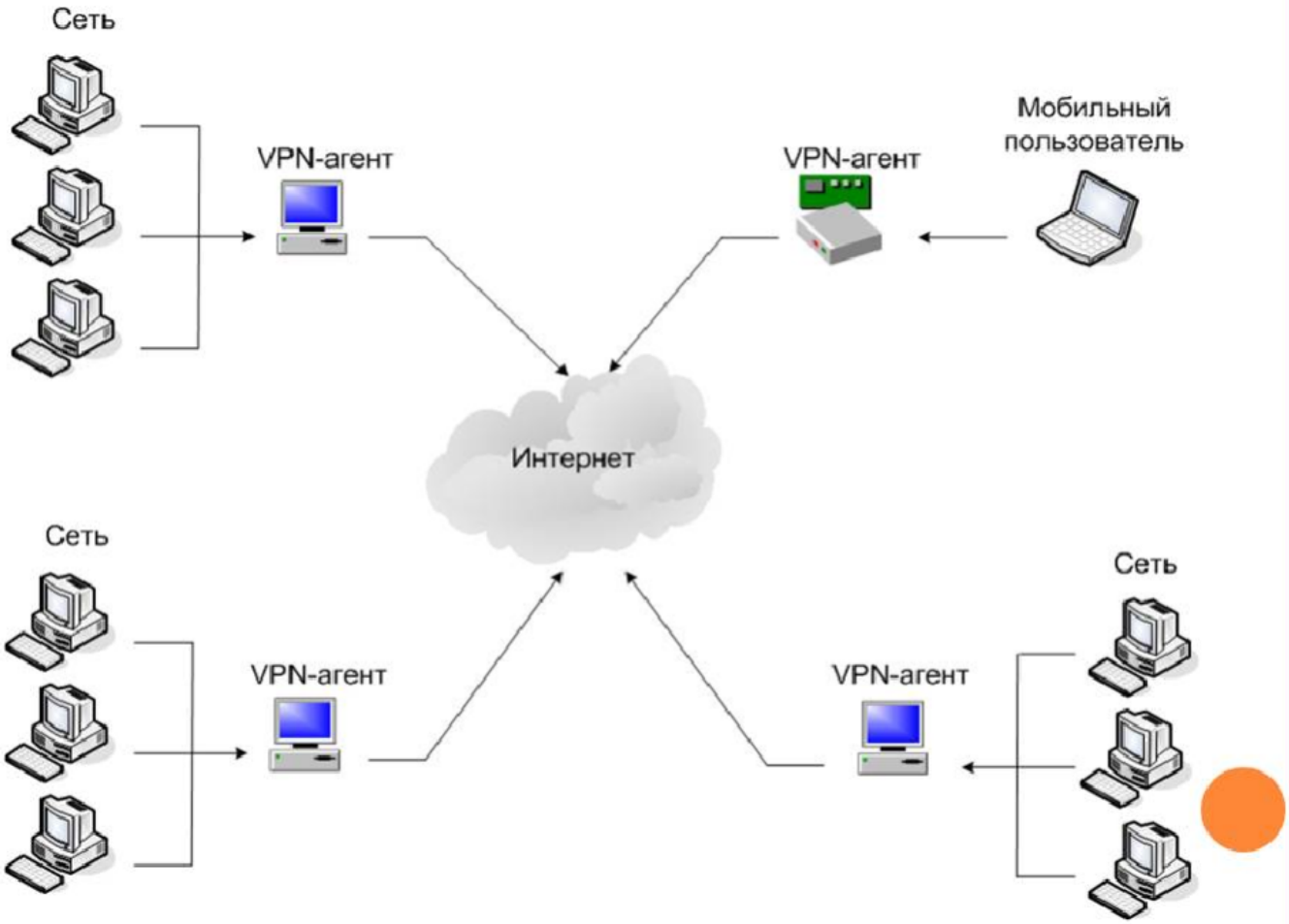
Рекурсия иерархии OSI

Уровень TCP/IP	Уровень OSI	
Уровень процессов	Прикладной (Application)	HTTP,
	Представления (Presentation)	ASC
	Сеансовый (Session)	Ne
Транспортный уровень	Транспорта (Transport)	
Уровень Internet	Сети (Network)	IPv4, IP
Уровень сетевого интерфейса	Канала (Data)	PPP, IEE
	Физический (Physical)	USB, I2C, IE 802

Туннельные интерфейсы

```
victor@victor-laptop:~> ip a l
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    link/ether 34:64:a9:c4:b9:e4 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    link/ether 10:08:b1:85:a9:55 brd ff:ff:ff:ff:ff:ff
    inet 172.19.34.12/24 brd 172.19.34.255 scope global dynamic wlan0
       valid_lft 28502sec preferred_lft 28502sec
    inet6 fe80::1208:b1ff:fe85:a955/64 scope link
       valid_lft forever preferred_lft forever
4: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc
pfifo_fast state UNKNOWN group default qlen 3
    link/ppp
    inet 192.168.1.129 peer 192.168.1.99/32 scope global ppp0
       valid_lft forever preferred_lft forever
```

VPN - назначение



VPN - технологии

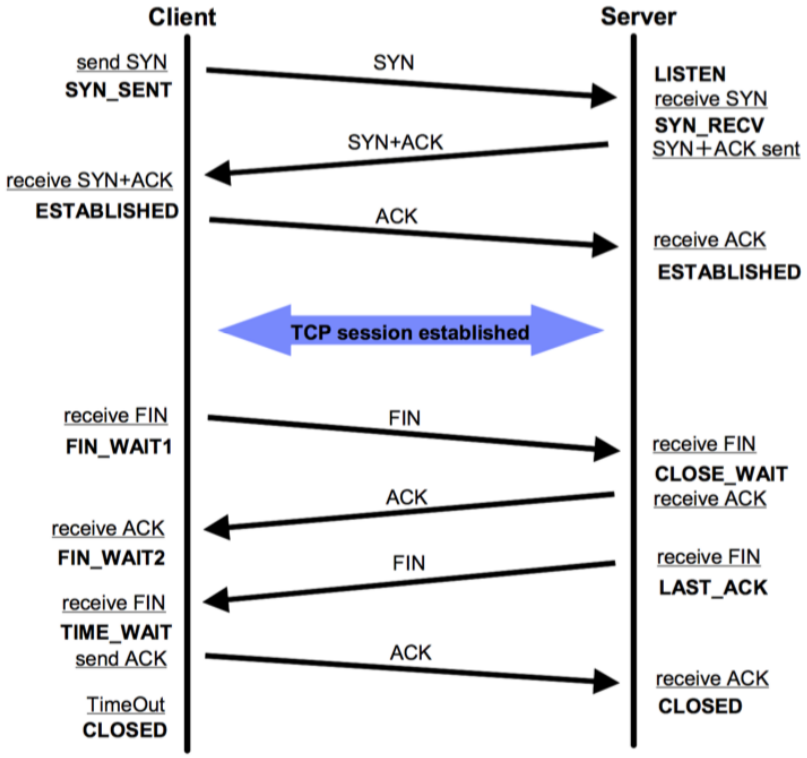
*VPN - это общее название **различных** технологий*

- Misrosoft: «Виртуальная частная сеть»
- OpenVPN - открытый протокол
- L2TP/IPSEC - один из самый старых

- Могут использовать как TCP, так и UDP
- Не совместимы между собой
- Номер порта не фиксирован

VPN: TCP v.s. UDP

TCP



UDP

Заголовок UDP				
Байты	**0**	**1**	**2**	**4**
0..4	**Порт отправителя**		Порт назначения	
5..8	Длина пакета		Контрольная сумма	

Маршрутизация траффика

```
victor@victor-laptop:~> /sbin/route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         *               0.0.0.0         U     50     0        0 ppp0
default         172.19.34.1     0.0.0.0         UG    600    0        0 wlan0
1dot1dot1dot1.c 172.19.34.1     255.255.255.255 UGH   600    0        0 wlan0
172.19.34.0     *               255.255.255.0   U     600    0        0 wlan0
192.168.1.99    *               255.255.255.255 UH    50     0        0 ppp0
t01.niisi.ras.r 172.19.34.1     255.255.255.255 UGH   600    0        0 wlan0
```

Использовать VPN для всего траффика - не всегда рационально!

Проксирование

Задачи прокси

Reverse HTTP Proxy

- Снизить нагрузку на серверы приложений
- Улучшить доступность сервисов в ралзичных регионах

HTTP Proxy

- Снизить нагрузку на канал передачи данных

Протокол HTTP

Запрос

```
GET /index.html HTTP/1.1
Host: www.example.com
Connection: keep-alive
DNT: 1
User-Agent: Mozilla/5.0 . . .
Accept-Encoding: gzip, deflate
Accept-Language: ru, en
```

Протокол HTTP

Базовые возможности

- GET – получить данные
- HEAD – получить только заголовки
- POST – отправить данные

Дополнительные команды для REST и WebDAV

- PUT – создать файл
- MKCOL – создать каталог
- PROPFIND – получить свойства файла
- COPY – скопировать файл
- MOVE – переместить файл

Протокол HTTP

Запрос

```
GET /index.html HTTP/1.1
Host: www.example.com
Connection: keep-alive
DNT: 1
User-Agent: Mozilla/5.0 . . .
Accept-Encoding: gzip, deflate
Accept-Language: ru, en
```

Ответ

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html
Transfer-Encoding: chunked
Date: Mon, 27 Apr 2015 13:40:00

<html>
  <head></head>
  <body><p>It works!</p></body>
</html>
```

Протокол HTTP

2XX – Успешно

200 – OK для HEAD/GET/POST
201 – файл успешно создан (для WebDAV PUT)
207 – Multi-Status (для WebDAV PROPFIND)

3XX – Переадресация

301 – Содержимое перемещено на новый адрес
304 – Содержимое не изменилось с прошлого раза

4XX – Ошибка по вине клиента

403 – Обрашение к ресурсу, на который нет прав доступа
404 – Неверный адрес
451 – 451 градус по Фаренгейту (232 градуса по Цельсию)

5XX – Ошибка по виде сервера

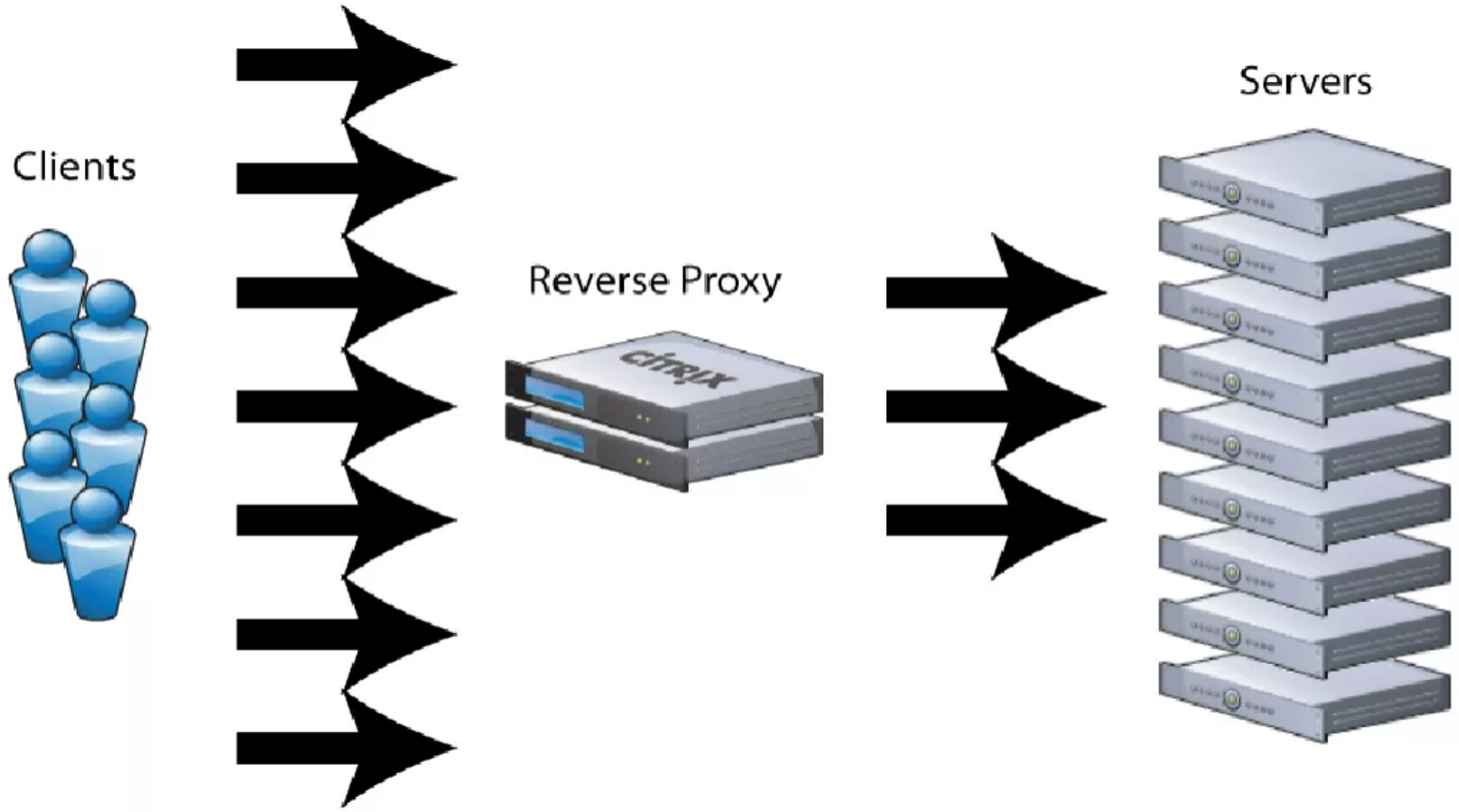
500 – Внутренняя ошибка
501 – Функциональность не реализована
502 – У http-reverse-proxy нет соединения с сервером
503 – “Временно не работает. Приносим свои извинения”
507 – Закончилось место (WebDAV)

HTTP-проски

- На стороне клиентов:
  - squid (порт 3129) - кеширующий HTTP, предназначен для экономии траффика
- На стороне серверов:
  - nginx (порт 80) - кеширует статику, принимает огромное количество подключений; предназначен для снижения нагрузки на серверы приложений

Подразумевается извлечение данных и/или модификация заголовков, поэтому только HTTP

Прокси: HTTP - nginx



Прокси: HTTP - nginx

Запрос

```
GET /index.html HTTP/1.1
Host: www.example.com
Connection: keep-alive
DNT: 1
User-Agent: Mozilla/5.0 . . .
Accept-Encoding: gzip, deflate
Accept-Language: ru, en
X-Forwarded-For: 12.34.56.78,
23.45.67.89
X-Real-IP: 12.34.56.78
X-Forwarded-Host: example.com
X-Forwarded-Proto: https
```

Ответ

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html
Transfer-Encoding: chunked
Date: Mon, 27 Apr 2015 13:40:00
X-Forwarded-For: 12.34.56.78,
23.45.67.89
X-Real-IP: 12.34.56.78

<html>
  <head></head>
  <body><p>It works!</p></body>
</html>
```

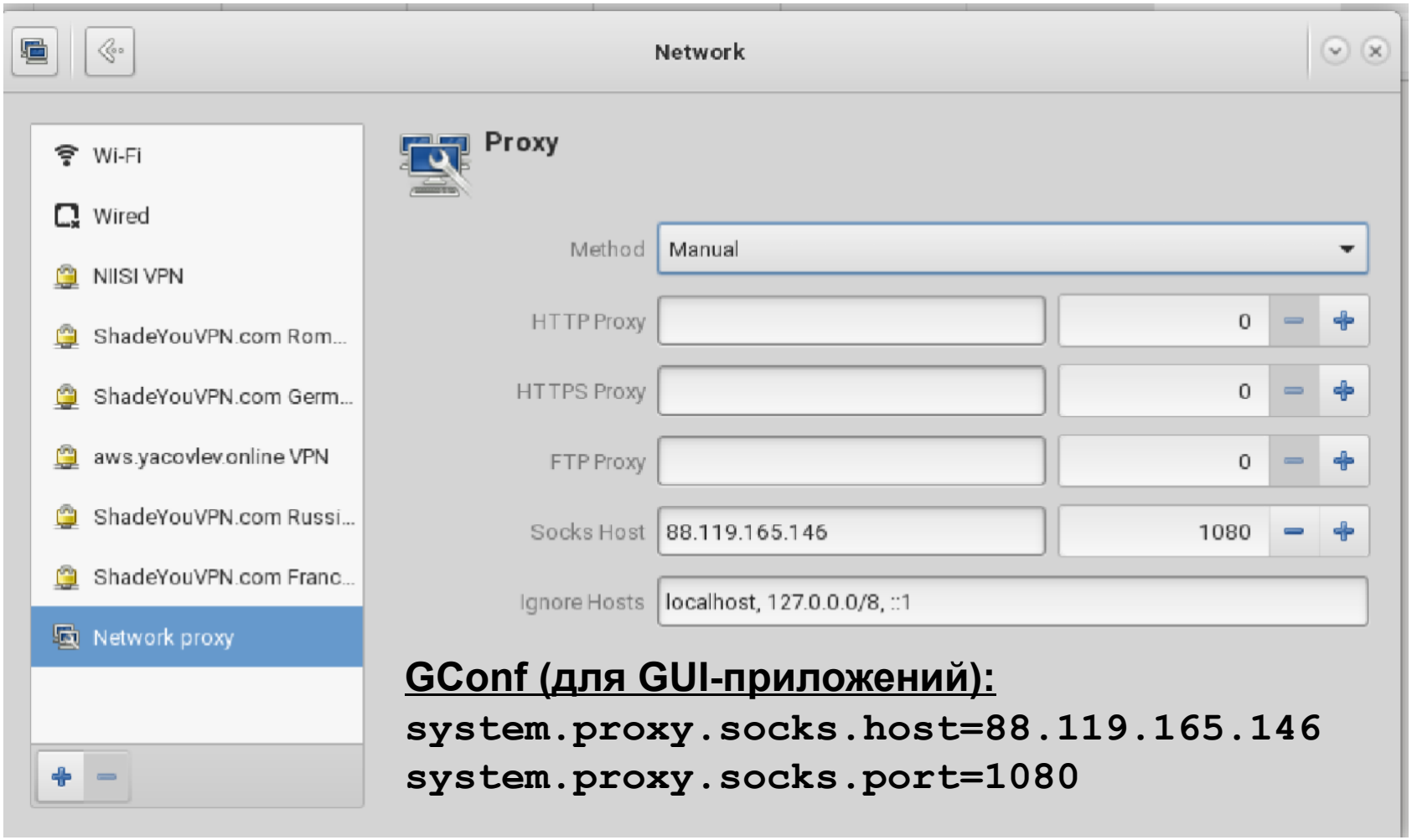
SOCKS4 / SOCKS5

- Бинарный протокол для выхода во внешнюю сеть из изолированной локальной сети
- Реализация под Linux: Dante [порт 1080]
- Является простой альтернативой файрволу
- Версия SOCKS5 поддерживает авторизацию, что можно использовать для учёта потребляемого траффика

SOCKS4 / SOCKS5

- В отличии от VPN не создает отдельный сетевой (туннельный) интерфейс
- Возможность работы через SOCKS - на уровне приложений
- Каждому отдельному приложению можно указать свой SOCKS-прокси
- Передача данных, в отличии от HTTP, осуществляется **As Is**

SOCKS4 / SOCKS5



GConf (для GUI-приложений):

```
system.proxy.socks.host=88.119.165.146
system.proxy.socks.port=1080
```

Немного про безопасность

- Протоколы TCP (транспортный уровень) / IP (уровень сети) не регламентируют шифрование
- HTTP - передается как plain text
- HTTPS подразумевает шифрование
- VPN может, но не обязан шифровать данные
- SOCKS передает данные без изменений

Спасибо за внимание!