

C3

Rețele de calculatoare

Sergiu Nisioi
sergiu.nisioi@unibuc.ro

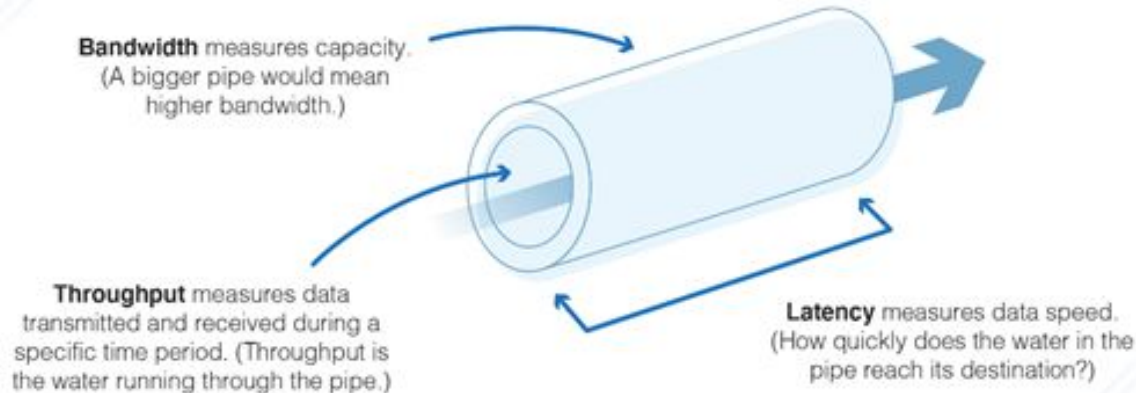
Anul II, FMI, UniBuc, 2021-2022

De data trecută

explicate mai bine si aici:

<https://blog.scaleway.com/understanding-network-latency/amp>

Network Latency vs. Throughput vs. Bandwidth



Latența, Debitul și Lățimea de bandă

De data trecută

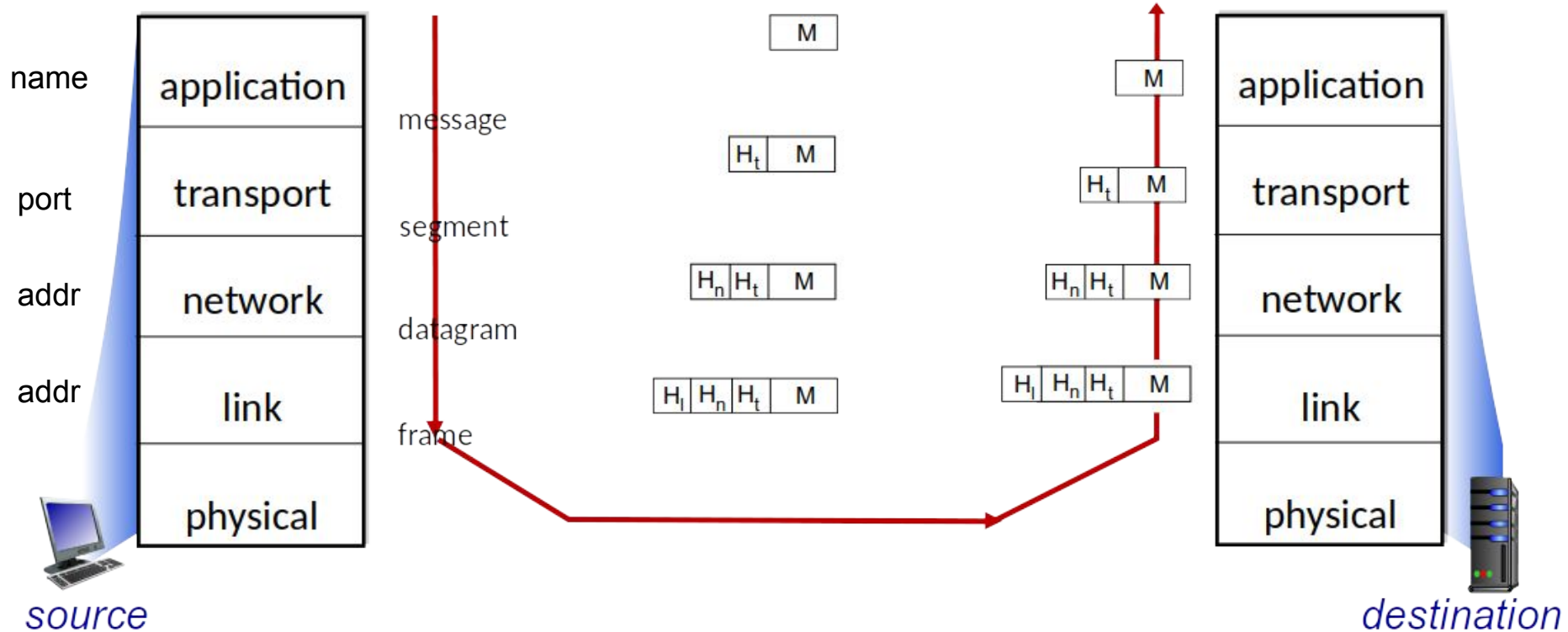
- ne conectăm pe AWS, facem o nouă instanță de EC2 cu ubuntu pe care instalăm traceroute

```
traceroute fmi.unibuc.ro
```

```
traceroute to fmi.unibuc.ro (193.226.51.15), 64 hops max
```

1	10.11.0.1	4,287ms	11,155ms	2,113ms
2	172.20.241.1	2,681ms	3,358ms	2,324ms
3	10.0.200.177	3,191ms	2,640ms	2,057ms
4	10.0.241.66	4,780ms	3,316ms	3,709ms
5	172.20.241.6	5,115ms	4,503ms	4,889ms
6	1.2.185.1	3,711ms	4,250ms	5,186ms

Nivelul Aplicației



Aplicațiile funcționează peste nivelul transport

application	data loss	throughput
file transfer/download	no loss	elastic
e-mail	no loss	elastic
Web documents	no loss	elastic
real-time audio/video	loss-tolerant	audio: 5Kbps-1Mbps video:10Kbps-5Mbps
streaming audio/video	loss-tolerant	same as above
interactive games	loss-tolerant	Kbps+
text messaging	no loss	elastic

La nivelul transport avem 2 protocoale principale

Transmission Control Protocol (TCP):

- transport sigur între emițător și receptor
- verificarea integrității **>segmentelor<**
- controlul fluxului
- controlul congestionării rețelei
- o conexiune este activă pe timpul transmisiei

User >Datagram< Protocol (UDP):

- datele nu sunt garantate că ajung la destinație
- verificarea integrității datagramelor

Câteva aplicații și protocoalele la nivelul transport

application	transport protocol
file transfer/download	TCP
e-mail	TCP
Web documents	TCP
Internet telephony	TCP or UDP
streaming audio/video	TCP
interactive games	UDP or TCP

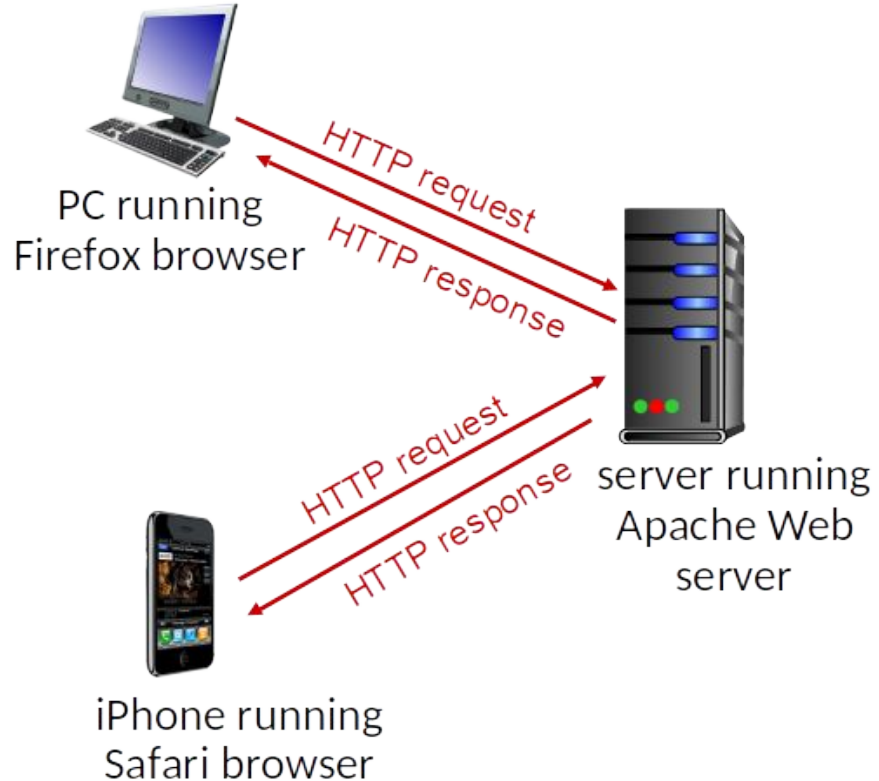
Porturi comune la nivelul transport TCP și UDP

7 Echo	554 RTSP	2745 Bagle.H	6891-6901 Windows Live
19 Chargen	546-547 DHCPv6	2967 Symantec AV	6970 Quicktime
20-21 FTP	560 rmonitor	3050 Interbase DB	7212 GhostSurf
22 SSH/SCP	563 NNTP over SSL	3074 XBOX Live	7648-7649 CU-SeeMe
23 Telnet	587 SMTP	3124 HTTP Proxy	8000 Internet Radio
25 SMTP	591 FileMaker	3127 MyDoom	8080 HTTP Proxy
42 WINS Replication	593 Microsoft DCOM	3128 HTTP Proxy	8086-8087 Kaspersky AV
43 WHOIS	631 Internet Printing	3222 GLBP	8118 Privoxy
49 TACACS	636 LDAP over SSL	3260 iSCSI Target	8200 VMware Server
53 DNS	639 MSDP (PIM)	3306 MySQL	8500 Adobe ColdFusion
67-68 DHCP/BOOTP	646 LDP (MPLS)	3389 Terminal Server	8767 TeamSpeak
69 TFTP	691 MS Exchange	3689 iTunes	8866 Bagle.B
70 Gopher	860 iSCSI	3690 Subversion	9100 HP JetDirect
79 Finger	873 rsync	3724 World of Warcraft	9101-9103 Bacula
80 HTTP	902 VMware Server	3784-3785 Ventrilo	9119 MXit
88 Kerberos	989-990 FTP over SSL	4333 mSQL	9800 WebDAV
102 MS Exchange	993 IMAP4 over SSL	4444 Blaster	9898 Dabber
110 POP3	995 POP3 over SSL	4664 Google Desktop	9988 Rbot/Spybot

Hypertext Transfer Protocol

<https://fmi.unibuc.ro/planuri-de-invatare>

- nu se menține starea apelurilor trecute
- se folosește de protocolul TCP la transport, protocol care asigură schimbul sigur de mesaje
- fiecare conexiune de TCP este o *sesiune* deschisă de comunicare



Tipuri de conexiuni HTTP

Persistent HTTP

- o conexiune TCP se deschide de către client și server
- mai multe obiecte HTTP circulă pe aceeași conexiune
- conexiunea TCP se închide

Non-persistent HTTP

- o conexiune TCP se deschide de către client și server
- un singur obiect este transferat prin acea conexiune
- pentru downloadarea mai multor obiecte, sunt necesare mai multe conexiuni
- conexiunea TCP se închide

Metode HTTP

https://www.w3schools.com/tags/ref_httpmethods.asp

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

- [GET](#) – object retrieval
- [HEAD](#) – GET with no body
- [OPTIONS](#) – get options associated with object
- [POST](#) – create, extend, submit data,
- [PUT](#) – create, replace, update with payload
- [PATCH](#) – partial update
- [DELETE](#) – delete resource
- [CONNECT](#) - intended only for use in requests to a proxy
- [TRACE](#) - debug purposes

Coduri de răspuns HTTP

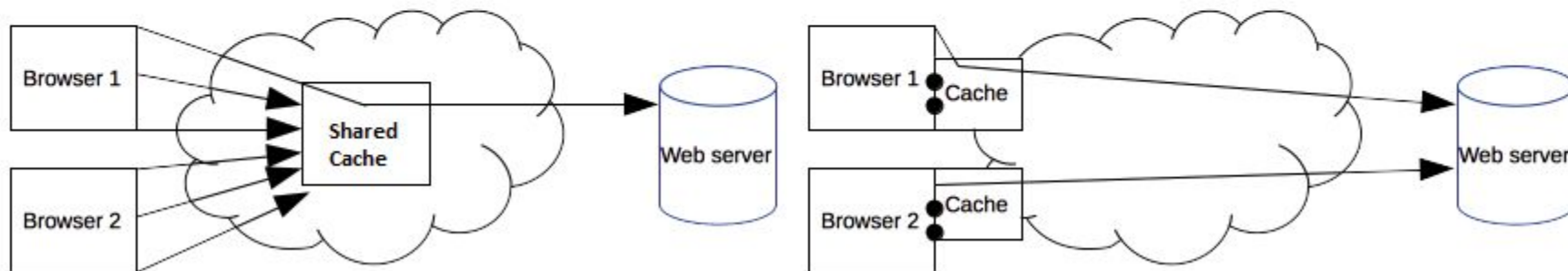
https://www.w3schools.com/tags/ref_httpmessages.asp

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

- 1xx: Information
- 2xx: Successful
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

Alte funcționalități

- caching - se stochează obiecte temporar (poate exista pe client, server sau în rețea)



- cookies - folosite pentru menținerea unei sesiuni cu privire la utilizatori, recomandări, coș de cumpărături, etc.



Vezi exemple de cod aici

<https://github.com/senisioi/computer-networks/tree/2022/capitolul2#https>

Probleme

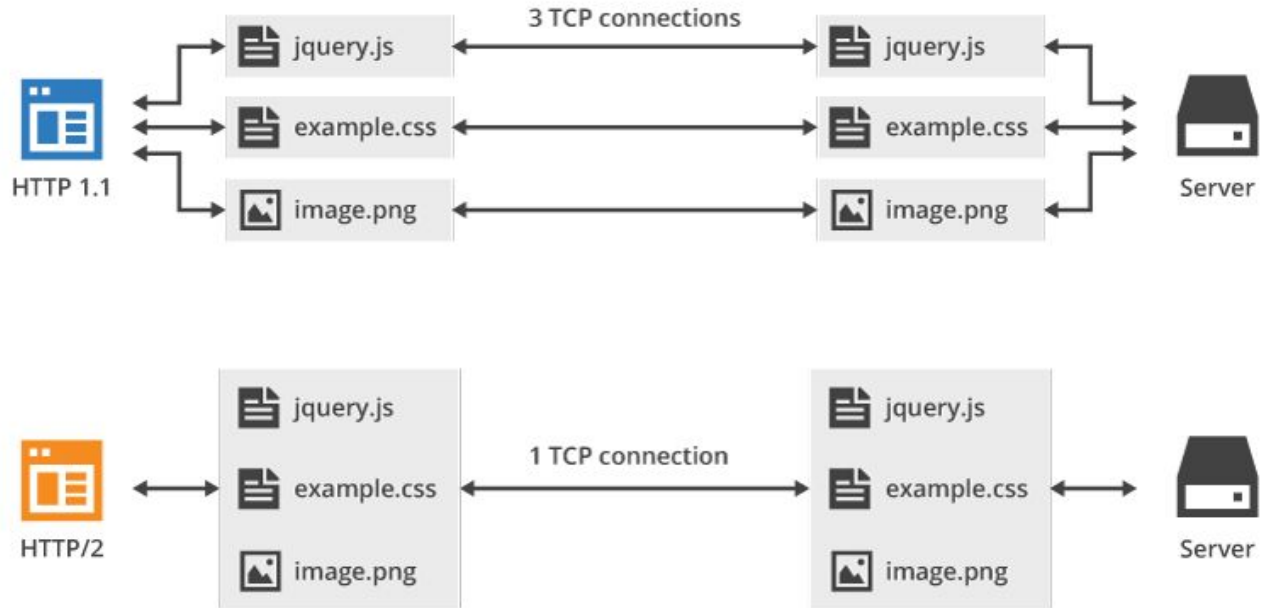
HTTP/1.1

- **FCFS first-come-first-served scheduling**, serverul răspunde la obiectele cerute în ordine
- obiecte mai mici riscă să stea blocate până la transmisia obiectelor mari, **head-of-line blocking (HOL)**
- retransmiterea segmentelor de TCP blochează transmisia tuturor obiectelor

HTTP/2 [[RFC 7540](#), 2015]

- obiectele cerute sunt returnate în funcție de prioritatea specificată de client (NU FCFS)
- posibilitatea de a trimite (push) obiecte care nu au fost cerute în prealabil către client
- divizarea obiectelor în frame-uri

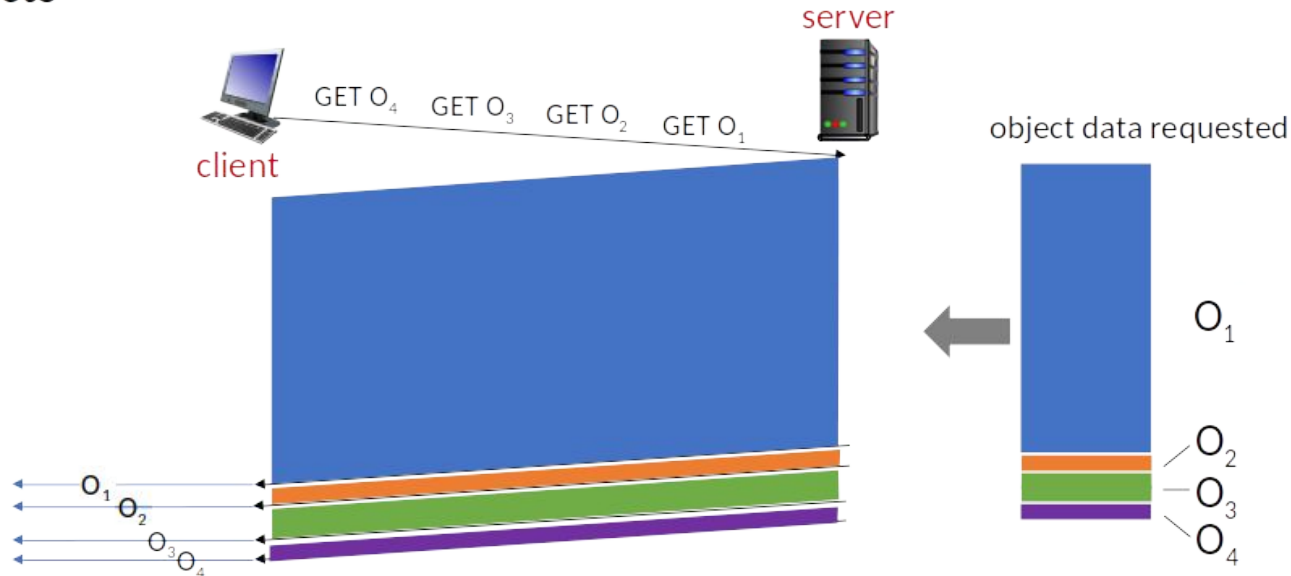
HTTP/2 vs. HTTP/1.1



<https://blog.cloudflare.com/http-2-for-web-developers/>

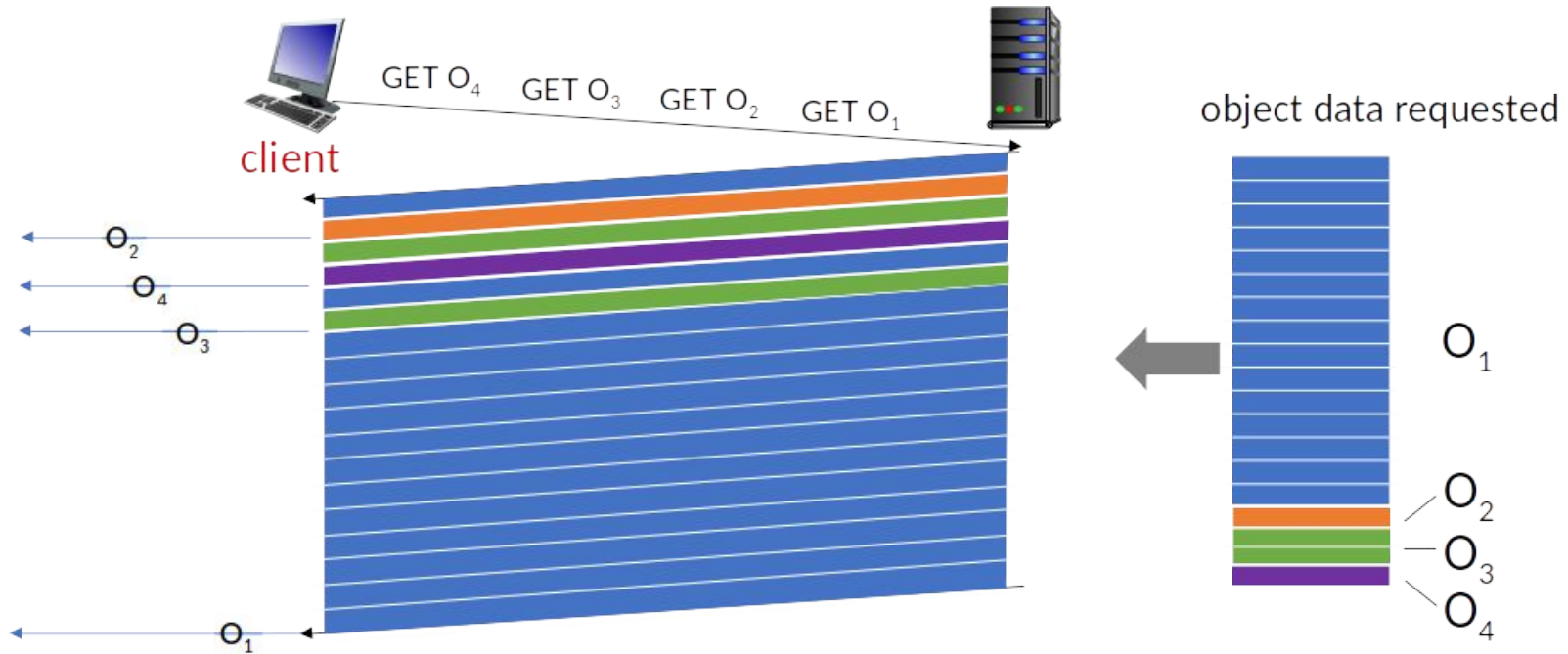
HOL blocking

HTTP 1.1: client requests 1 large object (e.g., video file) and 3 smaller objects

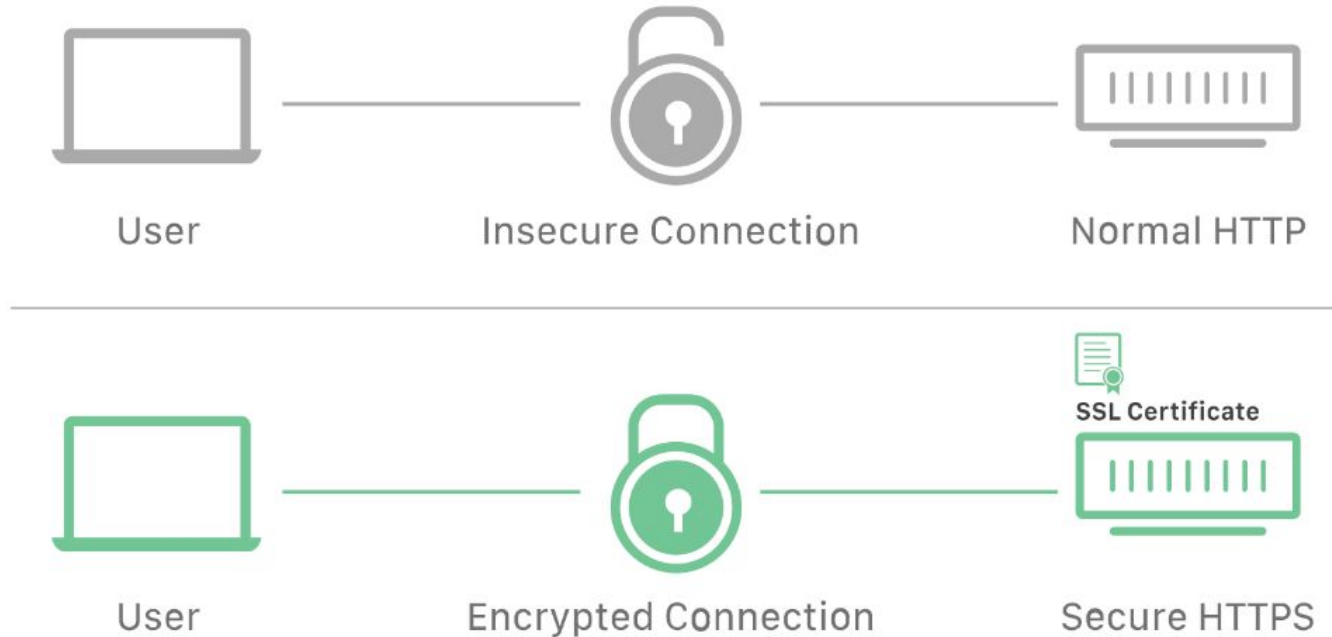


objects delivered in order requested: O₂, O₃, O₄ wait behind O₁

HTTP/2: HOL blocking



SSL/TLS (Secure Sockets Layer, Transport Layer Security)



<https://howhttps.works/>

Sfârșit