FlexHTTP:
An Intelligent and Scalable HTTP Version Selection System

Mengying Zhou, Zheng Li, Shihan Lin, Xin Wang, Yang Chen
School of Computer Science
Fudan University
HTTP/1.1 Has Fallen Behind

- Optimizing Internet traffic has been more important than ever before
- Users expect a faster and smooth experience with online services

1. Stubborn page loading pipeline
2. One open request per connection
3. Duplication of data

Hardly to satisfy these demands
Two Separate Improvement Approaches

### Approach 1: TCP-based
- BBR congestion algorithm
- TLS 1.2
- SPDY

---

**HTTP/2**

- Request multiplexing
- Server push
- Stream prioritization

### Approach 2: UDP-based
- On the top of UDP
- A reliable protocol
- QUIC

---

**HTTP-over-QUIC**

- Faster connection establishment
- Removal of TCP’s Head-of-Line problem
- Improved recovery mechanism

---

Evidence Tells Us: Neither of Them is Always Better

Network condition: low bandwidth, high latency, high packet loss

HTTP/3 👍 HTTP/2 🙅

Web page structure: few small files

HTTP/3 ⚔️ HTTP/2 👍

RQ: Could the HTTP version be automatically selected to achieve a better performance?

Performance differences between H2 and H3 in different network conditions

Performance differences between H2 and H3 in different web page structures
FlexHTTP:
An Intelligent and Scalable HTTP Version Selection System

Towards a flexible HTTP version selection according to network conditions and page structures

Equipping a supervised machine learning-based classifier in the browser as a plug-in
Challenge I: Network Condition Measurement Overhead

• Classifier needs the information on network condition and web page structure as the input

• Network condition feature is link-based

• The possible number of client-server pairs and corresponding features would be huge
Challenge II: Classifier Updating

- Network conditions are constantly fluctuating
- An immutable classifier will be out-of-date and make a wrong HTTP version selection

How to update classifier with ensuring both timeliness and generality?
Challenge I: How to reduce the measurement traffic to a moderate level
Similar Network Condition within A Region

Could we use a representative agent to aggregate network conditions?

31ms-0% loss rate-5Mbps

29ms-0% loss rate-5Mbps

32ms-0.01% loss rate-5Mbps

Similar network condition in the same region*

Aggregative Network Condition Measurement

- Deploying an agent server to aggregate the network conditions by representing the group of nearby web services in the same region


- Servers within this zone can reach each other with small latency
- Agent servers would achieve a reasonable approximation to represent those web servers*
Aggregative Network Condition Measurement

- Deploying an agent server to aggregate the network conditions by representing the group of nearby web services in the same region.

Servers within this zone can reach each other with small latency.
Agent servers would achieve a reasonable approximation to represent those web servers.*

Challenge II: How to update the classifier with ensuring the timeliness and generality?
Local OR Global?

- **High timeliness**
  - Easy to overfit and losing robustness
  - Local updating on the client-side

- **Generality and robustness**
  - Hard to match users’ specific characteristics
  - Global updating on the agent servers

**High Timeliness**
- Generality and Robustness
- Local updating + global information
Hybrid Global-Local Update Mechanism

1. Provide global info.
2. Enrich the global info.

*Global classifier is publicly accessible.
**Global trace pool is maintained in a distributed way, since agent servers are deployed worldwide.
HTTP Version Selection

- Making HTTP version selections before browsing
- Adding an additional cache
  - If a request does not hit the cache, FlexHTTP will randomly select a version (H2 or H3) and update the cache with local classifier
1. FlexHTTP **achieves a smaller Speed Index** in terms of either the mean or median value.

2. FlexHTTP can always **capture the appropriate HTTP version** in almost every experiment configuration.

Speed Index is a metric that indicates how quickly a page is loaded and visibly rendered.
1. Both network conditions and web page structures affect performance of H2 and H3
2. Agent servers and hybrid global-local updating ensure the scalability and information timeliness
3. Evaluation demonstrates the FlexHTTP’s capability of improving web browsing

Thanks for your listening!

The source code of prototype FlexHTTP could be accessed via https://github.com/mengyingzhou/flexhttp.