Internet and Web Applications, Übung 2

Henry Haustein

Aufgabe 1: Representational State Transfer

Die Vorteile sind

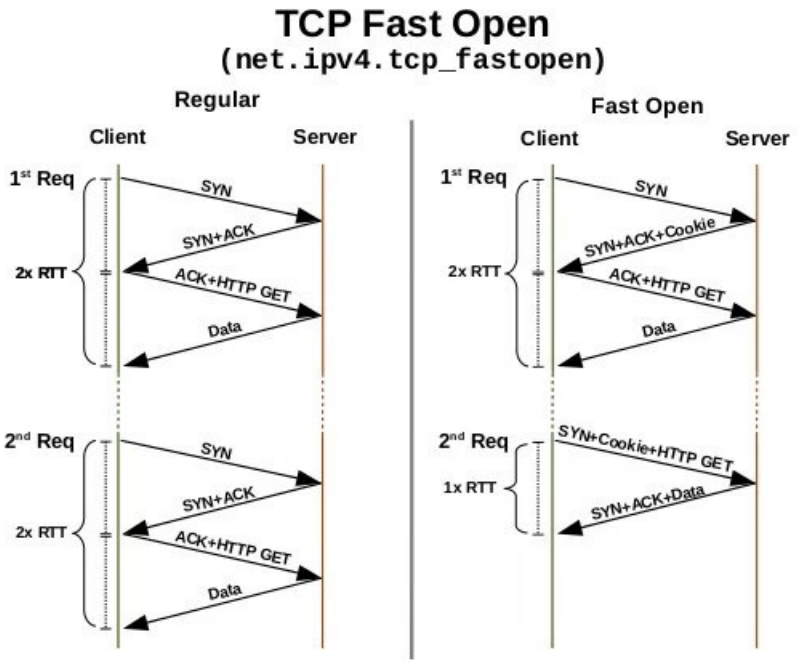
- Skalierbarkeit: Client und Server sind getrennt
- Flexibilität: API kann ohne Probleme von einem Server zu einem Server portiert werden
- Unabhängigkeit: REST ist unabhängig von der restlichen Architektur

Aufgabe 2: TCP Fast Open

TCP Fast Open wurde entwickelt um Daten direkt beim Handshake mit auszutauschen. Damit spart man sich eine voll Round-Trip-Time bevor Daten ausgetauscht werden können (beim klassischen TCP). Man steigert die Effizienz.

Aus https://datatracker.ietf.org/doc/html/rfc7413: Performing TCP Fast Open:

1. The client sends a SYN with data and the cookie in the Fast Open option.
2. The server validates the cookie:
   - If the cookie is valid, the server sends a SYN-ACK acknowledging both the SYN and the data. The server then delivers the data to the application.
   - Otherwise, the server drops the data and sends a SYN-ACK acknowledging only the SYN sequence number.
3. If the server accepts the data in the SYN packet, it may send the response data before the handshake finishes. The maximum amount is governed by TCP's congestion control [RFC5681].
4. The client sends an ACK acknowledging the SYN and the server data. If the client's data is not acknowledged, the client retransmits the data in the ACK packet.
5. The rest of the connection proceeds like a normal TCP connection. The client can repeat many Fast Open operations once it acquires a cookie (until the cookie is expired by the server). Thus, TFO is useful for applications that have temporal locality on client and server connections.



Aufgabe 3: HTTP/2 and HTTP/3

(a) HTTP/2 stellt einen schnelleren Transport der Daten durch Multiplexing und Header-Komprimierung sicher. In einer TCP Verbindung können mehrere Channel aufgebaut werden und damit mehrere Requests parallel verarbeitet werden.
Aus https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-27: *HTTP/2 introduced a binary framing and multiplexing layer to improve latency without modifying the transport layer. However, because the parallel nature of HTTP/2's multiplexing is not visible to TCP's loss recovery mechanisms, a lost or reordered packet causes all active transactions to experience a stall regardless of whether that transaction was impacted by the lost packet.*

(b) Unterschiede:

- HTTP/3 basiert auf QUIC und nicht mehr auf TCP
- TLS 1.3 auf QUIC-Ebene
- konstante Verbindung → weniger Datenpakete, weil kein Header jedes mal gesendet werden muss
- Fehlerkorrektur auf QUIC-Ebene
- bei Paketverlusten stockt die Verbindung nicht mehr, weil nicht mehr gewartet werden muss
- HTTP/3 verzichtet auf einleitende Handshakes
- HTTP/3 ist nicht mehr an IP-Adressen gebunden, sondern an individuelle Verbindungs-IDs, die selbst bei einem Netzwerkwechsel einen konstanten Download ermöglichen

Aufgabe 4: Cookie mechanism

(a) Cookies sind Key-Value-Paare, die bei jedem Requests vom Browser im Header-Feld `cookie` mitgesendet werden. Der Server kann in der Response auch Cookies mittels Header-Feld `set-cookie` setzen.

(b) mittels Header-Feld `authentication: bearer <token>`, mittels PHP-Session-IDs, mittels Fingerprinting (https://amiunique.org/fp)