



Servidor proxy HTTP

Grupo 6

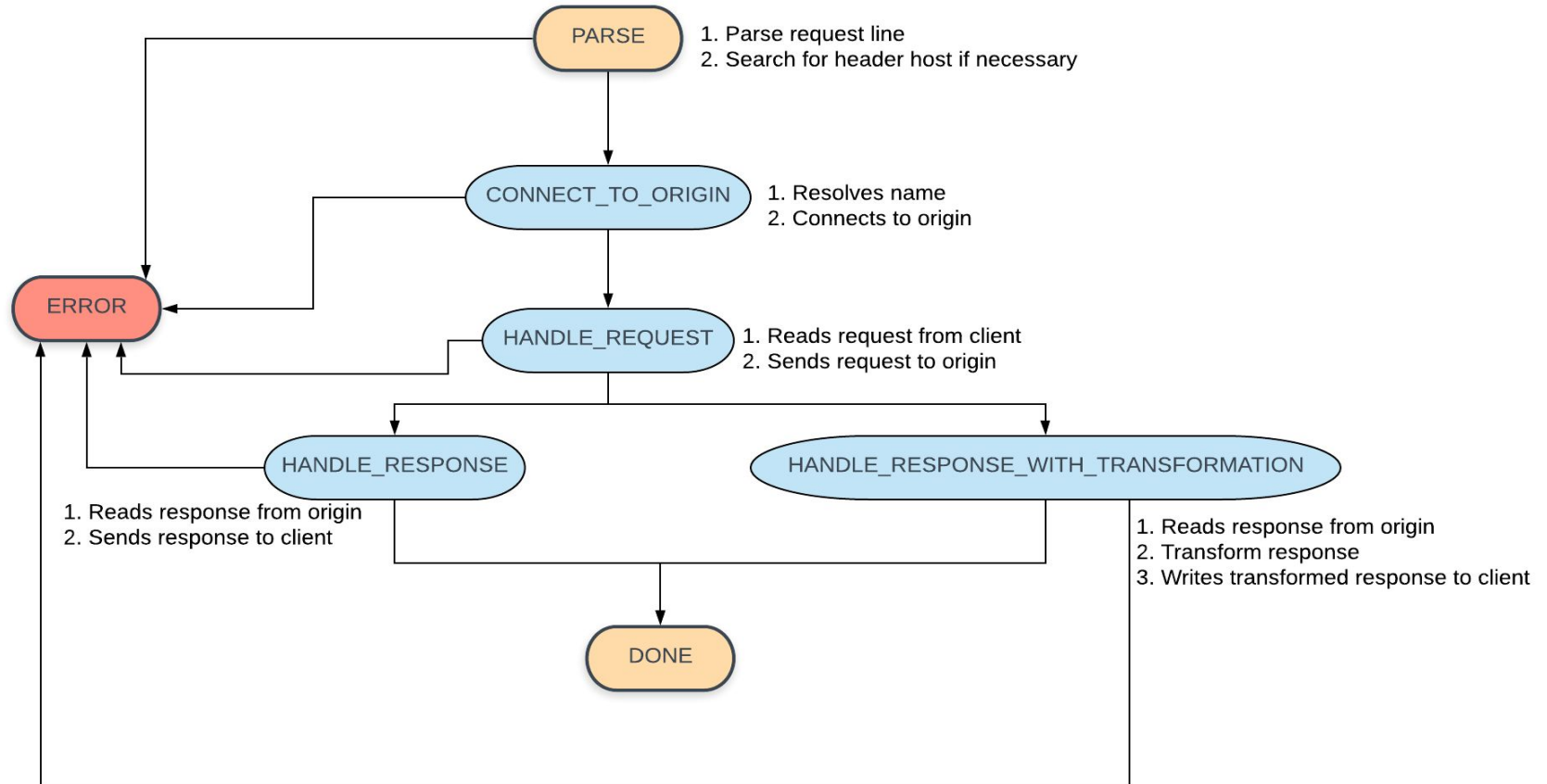
Dammiano, Agustin

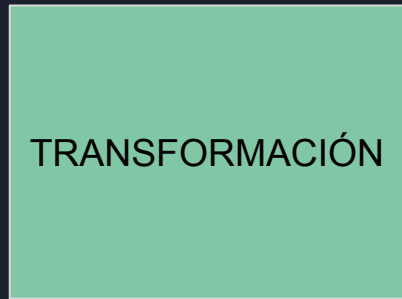
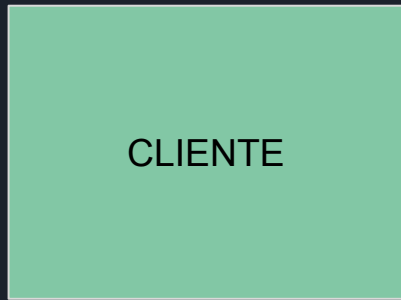
Donoso Naumczuk, Alan

Izaguirre, Agustin

Sanz Gorostiaga, Lucas

ESTADOS DEL PROXY





Host origin

- Puede venir en la primer línea o en el header HOST
- Parseamos la primer línea y los primeros headers hasta encontrarlo o quedarnos sin espacio.

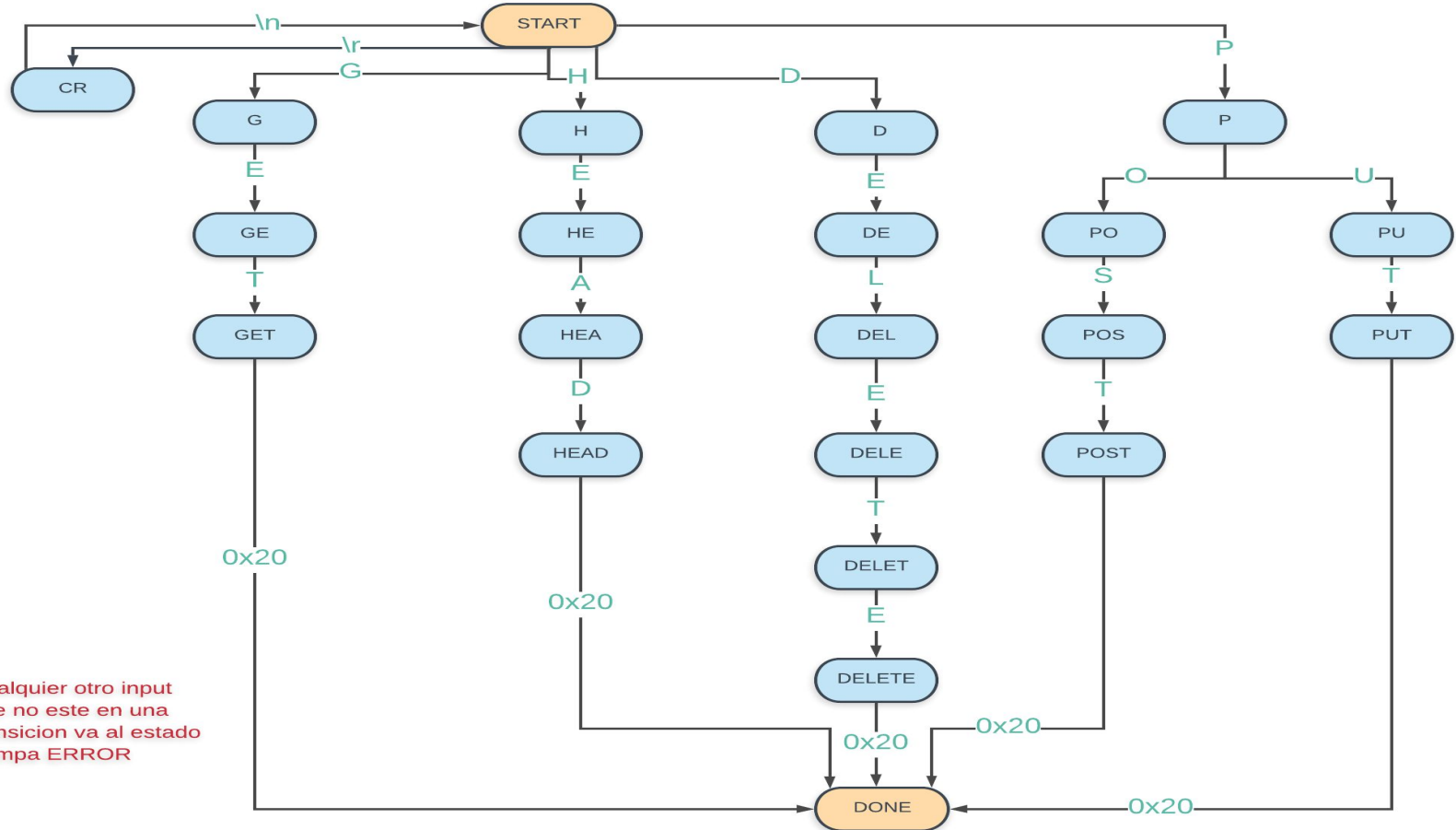




RFC 7230

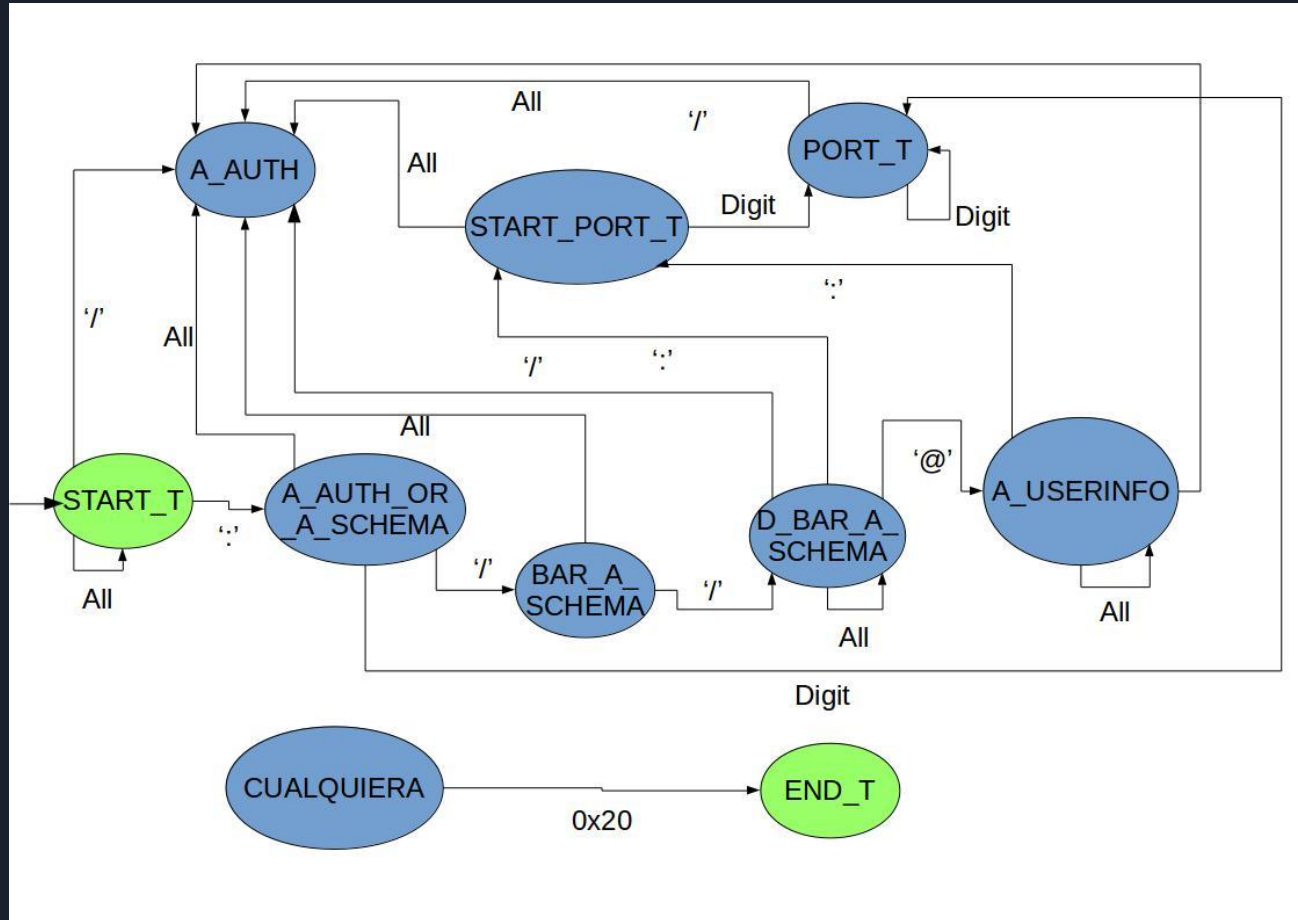
request-line = method SP request-target SP
HTTP-version CRLF

Parser del Método

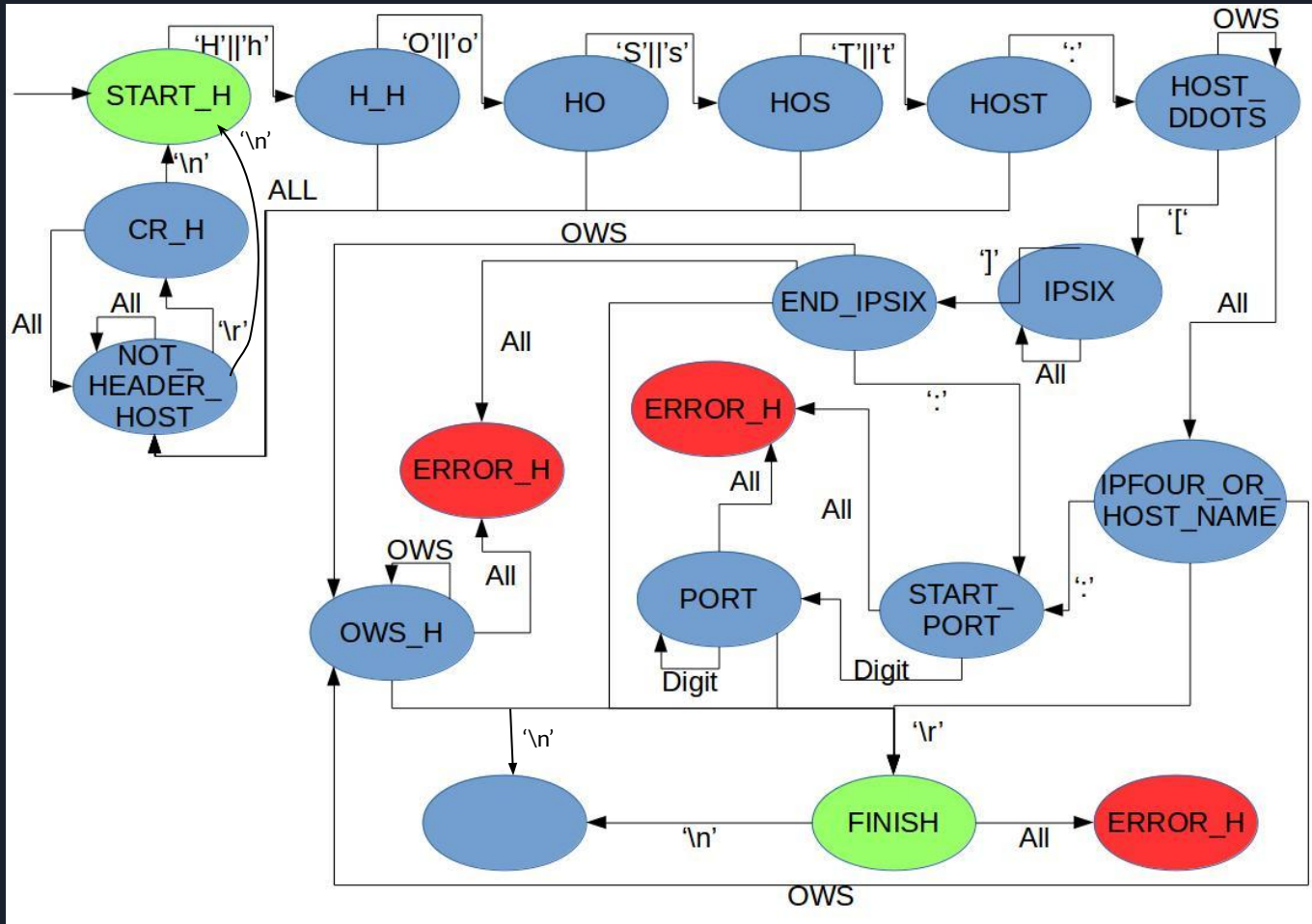


Cualquier otro input
que no este en una
transicion va al estado
trampa **ERROR**

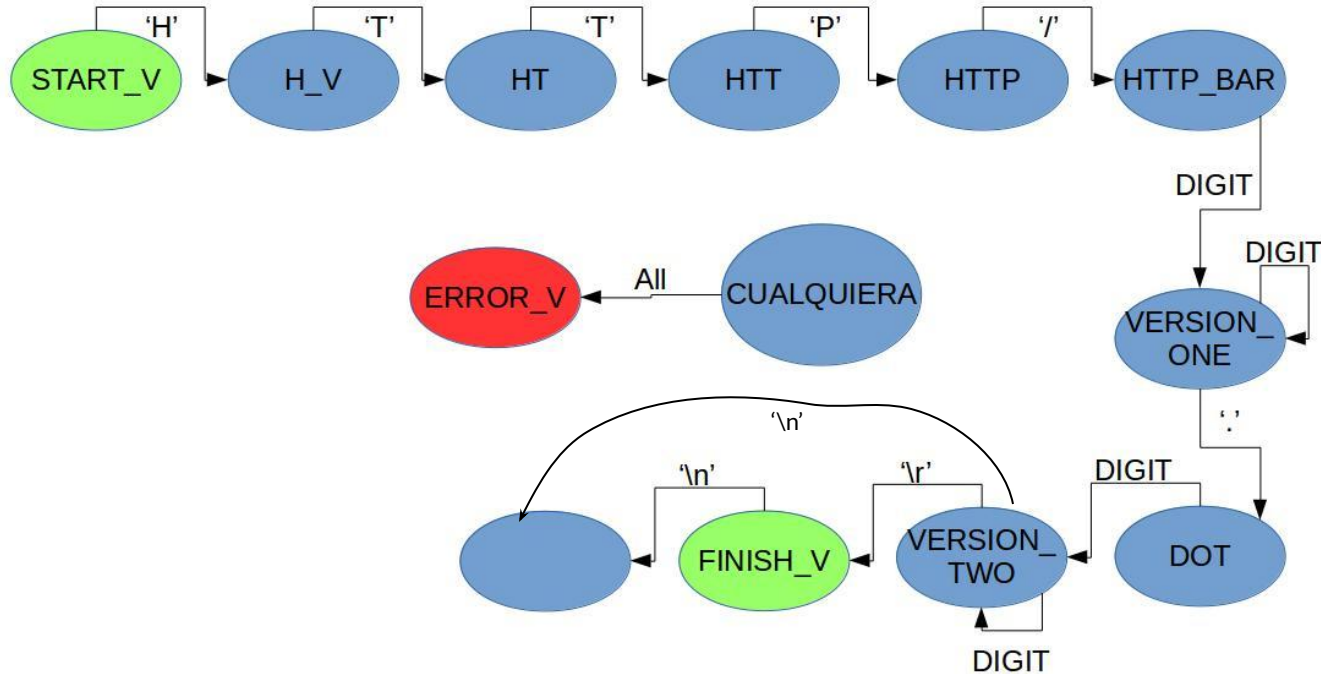
Parser del target



Parser del Header Host



Parser de la versión





Nuestro Protocolo

Características generales:

- Binario
- Orientado a conexión
- Provee autenticación
- Fácilmente extensible (versionado)



Operadores

GET: Solicita un recurso

SET: Intenta modificar un recurso

BYE: Cierra la conexión

* Tanto el **GET** como el **SET** son operadores condicionales



Comandos

get cmd

set cmd *transformer-command*

get mime

set mime *media-type1 media-type2 ... media-typeN*

get bf

set bf *integer*

get mtr

get mtr [*cn* | *hs* | *bt*]

bye



Implementación del Protocolo (ABNF)

Tipos de mensaje:

- Authentication-Request
- Authentication-Response
- Request
- Response



Authentication-Request

authentication-request = version username password

version = version-representation-bit-qty separator

version-value [fill-to-octet-multiple]

username = 1*CHAR %x00

password = 1*CHAR %x00



Authentication-Request

version = version-representation-bit-qty separator
version-value [fill-to-octet-multiple]

version-representation-bit-qty = "1"

separator = "0"

version-value = "0"

fill-to-octet-multiple = 1*BIT

;in v0 or v1, fill-to-octet-multiple can be any
combination of 5 bits



Authentication-Response

```
authentication-response = general-status version-status  
                          authentication-status [version / 5BIT]  
; version if version-status is 1, otherwise fill with  
; any 5 bits
```

```
general-status           = ok / error
```

```
version-status           = ok / error
```

```
authentication-status    = ok / error
```

```
ok                       = "0"
```

```
error                    = "1"
```


Request

request = opcode resource-id [time-tag] [data]

;get operator does not use data

;bye operator does not use data nor time-tag

opcode = bye / get / set

resource-id = buffer-size-id / media-types-id /
command-id / cn-metric-id / hs-metric-id / bt-metric-id
/ all-metrics-id

;in bye operation the resource-id is ignored

time-tag = 64BIT

data = *common-data-block final-data-block

Request

opcode = bye / get / set

bye = "00"

get = "01"


set = "10"

get-request = get resource-id time-tag

set-request = set resource-id time-tag data

bye-request = bye 6BIT

Request



resource-id = buffer-size-id / media-types-id /
command-id / cn-metric-id / hs-metric-id / bt-metric-id
/ all-metrics-id

buffer-size-id	= "000001"
media-types-id	= "000010"
command-id	= "000011"
cn-metric-id	= "000100"
hs-metric-id	= "000101"
bt-metric-id	= "000110"
all-metrics-id	= "000111"

Request

data = *common-data-block final-data-block

common-data-block = %x00 [fill-to-**k**-bytes-of-data]
start-data-byte
1*(**k** - **n**)concret-data

final-data-block = is-final-byte
[fill-to-**k**-bytes-of-data]
start-data-byte
1*(**k** - **n**)concret-data

start-data-byte = %x80

fill-to-k**-bytes-of-data** = 1*%x00

is-final-byte = %x10

concret-data = OCTET

;**k** is the number of bytes per block

;**n** is the number of bytes with concrete data



Response

```
response = general-status opcode-status time-tag-status  
           id-status 4BIT [time-tag] [data]  
; get: time-tag only if you get a new version of the  
;       resource, general-status 0  
; get: data only if you get a new version of the  
;       resource, general-status 0  
; set: time-tag only if you override the resource,  
;       general-status 0  
; set: no data
```



Response

response = general-status opcode-status time-tag-status
id-status 4BIT [time-tag] [data]

opcode-status = ok / error

time-tag-status = ok / error

id-status = ok / error

Ejemplo

Authentication request (v0)

1 0 0 00000 0x750x730x650x720x00 0x700x610x730x730x00
vers fill 'u' 's' 'e' 'r' '\0' 'p' 'a' 's' 's' '\0'

Authentication response (version error v2)

1 1 0 11 0 10

Authentication response (no error)

0 0 0 00000
fill

Authentication response (auth error)

1 0 1 00110
fill

Ejemplo

Get request

01 000011 0x000x000x000x000x000x000x000x000x00
op id time-tag

Get response

(With data blocks of 4 bytes, data = mycommand)

1 0 0 1 1111 0x000x000x000x000x5C0xF00x180xC2
status fill time-tag: time command

0x00 0x000x000x000 0x80 0x6D 0x00 0x80 0x790x630x6F0x6D
com fill zeros sdb 'm' com sdb 'y' 'c' 'o' 'm'
0x10 0x80 0x6D0x610x6E0x64
fin sdb 'm' 'a' 'n' 'd'

Ejemplo

Get request

01 000011 0x000x000x000x000x5C0xF00x180xC2
op id time-tag

Get response (No data because you've the last version)

0 0 0 0 1111
status fill

Bye request

00 000011
op fill



Decisión sobre Headers Hop by Hop

Header	Acción
Connection	Pisar por Connection: close
Keep-Alive	Censurar
Proxy-Authenticate	Dejar pasar
Proxy-Authorization	Dejar pasar
TE	Dejar pasar (siempre soporta chunked)



Decisión sobre Headers Hop by Hop

Header	Acción
Trailer	Sin transform: Dejar pasar Con transform: Censurar y terminamos de leer con el chunked 0.
Transfer-Encoding	Pisar en respuesta con transform por chunked
Upgrade	Censurar