PTTH

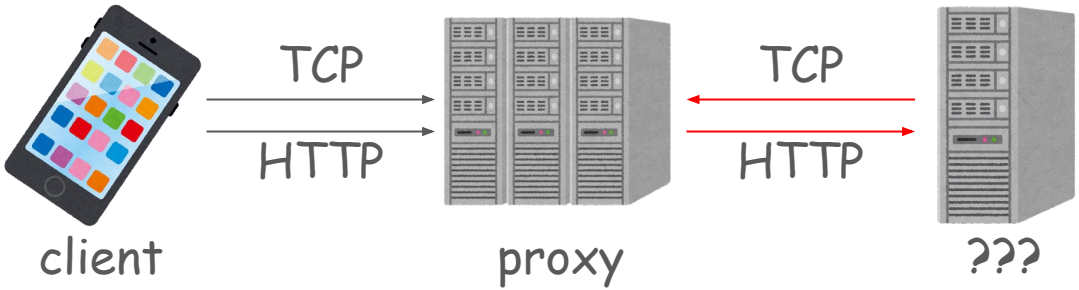
CDN / reverse proxy use cases

PTTH BoF, July 2025

Kazuho Oku, Fastly

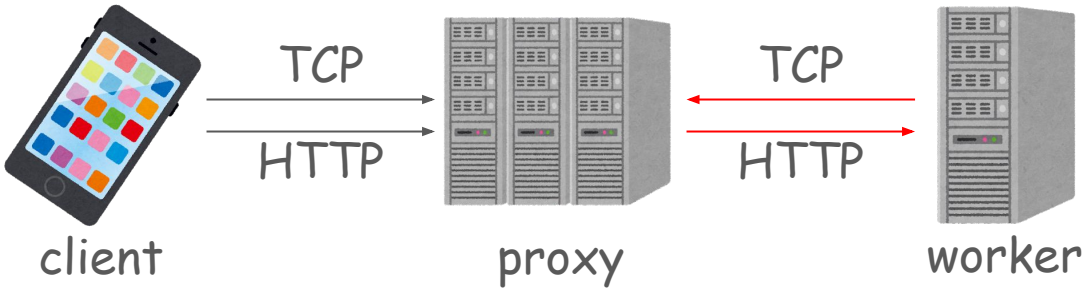
terminology

- in PTTH, the HTTP server that processes the client's request is a TCP client
  - or worse, it might act as an HTTP client to setup the channel for acting as a HTTP server
- how should we call it?



terminology

- let's call it "worker," at least for the time being
  - as it does the "work" of processing the HTTP request



hidden servers

- status quo: backend servers rely on firewalls that open holes for CDNs
  - firewall rules depend on fixed IP addresses
  - configuration is tedious and inflexible
- with PTTH:
  - backend workers establish outbound connections to CDNs using hostnames
  - no need to configure firewalls
  - CDNs can update IP addresses dynamically via DNS

load balancing / dynamic scaling

- status quo:
  - adding a backend server (or a group of servers behind a load balancer) requires manual updates to the CDN configuration
- with PTTH:
  - backend operators can clone and start PTTH worker instances that contain the necessary credentials
  - these workers automatically connect to the CDN and begin receiving requests - eliminating the need for manual configuration and having load balancers

prioritization

- status quo:
  - reverse proxies route requests to different backend servers based on the properties of the HTTP requests
  - prioritizing these requests is challenging, because:
    - reverse proxies often cannot determine if opening more connections will increase throughput
    - requests are frequently queued by load balancers near the backend servers
- with PTTH:
  - incoming requests are buffered in a priority queue and then fetched directly by the PTTH workers
  - the number of established connections and concurrent requests per connection serve as dynamic indicators of available backend capacity

limiting the concurrency of heavyweight requests

- status quo:
  - clients submit requests to a queue server through HTTP, then periodically poll to check for completion
  - workers retrieve requests from the queue, process them, and update the poll statuses accordingly
  - this process is complex that uses a separate queue server, involves multiple HTTP requests, requires matching of requests/responses
- with PTTH:
  - the reverse proxy functions as a queue
  - clients send HTTP requests and wait for responses, while PTTH workers pull HTTP requests from the reverse proxy and submit HTTP responses

why standardize?

- status quo:
  - requires custom-made PTTH-to-HTTP gateways to be installed into CDN customer's server instances
- with PTTH:
  - standardization of PTTH will enable HTTP servers to directly support PTTH, removing the dependency on custom PTTH-to-HTTP gateways
  - during the HTTP Workshop 2024, HTTP server vendors expressed their interests

FAQ: authentication of PTTH workers

- Q. When Web PKI is based on servers providing certificates, how are PTTH workers supposed to authenticate themselves?
- A. Web PKI authenticates the CDN (as the origin) to the client. Authentication of PTTH workers is an internal matter for the CDN, and it is possible to use different methods, such as PTTH workers providing a login credential to the CDN.

summary: benefits of PTTH

- simplified and flexible management of hidden servers
- facilitates autonomous scaling, load balancing, and fine-grained prioritization
- enhances processing of heavyweight requests using HTTP
- standardization eliminates the need for installing custom gateways when providing PTTH-like solutions