Incremental HTTP Messages

IETF 121 Dublin

Kazuho Oku*, Tommy Pauly, Martin Thomson

Are HTTP messages transferred incrementally?

An HTTP message can be parsed as a stream for incremental processing or forwarding downstream. However, **senders and recipients cannot rely on incremental delivery of partial messages, since some implementations will buffer or delay message forwarding** for the sake of network efficiency, security checks, or content transformations.

RFC 9110 Section 7.6

Some applications need incremental delivery

- Server-Sent Events
- gRPC
- Resumable Uploads
- Chunked OHTTP
- …

Some applications need incremental delivery

- Server-Sent Events
- gRPC
- Resumable Uploads
- Chunked OHTTP
- …

they work

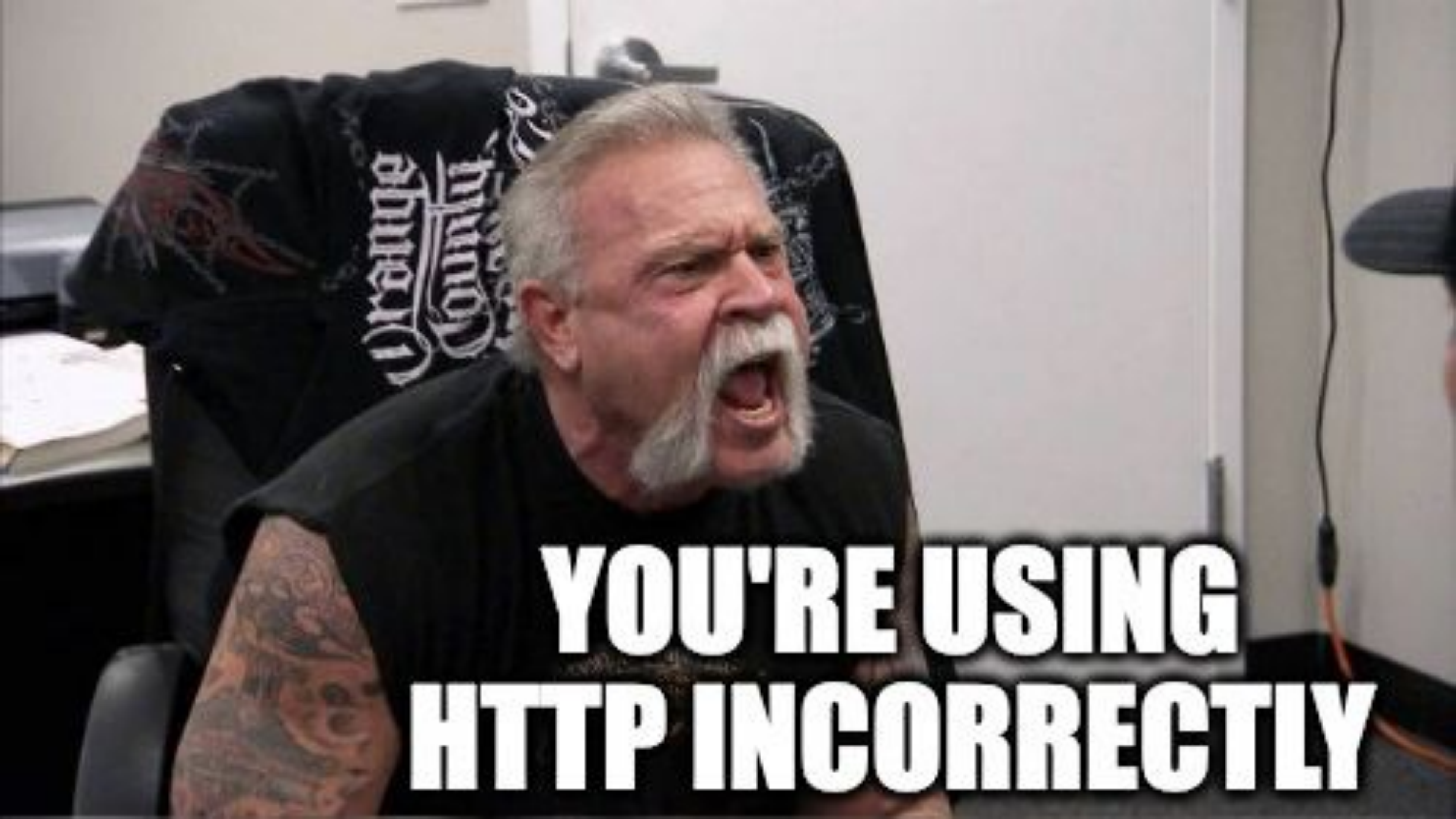
Some applications need incremental delivery

- Server-Sent Events
- gRPC
- Resumable Uploads
- Chunked OHTTP
- ...

they work
until buffering intermediaries appear on the path

INCREMENTAL DELIVERY ISN'T GUARANTEED

BUT I NEED IT


YOU'RE USING
HTTP INCORRECTLY

JUST DO IT

NEXT TIME, FOLLOW THE BEST PRACTICES
imgflip.com

applications might persuade intermediary operators to update the config

Syntax: proxy_request_buffering on | off;
Default: proxy_request_buffering on;
Context: http, server, location

This directive appeared in version 1.7.11.

Enables or disables buffering of a client request body.

When buffering is enabled, the entire request body is read from the client before sending the request to a proxied server.

When buffering is disabled, the request body is sent to the proxied server immediately as it is received. In this case, the request cannot be passed to the next server if nginx already started sending the request body.

from Nginx documentation

intermediaries can forward incrementally

- intermediaries (often) have a knob to forward incrementally
  - as can be seen in the Nginx documentation
- however, intermediaries prefer not to, unless required
  - due to security, efficiency, flexibility concerns
  - recall Slowloris attack

We are in a lose-lose situation

- some applications want incremental behavior
  - it makes their design a lot simpler
- intermediaries don't want to provide incremental by default
- as a result, we see interoperability issues <u>each time</u> such applications are deployed, leading to either
  - redesign of the application (becomes complex), or
  - config change at the intermediary

Can we do better?

Our proposal: send a signal

- Incremental: ?1
  - sender of the HTTP message sets this header field,
  - indicates that the message should be delivered incrementally
- intermediaries can recognise the header field and either:
  - forward the message incrementally, or
  - refuse with an error response (rather than buffering)
    - because explicit errors are better than timeouts caused by buffering

The situation becomes a win-win

- lesser interoperability issues as intermediaries support the new signal
- applications can rely on incremental delivery
- intermediaries no longer need manual config changes

Thoughts?

Open issues

- #7 - allow for Incremental: ?0
- #8 - signal for falling back to buffered rather than reject
- #9 - let intermediaries signal HTTP servers the intent
- #10 - buffering delays and interaction with incremental delivery