2nd European Workshop on Machine Learning and Systems (EuroMLSys'22), co-located with EuroSys'22

FlexHTTP:

An Intelligent and Scalable HTTP Version Selection System

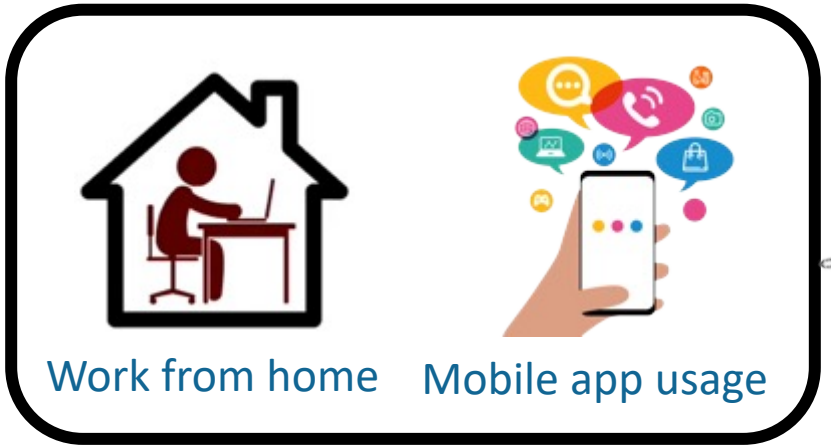
Mengying Zhou, Zheng Li, Shihan Lin, Xin Wang, Yang Chen

School of Computer Science

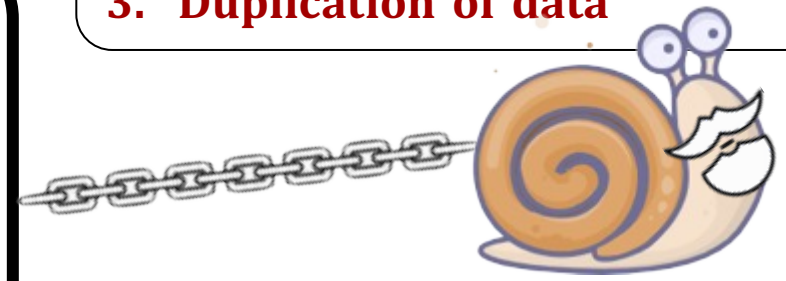
Fudan University

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**



HTTP/1.1

Hardly to satisfy these demands

Two Separate Improvement Approaches

Approach 1: TCP-based

- BBR congestion algorithm
- TLS 1.2
- SPDY

HTTP/2 [1]

Request multiplexing

Sever push

Stream prioritization

Approach 2: UDP-based

- On the top of UDP
- A reliable protocol
- QUIC

HTTP-over-QUIC

HTTP/3 [2]

Faster connection establishment

Removal of TCP's Head-of-Line problem

Improved recovery mechanism

[1] Mike Belshe, Roberto Peon, and Martin Thomson. 2015. Hypertext Transfer Protocol Version 2 (HTTP/2).
[2] Mike Bishop. 2021. Hypertext Transfer Protocol Version 3 (HTTP/3).

Evidence Tells Us: Neither of Them is Always Better

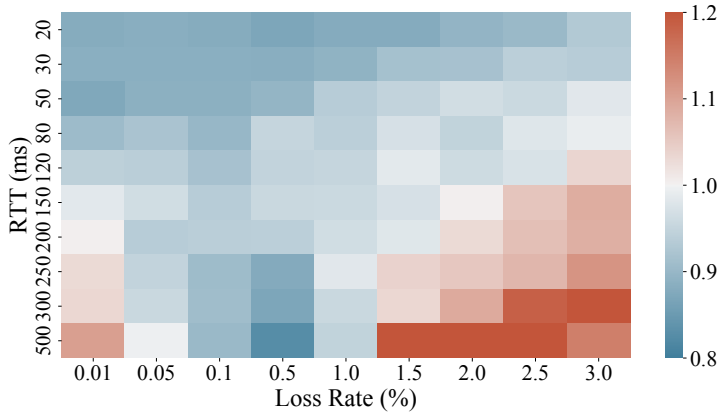
Network condition: low bandwidth, high latency, high packet loss

HTTP/3 👍 HTTP/2 🙅

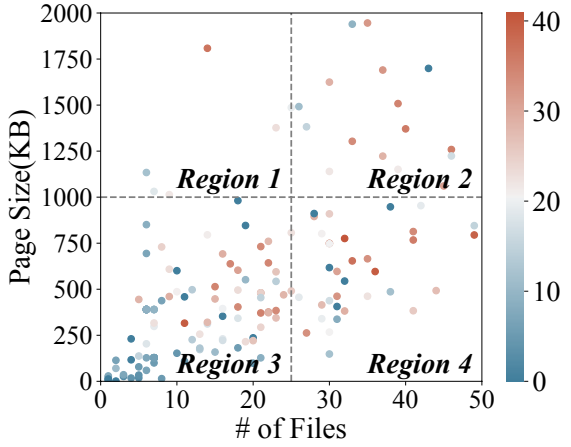
Web page structure: few small files

HTTP/3 🙅 HTTP/2 👍

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



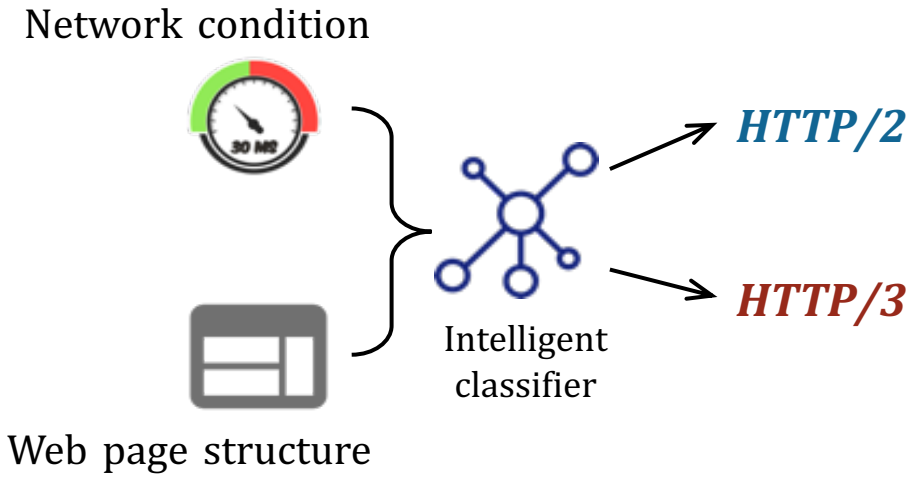
Performance differences between H2 and H3 in different network conditions



Performance differences between H2 and H3 in different web page structures

FlexHTTP:

An Intelligent and Scalable HTTP Version Selection System

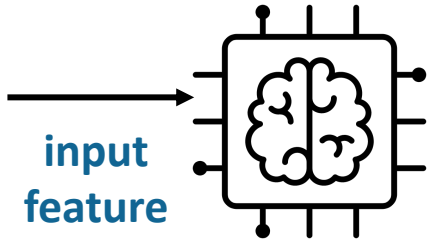


*Towards a **flexible** HTTP version selection according to **network conditions** and **page structures***

Equipping a supervised machine learning-based classifier in the browser as a plug-in

Challenge I: Network Condition Measurement Overhead

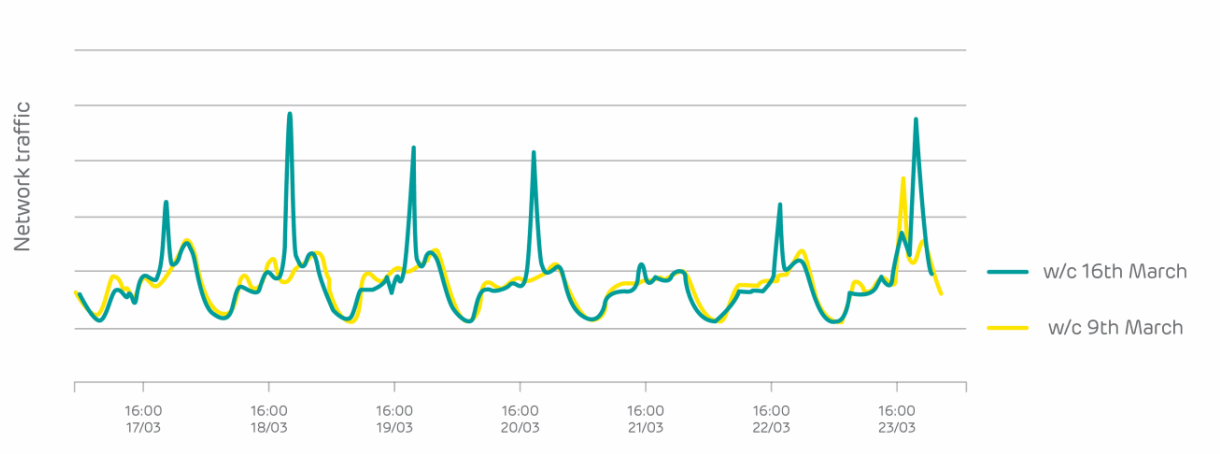
- Classifier needs the information on network condition and web page structure as the input
- Network condition feature is link-based
- The possible number of client-server pairs and corresponding features would be huge

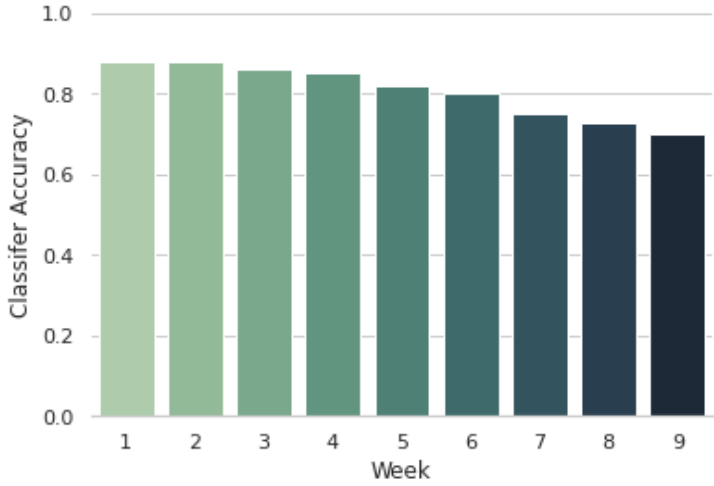


One link, one network condition

Challenge II: Classifier Updating

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



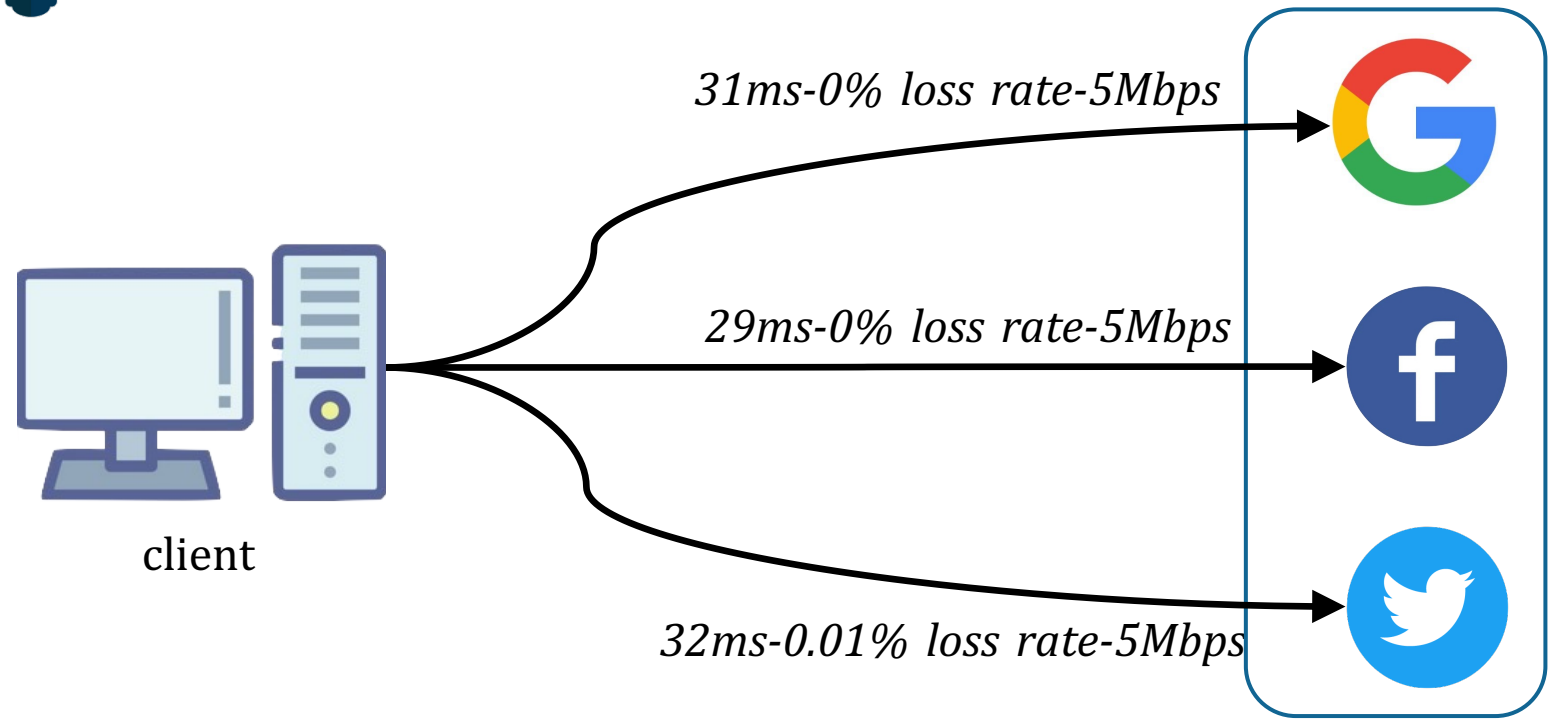


How to update classifier with ensuring both timeliness and generality?

Challenge Ⅰ:
How to reduce the measurement traffic to a moderate level

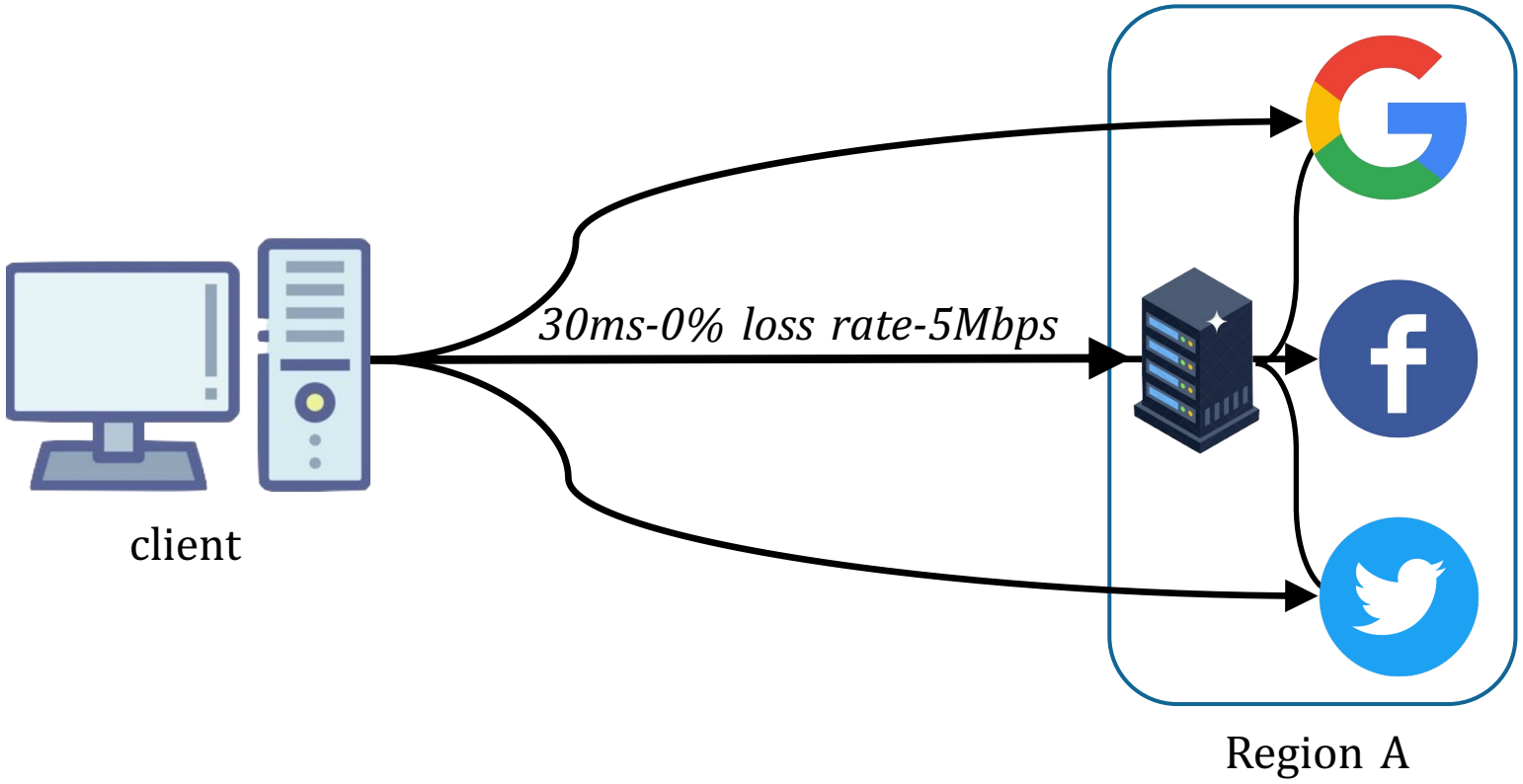
Similar Network Condition within A Region

Could we use a representative agent to aggregate network conditions?



Aggregative Network Condition Measurement

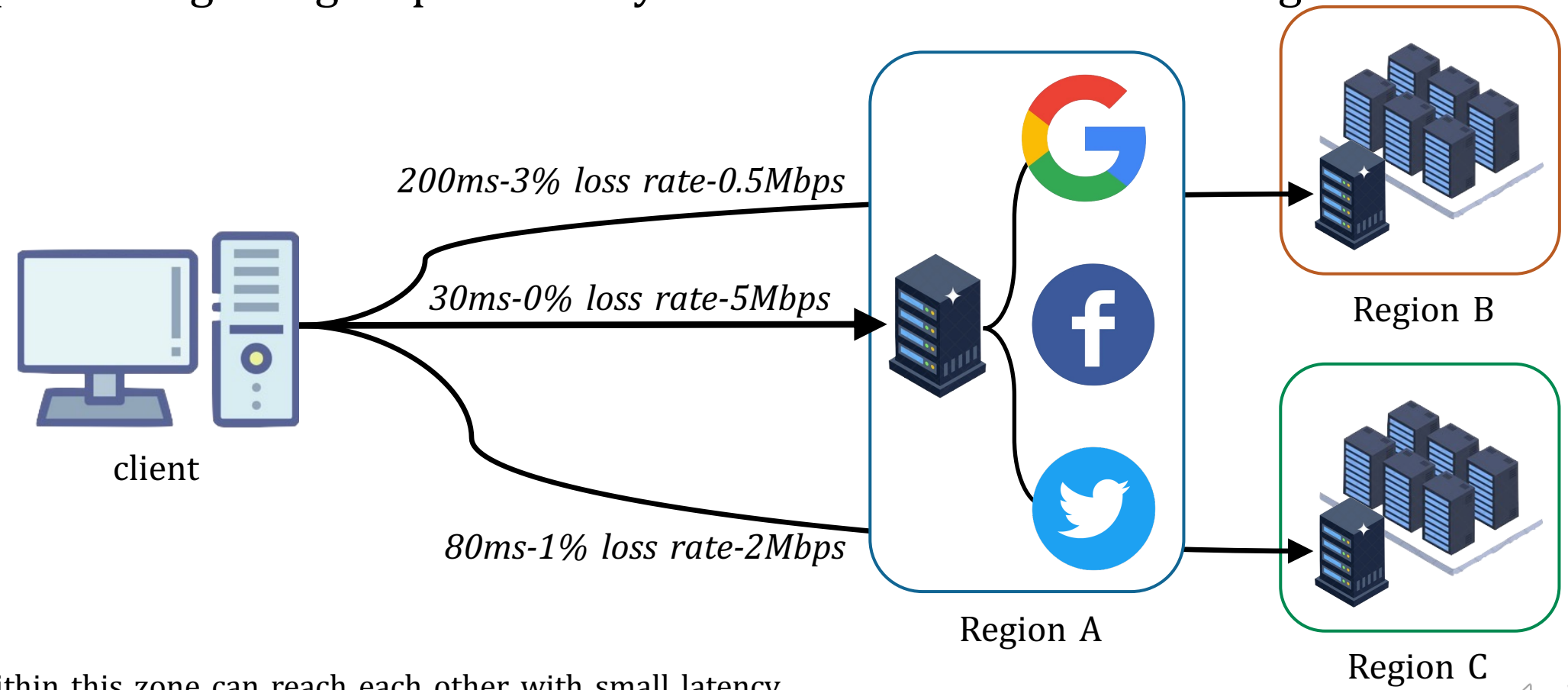
- Deploying an agent server to aggregate the network conditions by representing the group of nearby web services in the same region



- 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*

*Hu et al. 2015. GeoTopo: A PoP-level Topology Generator for Evaluation of Future Internet Architectures. In Proc. of ICNP.

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

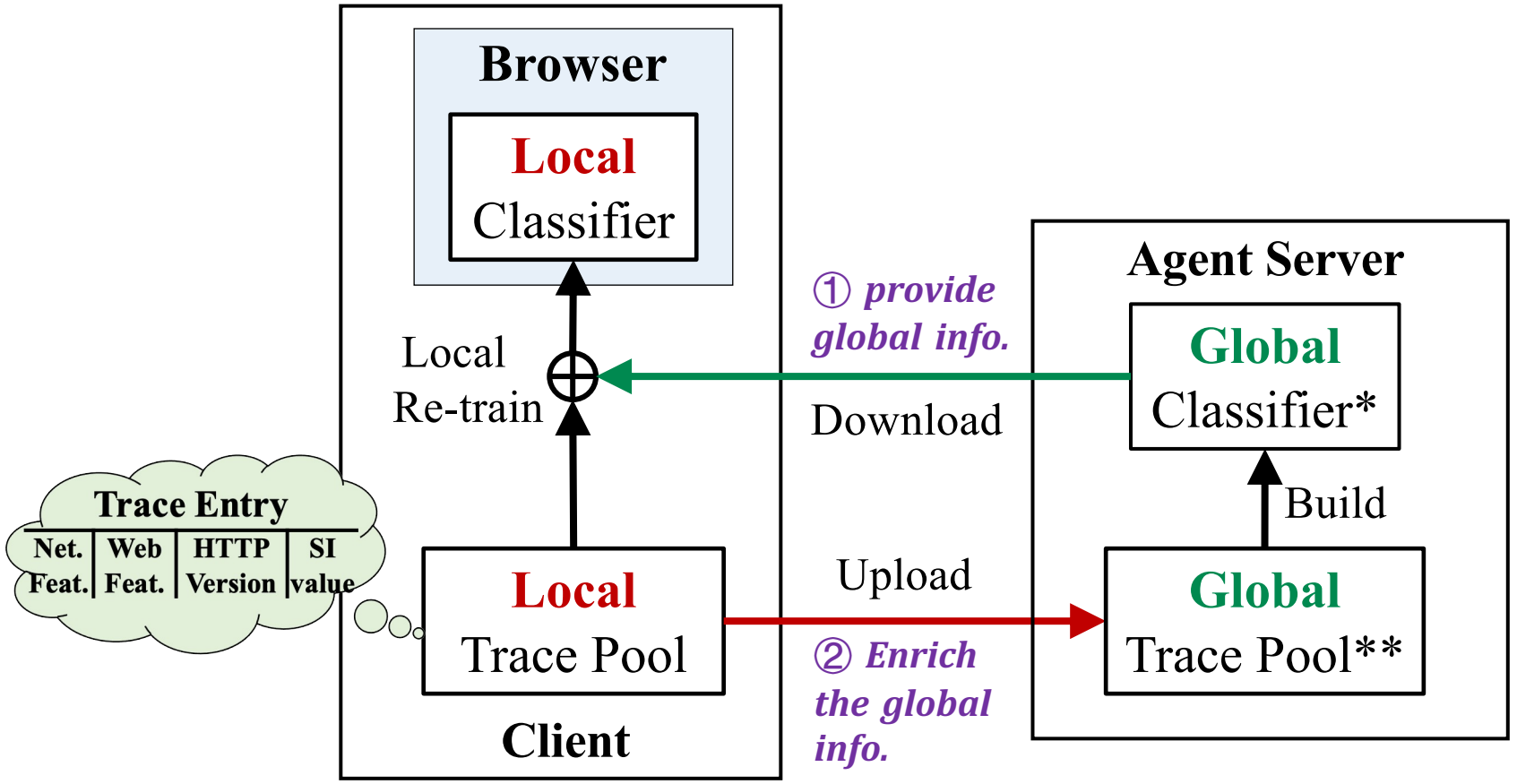
Global updating on the agent servers

High Timeliness

Generality and Robustness

Local updating + global information

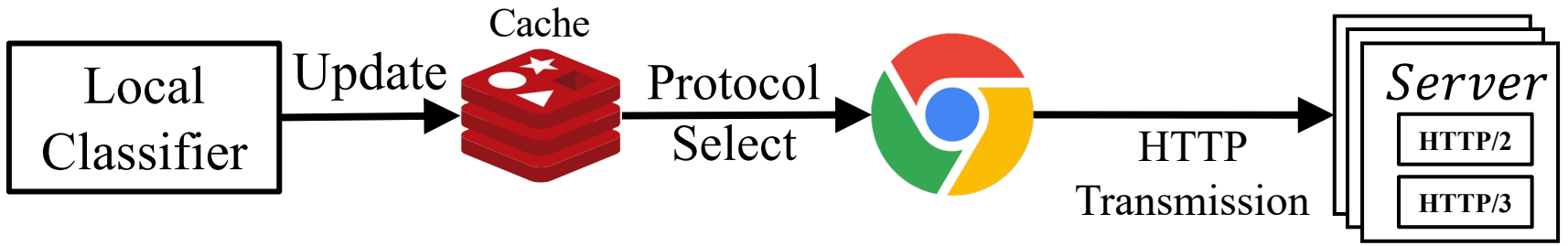
Hybrid Global-Local Update Mechanism



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

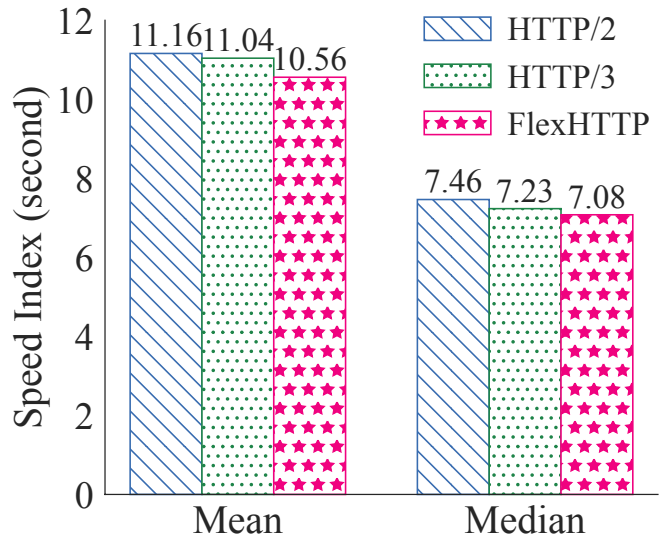
HTTP Version Selection

- 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

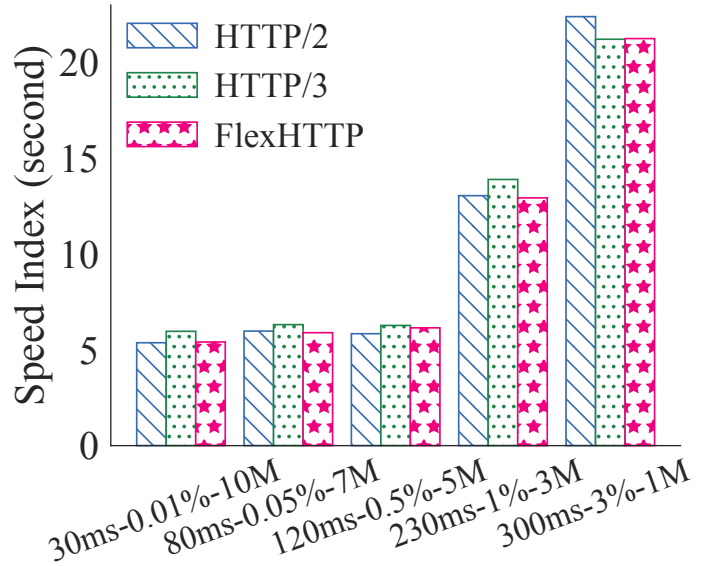


Less Speed Index Time

Comparison of Speed Index* time performance among HTTP/2, HTTP/3, and FlexHTTP



(a) The mean and median values of Speed Index



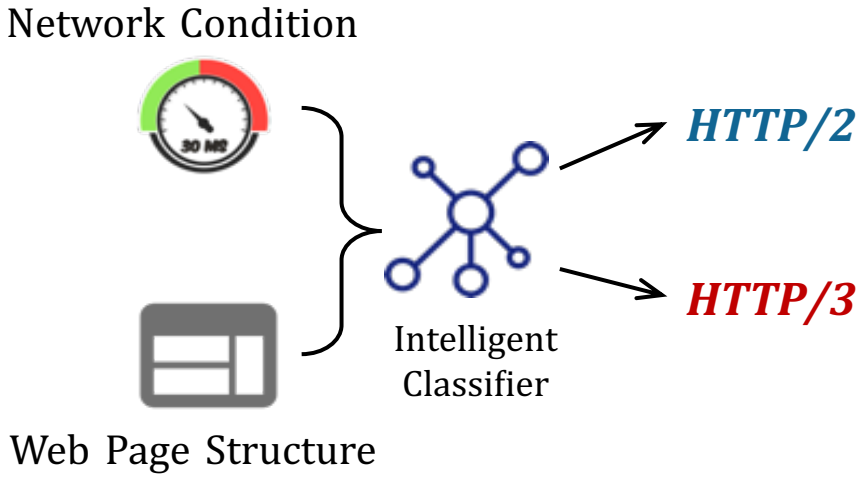
(b) Median values of Speed Index of each configuration

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.

FlexHTTP:

An Intelligent and Scalable HTTP Version Selection System



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.