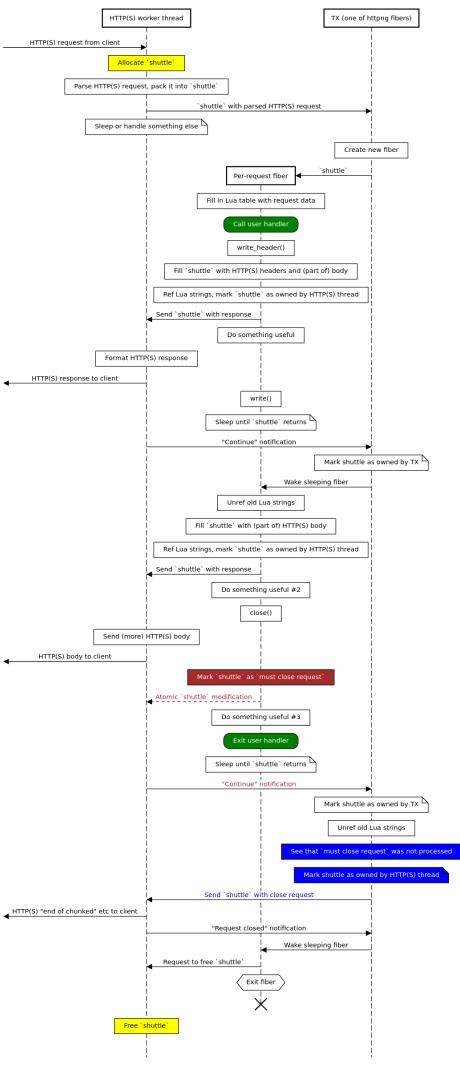HTTPNG threads interaction concept (proposed design, stage 3, "unlucky and too late")

```
┌─────────────────────┐                              ┌──────────────────────┐
│ HTTP(S) worker thread │                              │ TX (one of httpng fibers) │
└─────────────────────┘                              └──────────────────────┘
```

HTTP(S) request from client →

Allocate `shuttle`

Parse HTTP(S) request, pack it into `shuttle`

`shuttle` with parsed HTTP(S) request →

Sleep or handle something else

Create new fiber

Per-request fiber ← `shuttle`

Fill in Lua table with request data

Call user handler

write_header()

Fill `shuttle` with HTTP(S) headers and (part of) body

Ref Lua strings, mark `shuttle` as owned by HTTP(S) thread

← Send `shuttle` with response

Do something useful

Format HTTP(S) response

← HTTP(S) response to client

write()

Sleep until `shuttle` returns

"Continue" notification →

Mark shuttle as owned by TX

← Wake sleeping fiber

Unref old Lua strings

Fill `shuttle` with (part of) HTTP(S) body

Ref Lua strings, mark `shuttle` as owned by HTTP(S) thread

← Send `shuttle` with response

Do something useful #2

close()

Send (more) HTTP(S) body

← HTTP(S) body to client

Mark `shuttle` as `must close request`

Atomic `shuttle` modification

Do something useful #3

Exit user handler

Sleep until `shuttle` returns

"Continue" notification →

Mark shuttle as owned by TX

Unref old Lua strings

See that `must close request` was not processed

Mark shuttle as owned by HTTP(S) thread

← Send `shuttle` with close request

← HTTP(S) "end of chunked" etc to client

"Request closed" notification →

← Wake sleeping fiber

← Request to free `shuttle`

Exit fiber

✗

Free `shuttle`