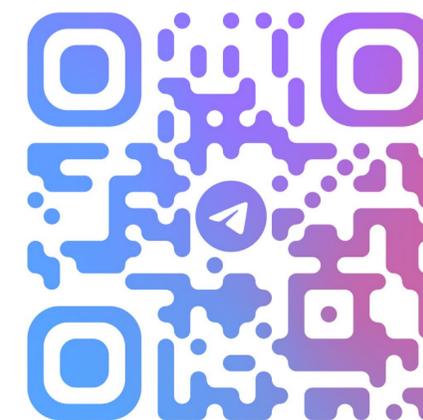


Сетевая Безопасность



Секреты производительности NGFW

Денис Батранков
Директор по продуктам
информационной
безопасности
ИКС Холдинг



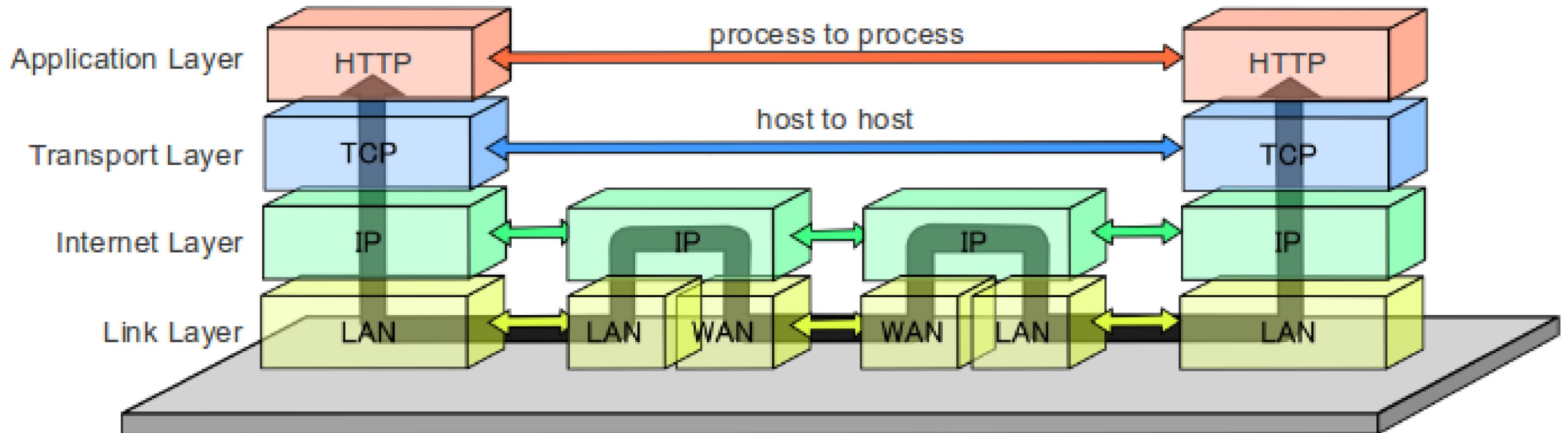
@SAFEBDV



Каждое сетевое устройство обрабатывает трафик на своем уровне абстракции

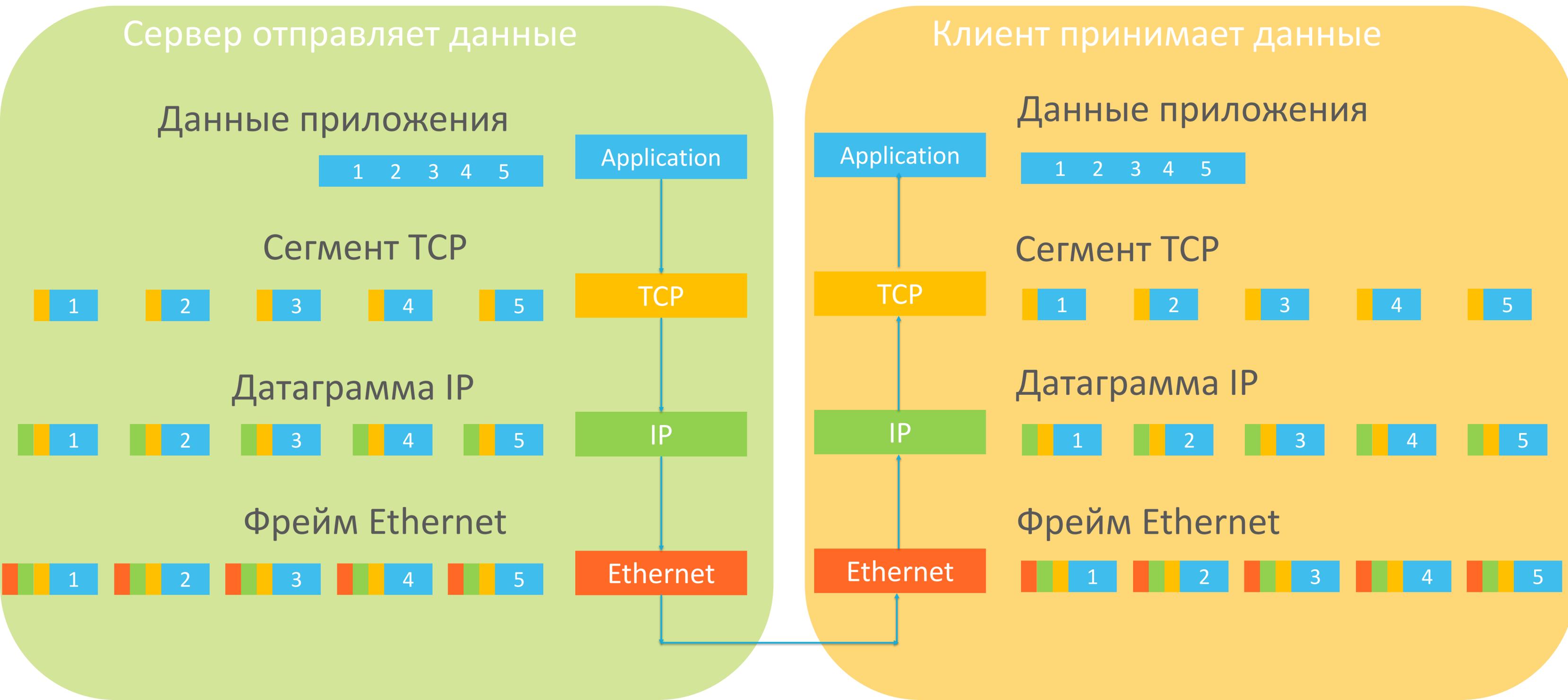


Data Flow of the Internet Protocol Suite



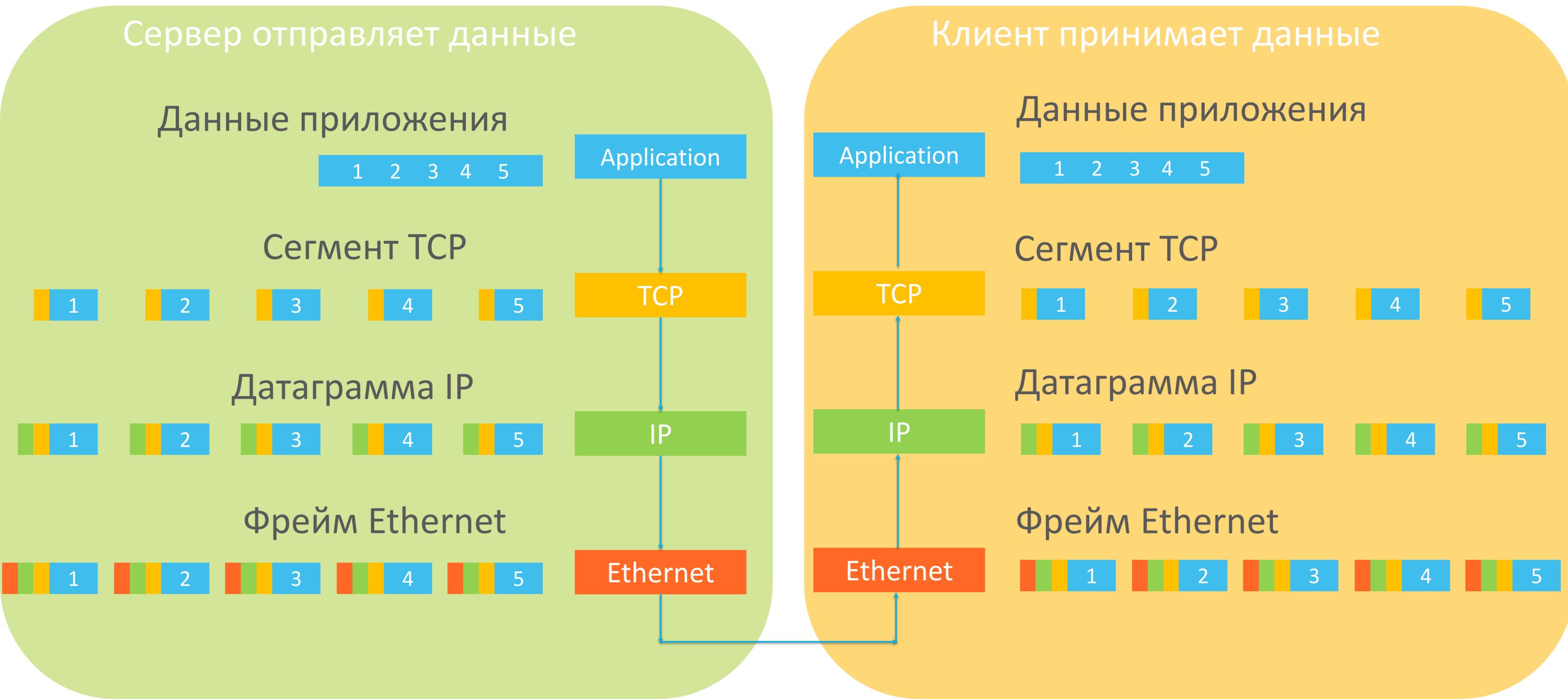
Фрагментация данных влияет на скорость

TCP делит данные приложения на сегменты и добавляет заголовок минимум 20 байт.
MSS – макс. длина сегмента данных TCP – зависит от мин. длины фрейма



Фрагментация нужна для прохождения через физическую среду передачи

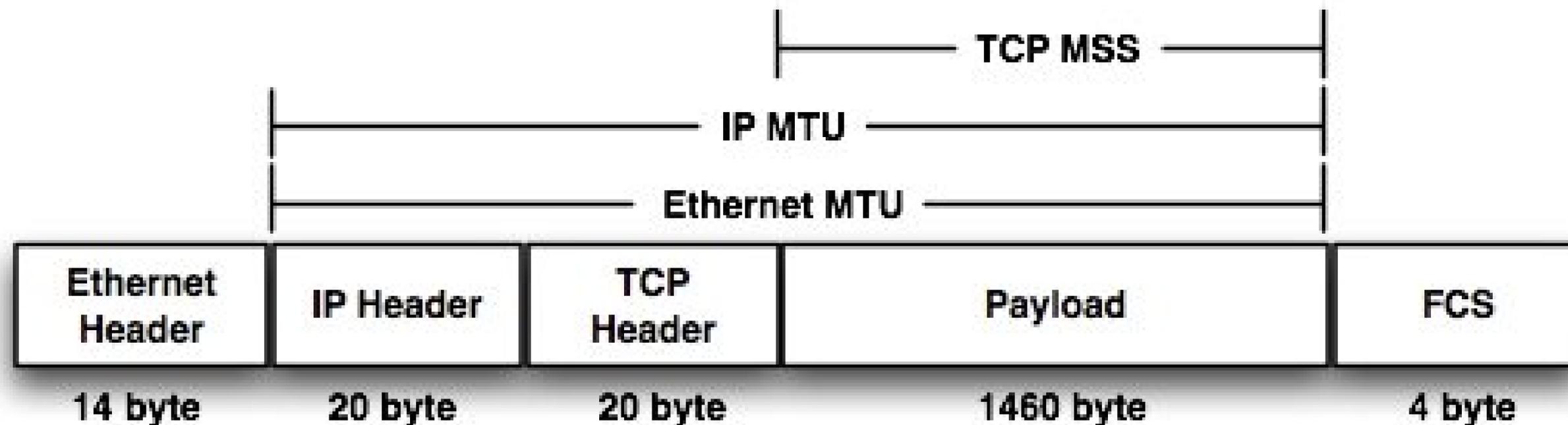
MTU – максимальная длина фрейма. 1500 у фреймов Ethernet, 9000 – это jumbo фрейм
MSS – это длина фрейма минус длина заголовков. $MSS = MTU - 40 = 1460$



Формат одного фрейма Ethernet

MTU – максимальный размер фрейма для данной среды передачи.

MSS – максимальный размер сегмента данных TCP. $MSS = MTU - 40 = 1460$



Правила работы стека TCP/IP:

- *Если IP пакет больше чем MTU то он фрагментируется в несколько фреймов.*
- *Если TCP пакет больше чем MSS, то он удаляется.*

Любая фрагментация накладывает доп. нагрузку, поэтому быстрее всего отправлять и принимать пакеты UDP 1500 байт – они не требуют фрагментации.

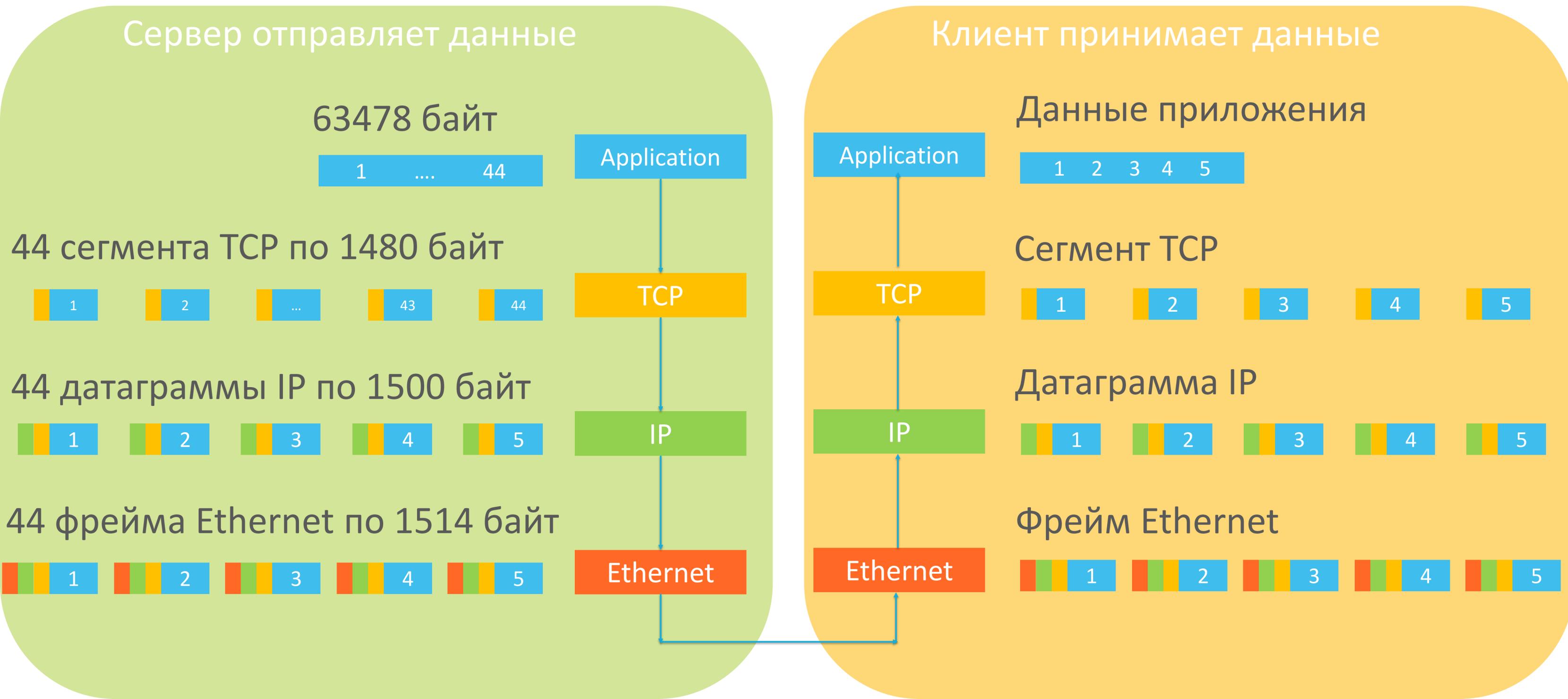
Основы протокола TCP/IP

Идеальный размер транзакций для передачи любым устройством: 1460 байт, потому что тогда он помещается и в IP пакет и в фрейм и не нужно тратить ресурсы на фрагментирование и сборку.

- ▶ Frame 9550: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
- ▶ Ethernet II, Src: ZyxelCom_cb:c6:7c (5c:f4:ab:cb:c6:7c), Dst: Apple_4d:54:b7 (8c:85:90:4d:54:b7)
- ▶ Internet Protocol Version 4, Src: 78.41.199.241, Dst: 192.168.1.47
- ▶ Transmission Control Protocol, Src Port: 443, Dst Port: 61052, Seq: 968715, Ack: 52392, Len: 1460
- ▼ [44 Reassembled TCP Segments (63478 bytes): #9471(1460), #9473(1460), #9476(1460), #9477(1460), #9478(1460),
 - [\[Frame: 9471, payload: 0-1459 \(1460 bytes\)\]](#)
 - [\[Frame: 9473, payload: 1460-2919 \(1460 bytes\)\]](#)
 - [\[Frame: 9476, payload: 2920-4379 \(1460 bytes\)\]](#)
 - [\[Frame: 9477, payload: 4380-5839 \(1460 bytes\)\]](#)
 - [\[Frame: 9478, payload: 5840-7299 \(1460 bytes\)\]](#)
 - [\[Frame: 9481, payload: 7300-8759 \(1460 bytes\)\]](#)
 - [\[Frame: 9482, payload: 8760-10219 \(1460 bytes\)\]](#)
 - [\[Frame: 9483, payload: 10220-11679 \(1460 bytes\)\]](#)
 - [\[Frame: 9484, payload: 11680-13139 \(1460 bytes\)\]](#)
 - [\[Frame: 9488, payload: 13140-14599 \(1460 bytes\)\]](#)
 - [\[Frame: 9489, payload: 14600-16059 \(1460 bytes\)\]](#)
 - [\[Frame: 9490, payload: 16060-17519 \(1460 bytes\)\]](#)
 - [\[Frame: 9492, payload: 17520-18979 \(1460 bytes\)\]](#)
 - [\[Frame: 9493, payload: 18980-20439 \(1460 bytes\)\]](#)
 - [\[Frame: 9496, payload: 20440-21899 \(1460 bytes\)\]](#)
 - [\[Frame: 9497, payload: 21900-23359 \(1460 bytes\)\]](#)
 - [\[Frame: 9501, payload: 23360-24819 \(1460 bytes\)\]](#)

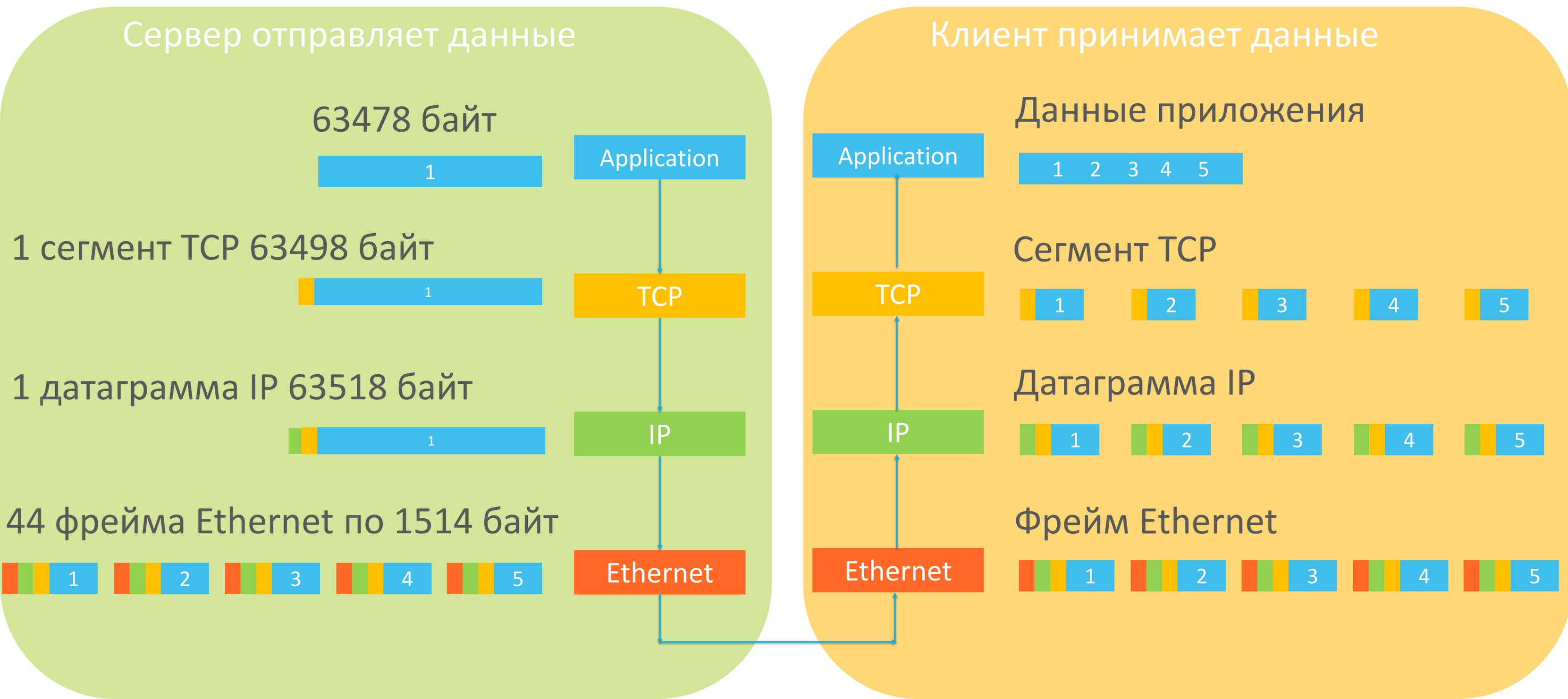
Все начинается с основ TCP/IP

Например, 63478 байт TCP умещается в 44 фреймах, поскольку данных умещается только 1460 байт



Все начинается с основ TCP/IP

Например, 63478 байт TCP умещается в 44 фреймах, поскольку данных умещается только 1460 байт. Максимальная длина IP датаграммы 65535



А что будет, если размер одной транзакции на уровне приложений 1 мегабайт?

Каждому сетевому устройству нужно собрать данные приложения из множества фреймов, ведь MTU обычно 1500 байт

```
36 0.20 174.143.213.184 192.168.1.2 HTTP 1296 HTTP/1.1 200 OK (PNG)
```

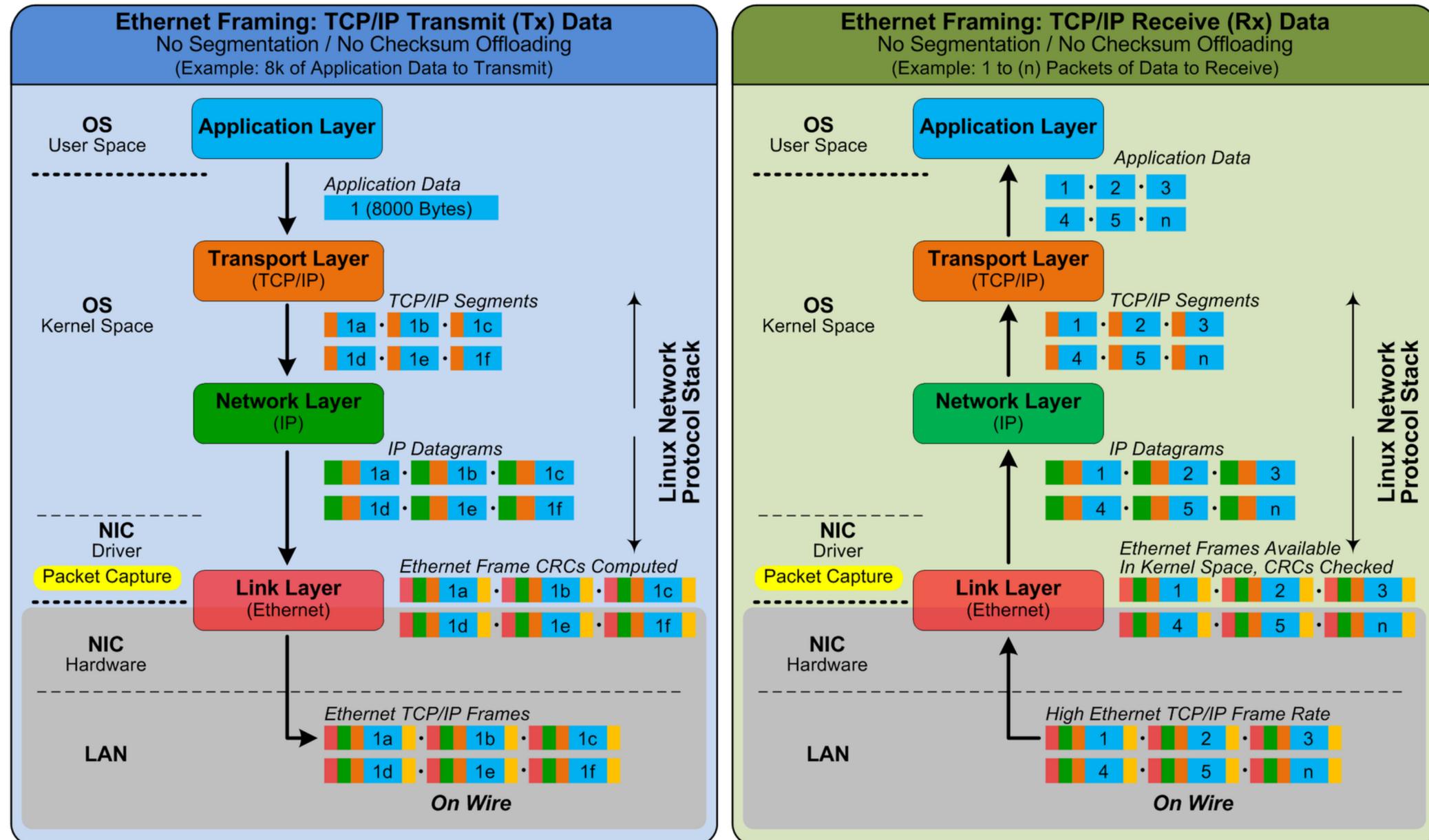
Картинка PNG размером 22Кб уместилась в 16 сегментах TCP

```
⊞ Frame 36: 1296 bytes on wire (10368 bits), 1296 bytes captured (10368 bits) on interface 0
⊞ Ethernet II, Src: 00:26:62:2f:47:87 (00:26:62:2f:47:87), Dst: 00:1d:60:b3:01:84 (00:1d:60:b3:01:84)
⊞ Internet Protocol Version 4, Src: 174.143.213.184 (174.143.213.184), Dst: 192.168.1.2 (192.168.1.2)
⊞ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 54841 (54841), Seq: 21721, Ack: 726, Len: 1230
⊞ [16 Reassembled TCP Segments (22950 bytes): #6(1448), #8(1448), #10(1448), #12(1448), #14(1448), #16(1448)]
⊞ Hypertext Transfer Protocol
⊞ Portable Network Graphics
```

Напомним как работает передача файла

Ethernet Framing without Segmentation or Checksum Offloading

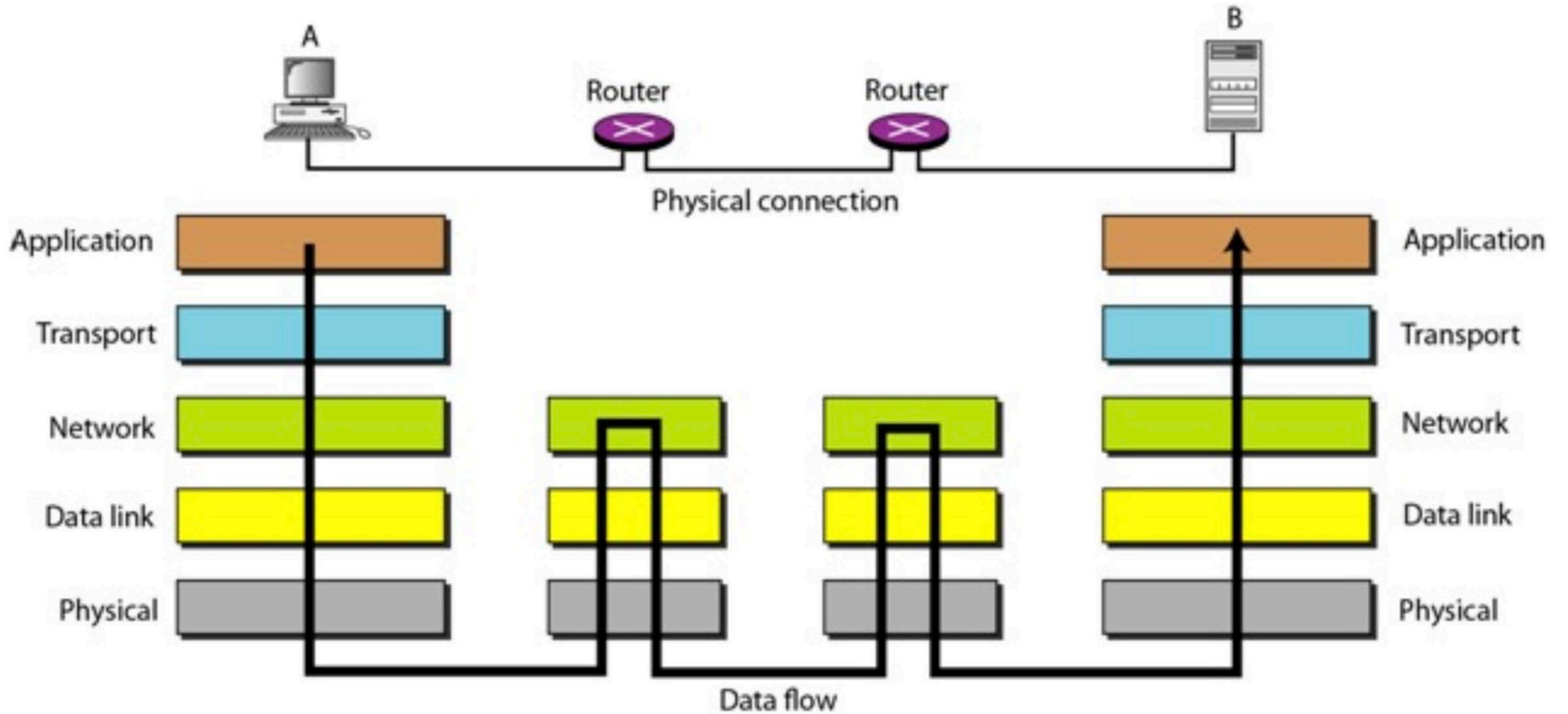
Linux Ethernet Tx / Rx Framing No Segmentation or Checksum Offloading for TCP/IP.



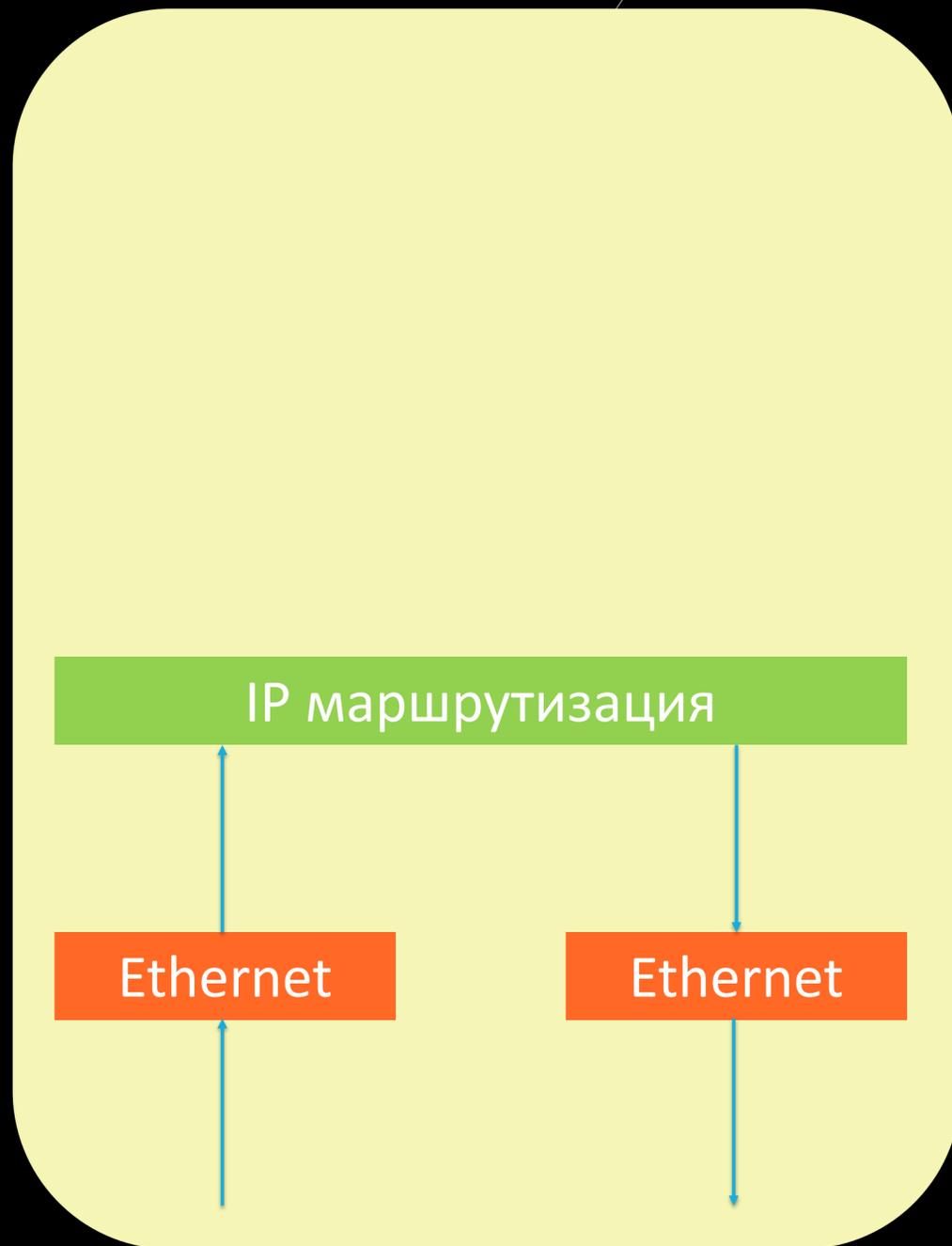
NST - 2011

<https://www.youtube.com/watch?v=zhlMLRNY5-4>

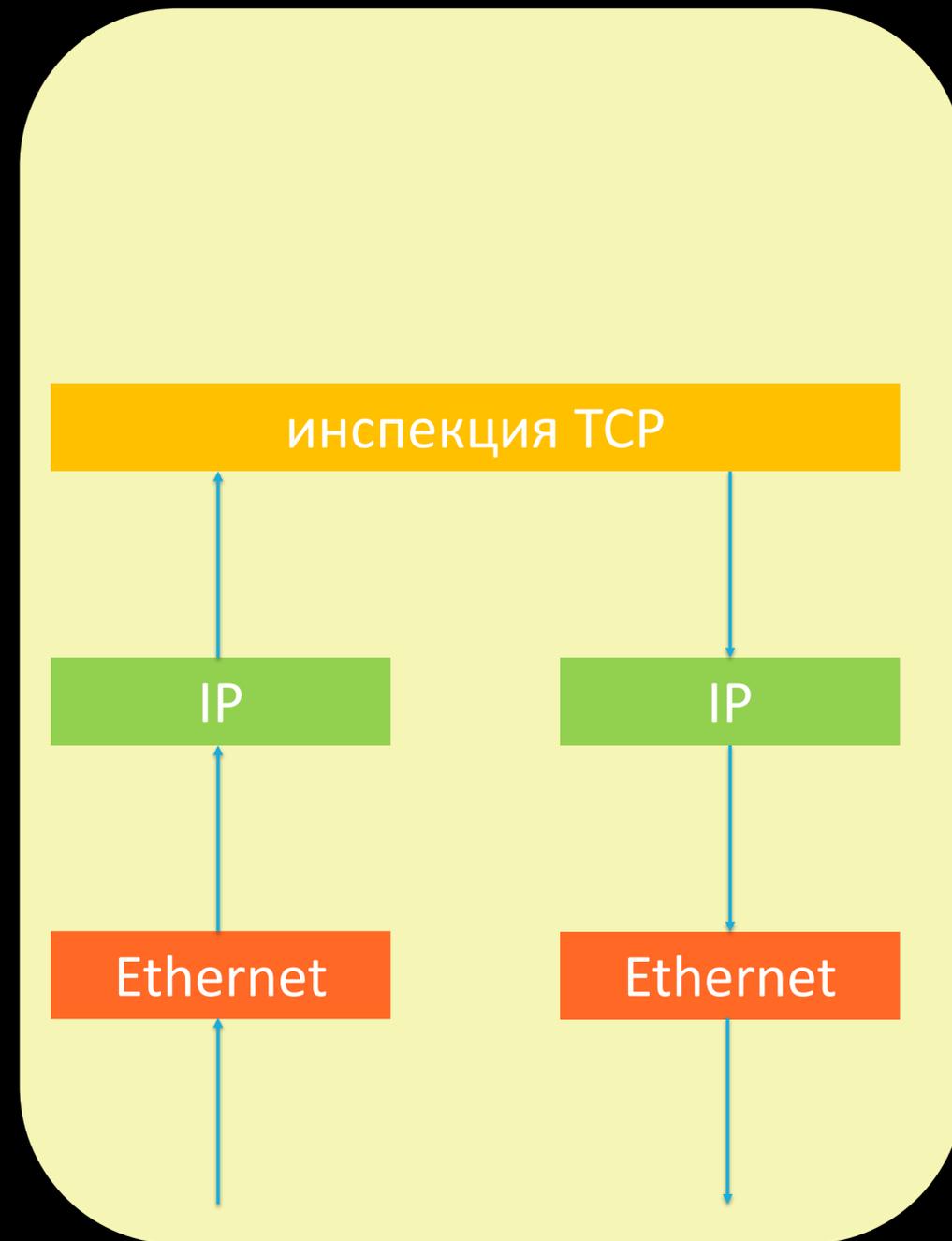
Ваше пограничное устройство инспектирует трафик



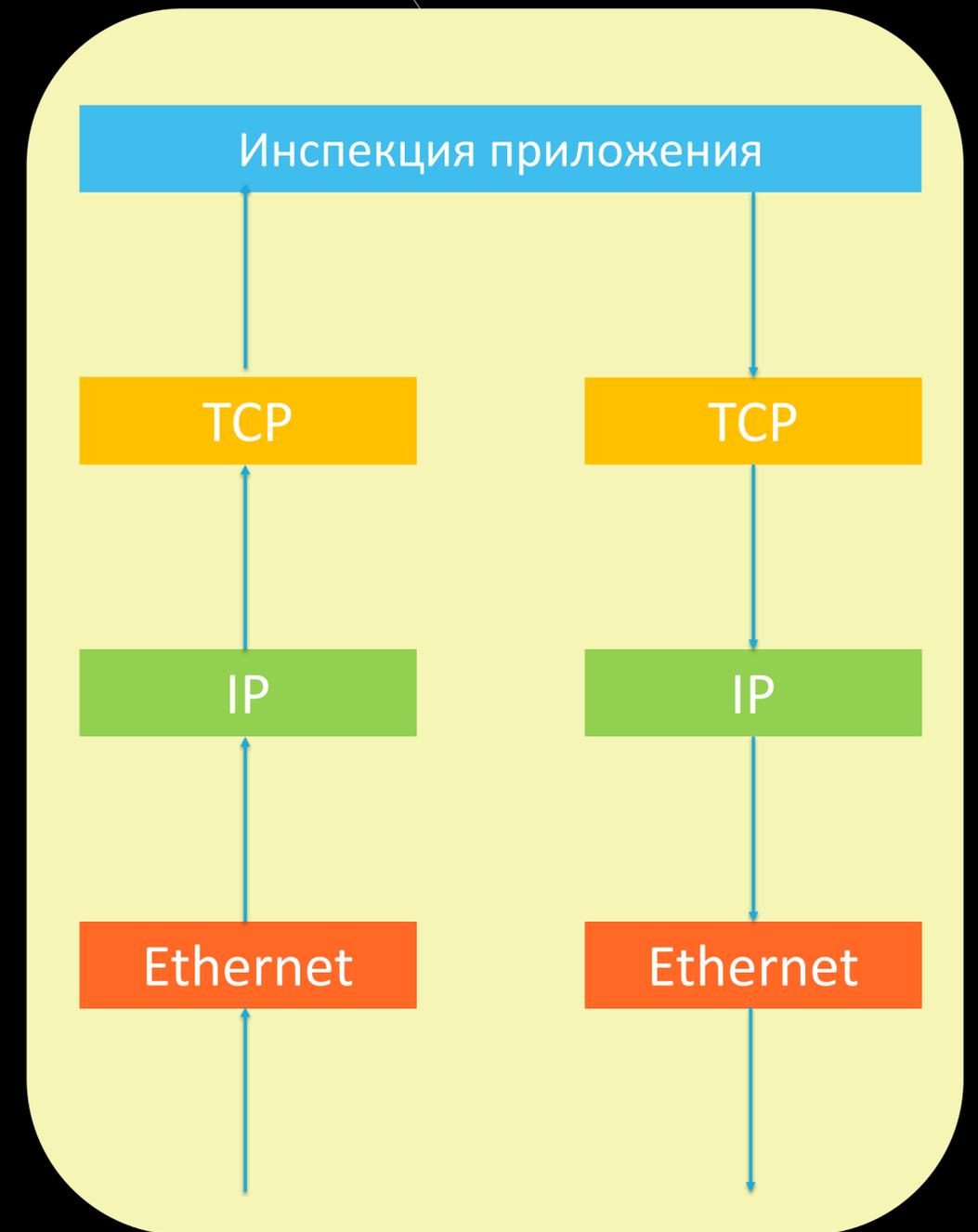
Чем больше функционала – тем медленнее обрабатывается



роутер



L4 firewall

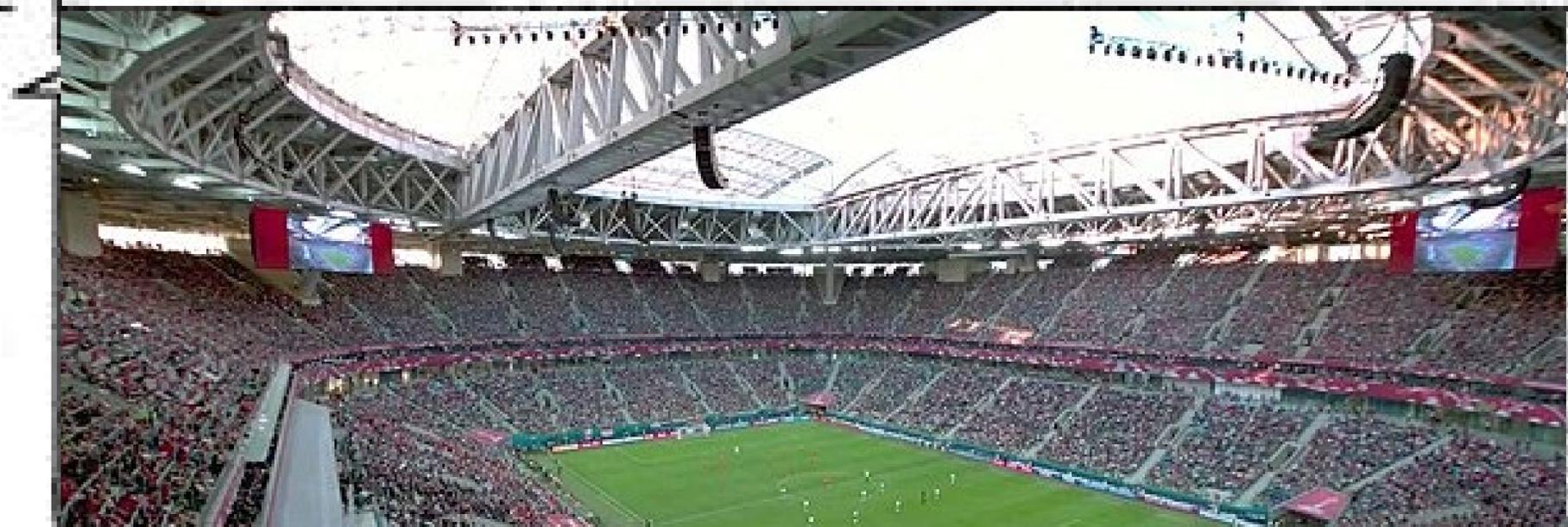


NGFW/UTM

ГАЗПРОМ АРЕНА - это 64 тысячи мест. Просто ли проверить 64000 человек?



Из TCP пакета длиной 64000 байт, L4 firewall смотрит только 40 байт



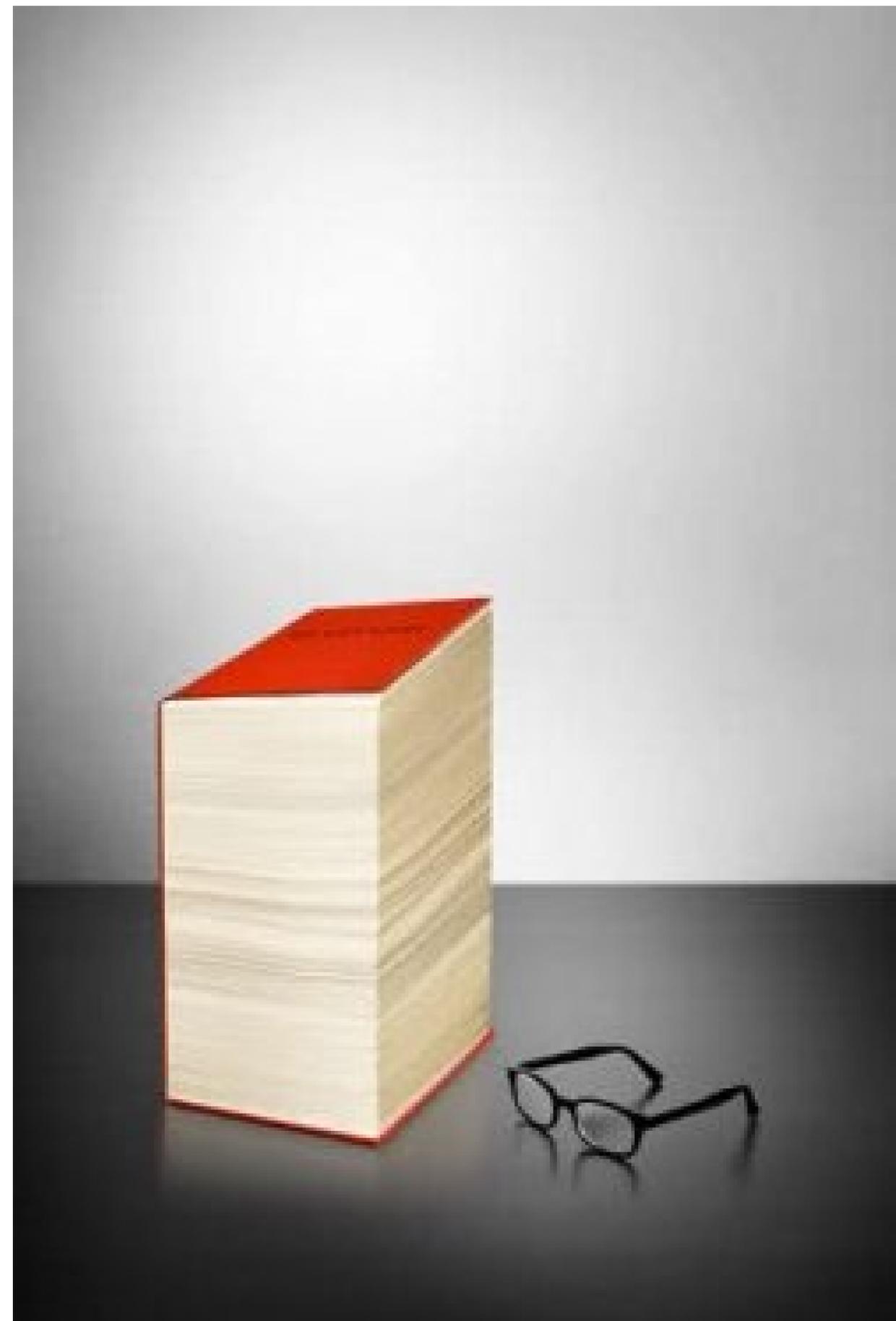
← надо проверить!

Из TCP пакета 64Кбайт: L4 firewall проверяет 40 байт, L7 firewall - все 64000 байт



Закон 1: чем проще работа, тем быстрее ее выполнить!

Что быстрее прочитать:
заголовок книги или всю книгу?



L4 firewall и L7 firewall абсолютно разные устройства по работе с трафиком!

Для одного пакета в 64 Кбайт объем работы отличается в 1600 раз

Отличие в анализе приложений

HTTPS

URL фильтрация
IPS
антивирус
песочница
DLP

Расшифрование
SSL

Шифрование
SSL

TCP

TCP

IP

IP

Ethernet

Ethernet

FTPS

IPS
антивирус
песочница
DLP

Расшифрование
SSL

Шифрование
SSL

TCP

TCP

IP

IP

Ethernet

Ethernet

SMB

IPS
антивирус
песочница
DLP

TCP

TCP

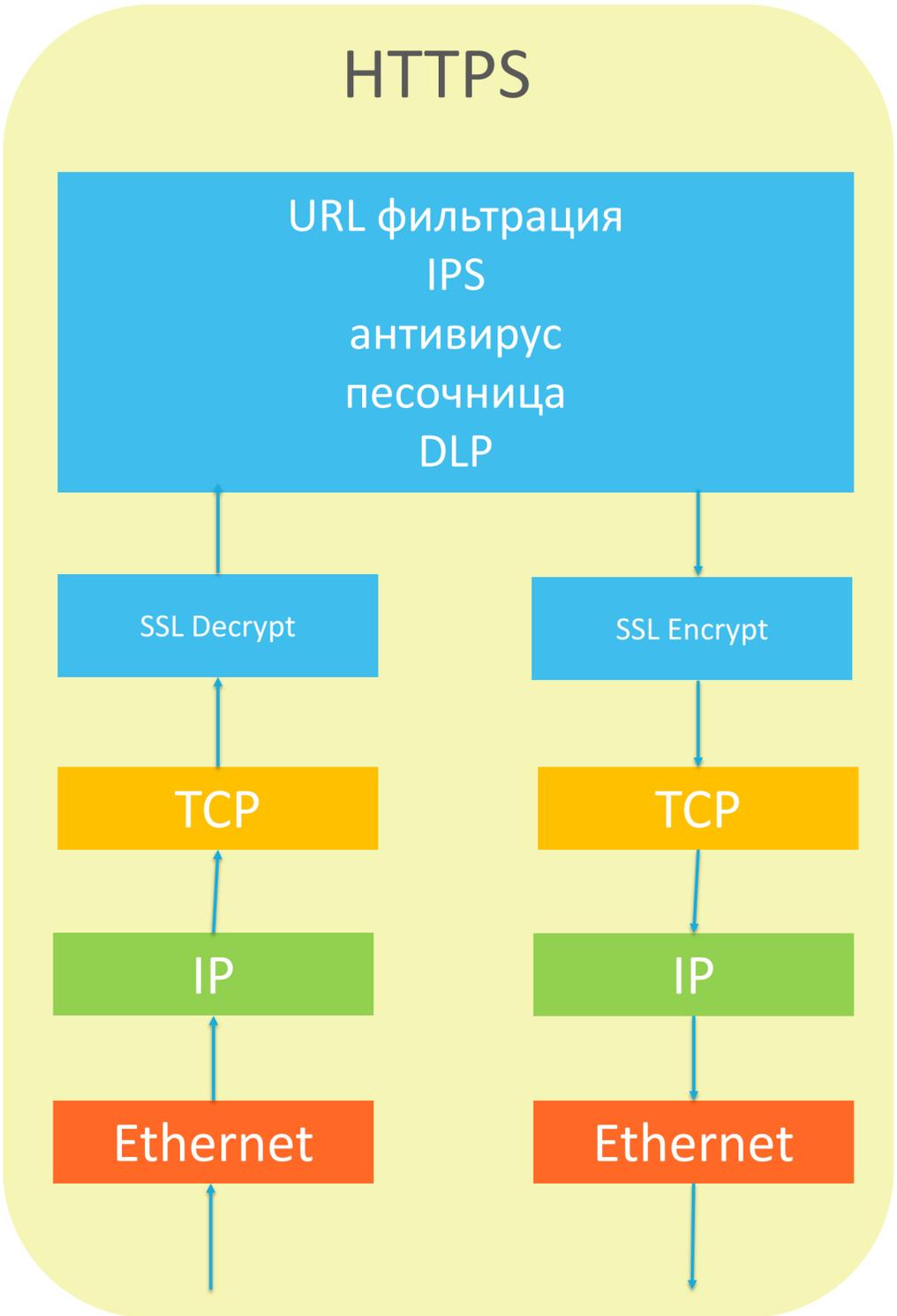
IP

IP

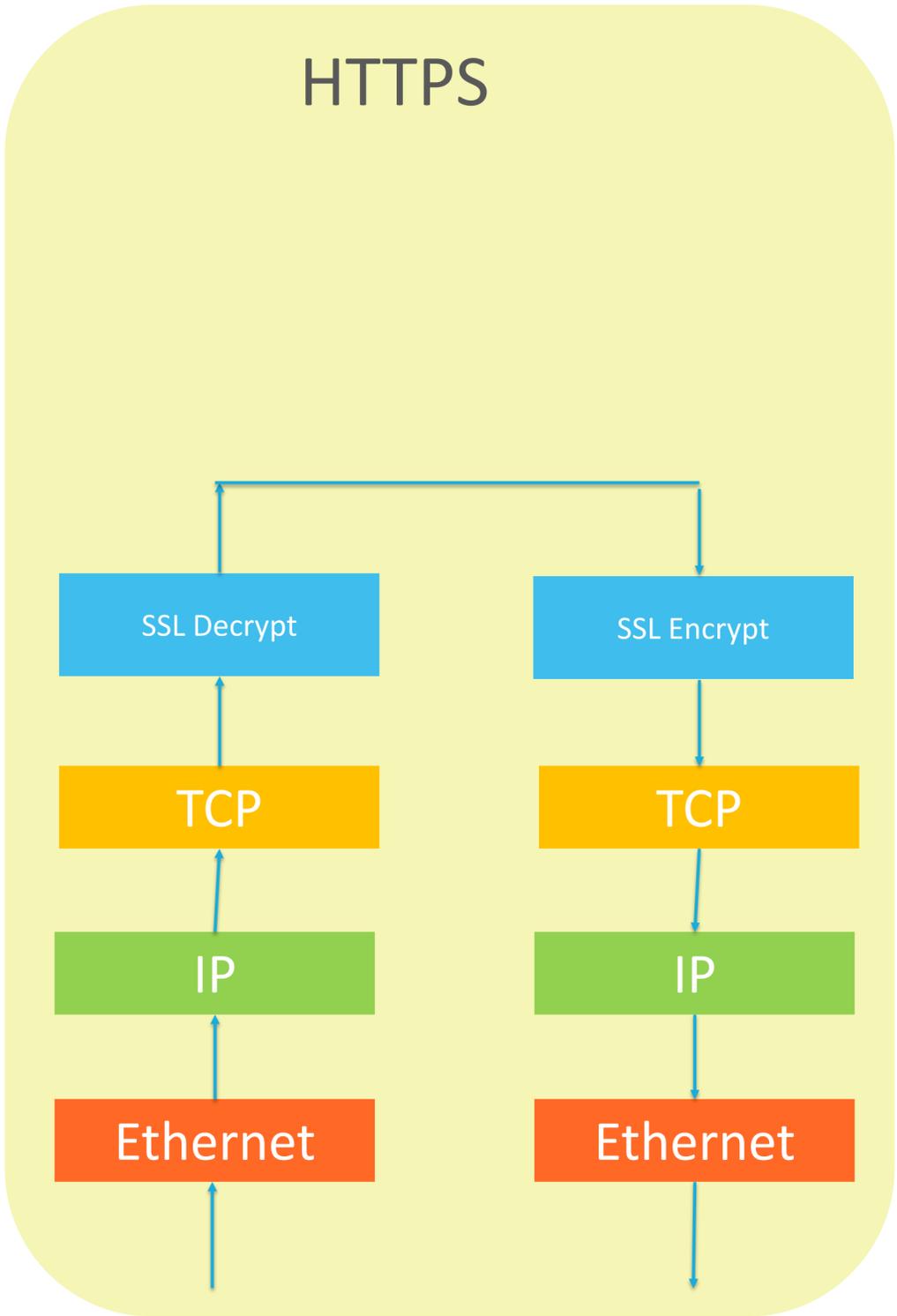
Ethernet

Ethernet

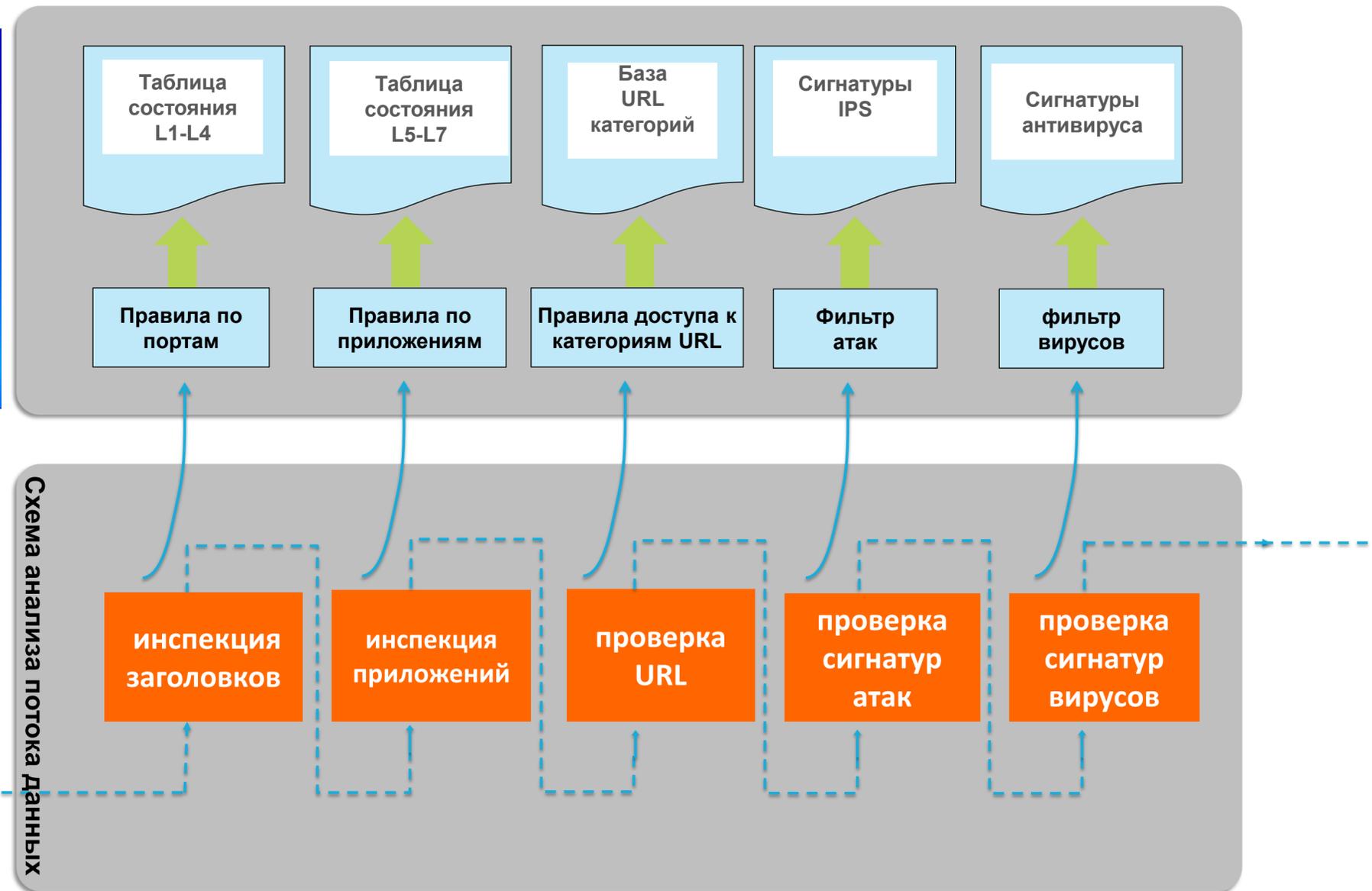
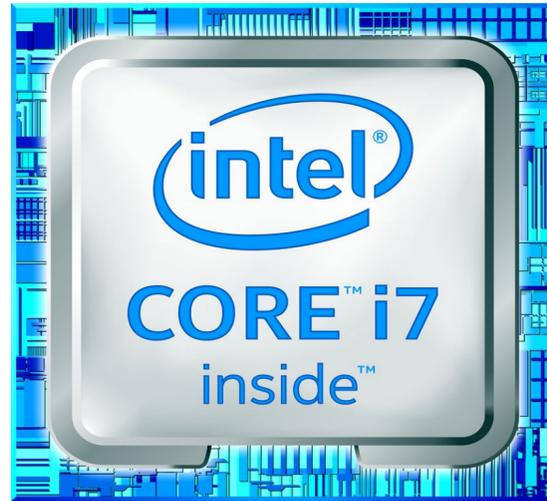
Так работает NGFW в боевом режиме



Так тестирует



Разные стадии проверки - последовательно



**Закон 2: чем больше проверок,
тем медленнее!**

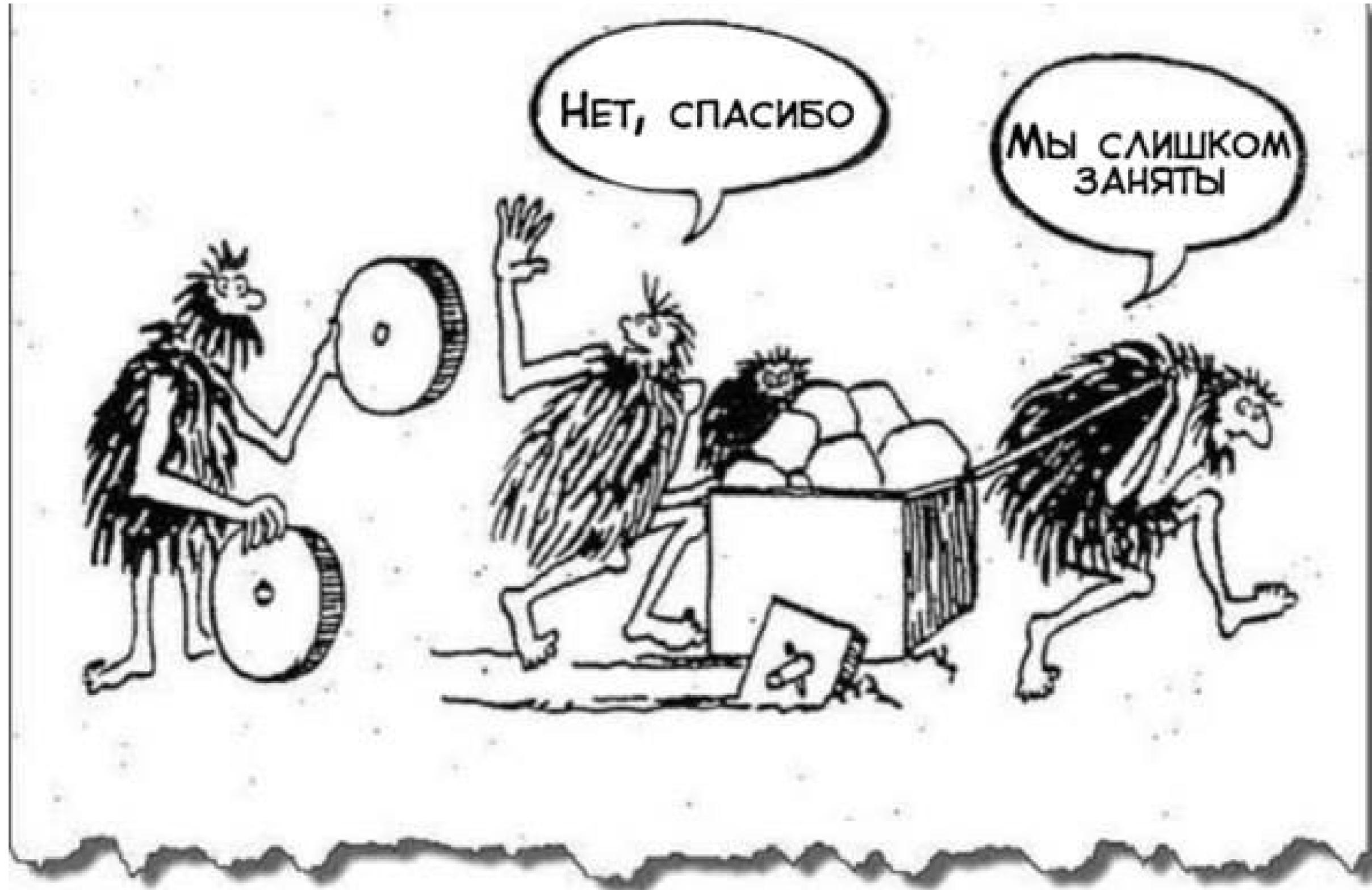


**Закон 2: чем больше проверок,
тем медленнее!**



Правила (банальные):
чтобы быстрее работало,
нужно больше аппаратных ресурсов

Без тестирования не проверишь!



Зависит ли скорость роутера от типа данных?

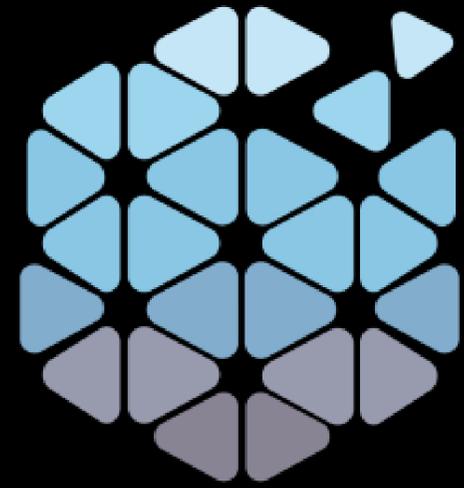
Роутеру все равно какие данные он передает –
он анализирует только заголовки TCP/IP протокола

Зависит ли скорость NGFW от типа данных?

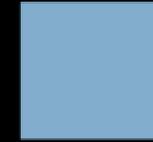
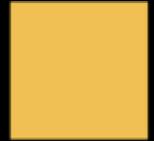
NGFW должен анализировать все данные
и также ему надо анализировать заголовки TCP/IP протокола

От разных типов атак нужна разная защита

Этапы атаки	Рекогносцировка	Доставка	Эксплуатация уязвимости	Загрузка ПО для «черного хода»	Установление обратного канала	Разведка и кража данных
DDoS защита	Блокировка сканирований, аномалий в пакетах, защита NGFW					
Контроль приложений Контроль пользователей		Блокировка нежелательных приложений, контроль пользователей			Блокировка коммуникаций на нестандартных портах	
Продвинутая URL фильтрация		Блокировка известных и неизвестных вредоносных сайтов			Блокировка вредоносных ресурсов, fast-flux	
Threat Intelligence		Блокировка известных вредоносных ресурсов			Блокировка вредоносных ресурсов, fast-flux	
Безопасность DNS					Блокировка неизвестных вредоносных ресурсов (DGA), DNS-туннели и т.д.	
IPS	Threat Prevention	Блокировка сканирований, аномалий в пакетах	Блокировка эксплоитов			Скоординированная интеллектуальная блокировка активных атак по сигнатурам, источникам, поведению
Anti-Spyware					Блокировка Spyware, C&C	
Антивирус					Блокировка вредоносного ПО	
Блокировка типов файлов				Блокировка drive-by downloads		
Проверка неизвестных файлов в песочнице				Блокировка неизвестного вредоносного ПО	Блокировка новых C&C	
Machine learning на борту NGFW		Блокировка неизвестных вредоносных ресурсов		Блокировка неизвестного вредоносного ПО		



Сетевая Безопасность



Будьте бдительны!



@SAFEBDV

