network protocol

a set of rules that define how to transmit and receive data

TCP   UDP   HTTP

Network protocols

connection-oriented

connection between client and server is
established before data can be sent

handshake

a 3-step connection establishment process

reliable

lost packets are detected and retransmitted

sequence numbers

allow receivers to discard duplicate packets and
properly sequence reordered packets

acknowledgments

allow senders to determine when to
retransmit lost packets

order

all bytes received will be in the same order
as those sent

checksums

ensure data correctness

TCP

prioritizes reliability
over time

flow control

limit the rate a sender transfers data to a receiver

congestion control

the rate of data entering the network is controlled

Network protocols

no connection

sender transmits messages to the receiver without verifying the readiness or state of the receiver

checksums

ensure data correctness

no reliability

messages may get lost on the way

broadcast

a single message can be transferred to **all** recipients on the subnet simultaneously

no acknowledgments

UDP

prioritizes time over reliability

multicast

a single message is routed **only** to **intended** recipients

no retransmission

no order

no flow control

no congestion control

UDP is used for
- video and audio live streaming
- online games
- automatic detection of devices and services on a computer network
- applications where latency is a critical concern and we can afford losing packets occasionally

Network protocols

request/response

the client submits an HTTP request message to the server
and the server sends back an HTTP response message

TCP

older HTTP protocol versions use TCP
as an underlying transport protocol

QUIC
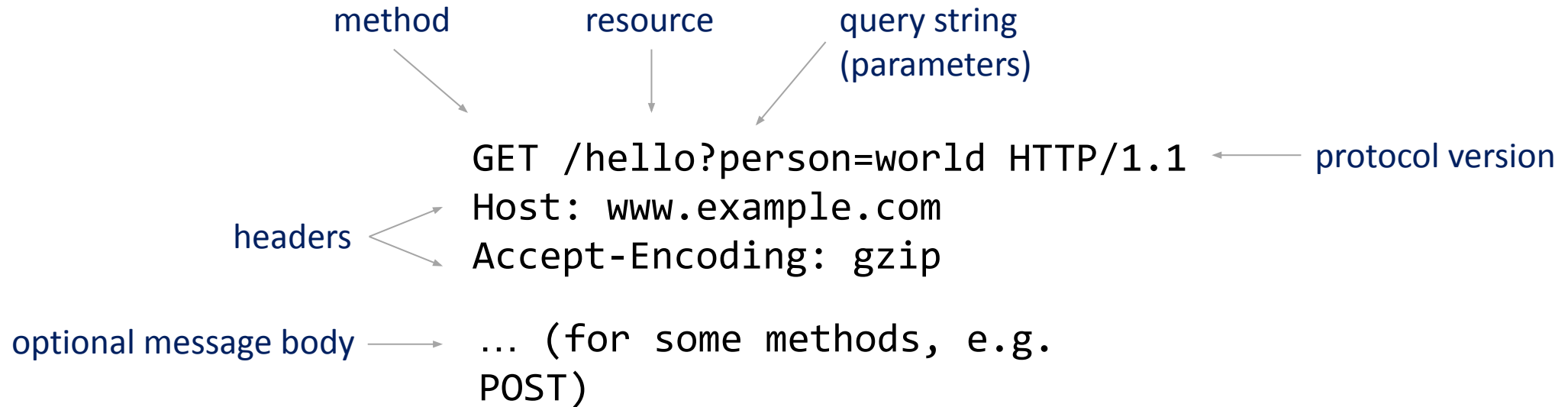
HTTP/3 uses QUIC protocol as a transport

persistent connection

a single TCP connection can be reused
for more than one request

multiplexing

multiple requests are sent over the same TCP
connection without waiting for responses

compression

allows content to be compressed on the server
before transmission to the client

HTTP

HTTP request

method   resource   query string
                                  (parameters)

GET /hello?person=world HTTP/1.1 ⟵—— protocol version
Host: www.example.com
Accept-Encoding: gzip

headers

optional message body ——⟶ ... (for some methods, e.g.
                              POST)

HTTP response

protocol version   status code   status message

```
HTTP/1.1 200 OK
Date: Mon, 04 May 2017 21:00:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 101
Content-Encoding: gzip
```

headers

optional message body ⟶ `<!DOCTYPE html... (here comes the requested web page)`