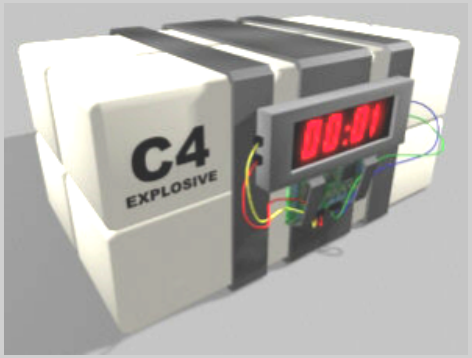


HTTP Timeouts

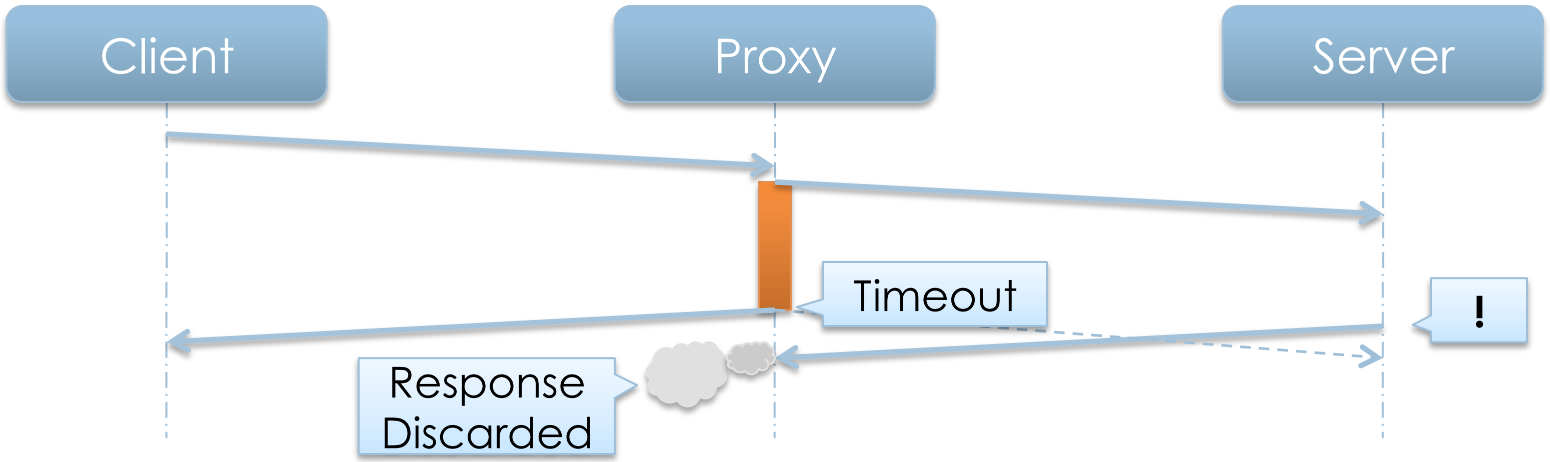
draft-thomson-hybi-http-timeout-00

Martin Thomson, Salvatore Loreto, Greg Wilkins

IETF-78/80

Request Timeout

- Long-polling is widely used
- *Problem:* no information on how long to hold a request open
  - Conservative guesses are made to avoid timeouts at intermediaries, NAT bindings, etc…



Request-Timeout* Header

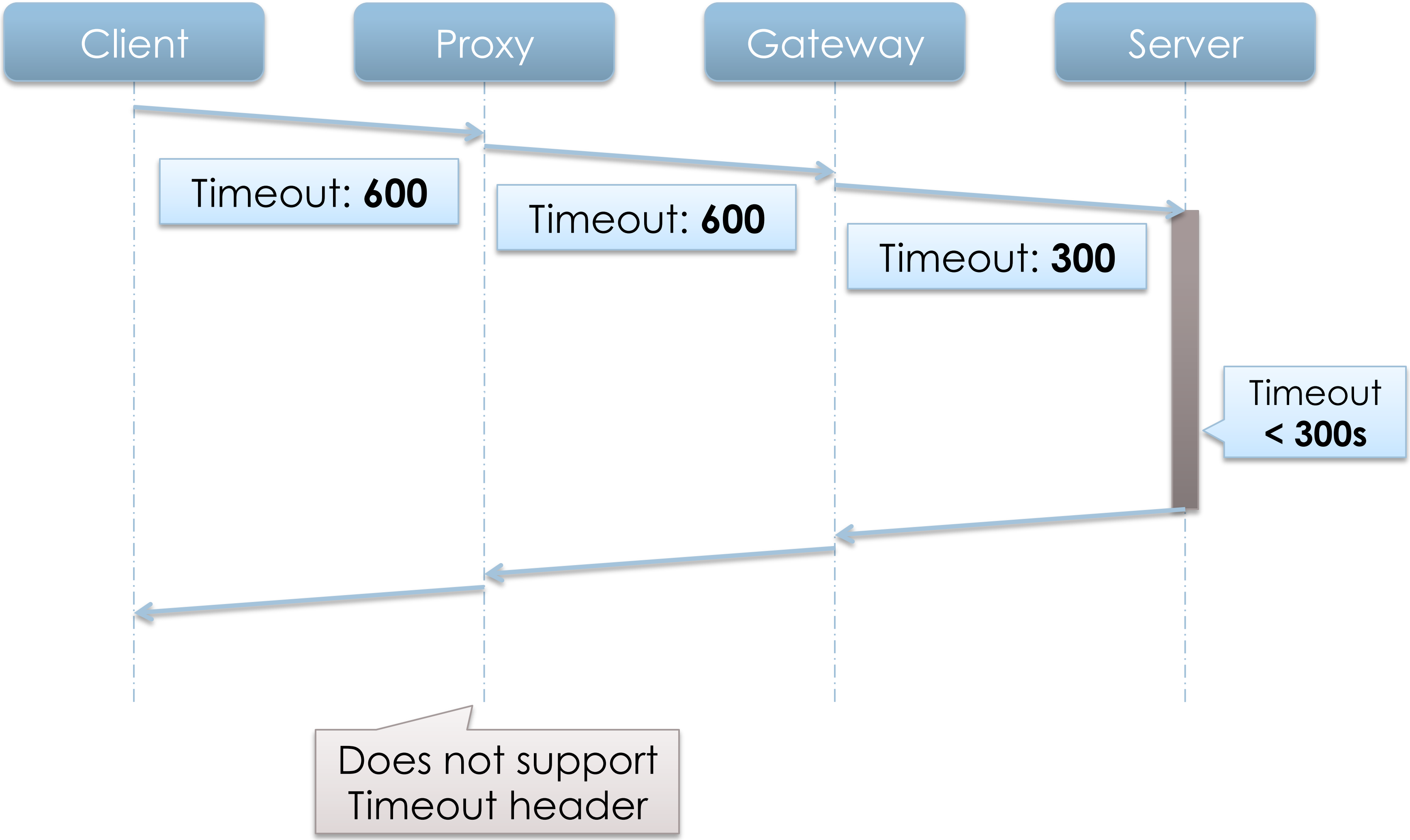
- Advertise client timeout requirements
  - Intermediaries can reduce *Request-Timeout* according to policy or their knowledge of connection timers
  - Header gives intermediaries an explicit indicator that this is a long-lived request
  - Origin server sees lowest value

```
Request-Timeout    = "Request-Timeout" ":" timeout-value
timeout-value      = 1*DIGIT ; in seconds
```

- Proposed:

```
Prefer: response-within=100
```

Request-Timeout



Client

Proxy

Gateway

Server

Timeout: **600**

Timeout: **600**

Timeout: **300**

Timeout
< 300s

Does not support
Timeout header

Idle Connection Timeout

- Idle HTTP/1.1 connections are reusable
  - …in theory
  - in practice, not so much (see §8.1.4 of RFC 2616)
- *Problem:* Connection reuse can fail
  - The connection could be closed at the other end when a request is started
  - Bigger problem for non-idempotent requests
- Many clients seek to avoid the problem by making new connections for POST

Connection-Timeout* Header

- Hop-by-hop header
  - Token is added to the *Connection* header
- Both peers advertise how long they are willing to keep the connection open
- **Timeouts apply to upgraded connection**

```
Connection-Timeout = "Connection-Timeout" ":" timeout-value
timeout-value      = 1*DIGIT ; in seconds
```

- Proposed:

```
Keep-Alive: timeout=100;max=300
```

http://tools.ietf.org/id/draft-thomson-hybi-http-timeout-00

Comments