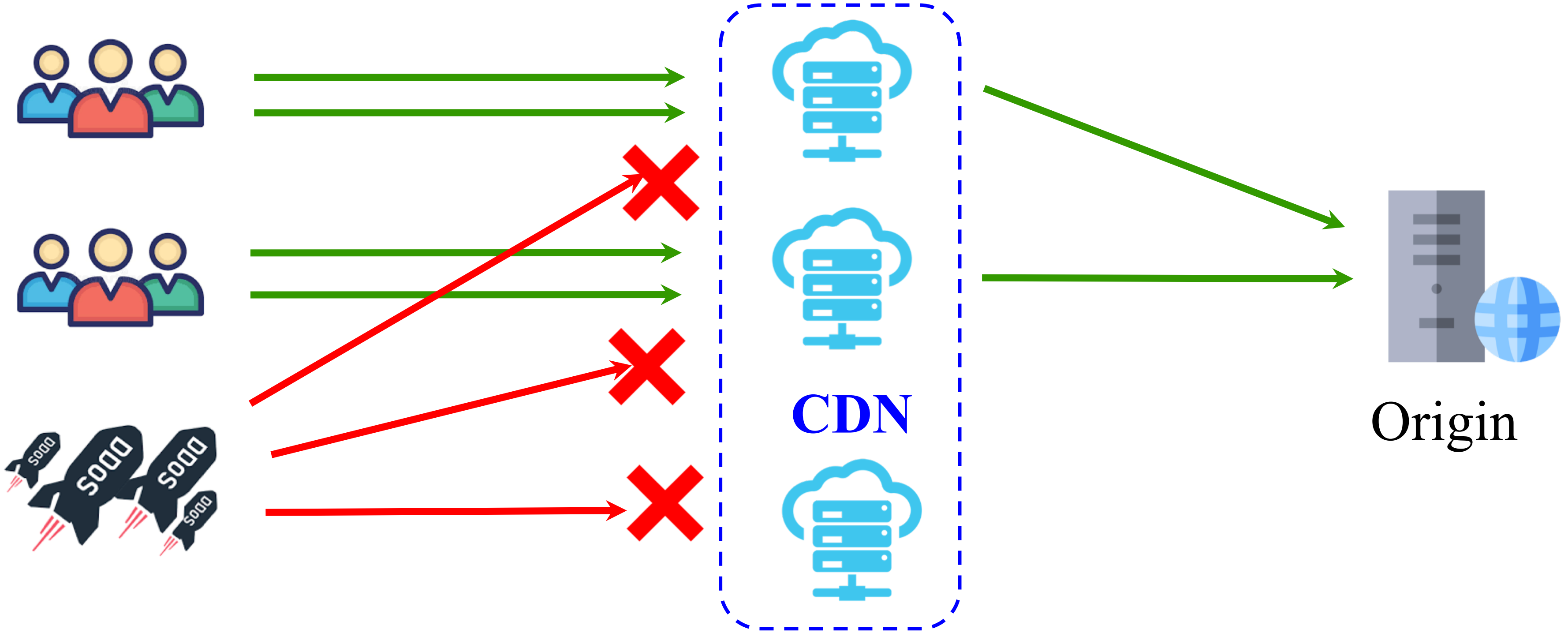
# CDN Judo : Breaking the CDN DoS Protection with Itself

Run Guo, Weizhong Li, Baojun Liu, Shuang Hao,
Jia Zhang, Haixin Duan, Kaiwen Shen**, Jianjun Chen,** Ying Liu

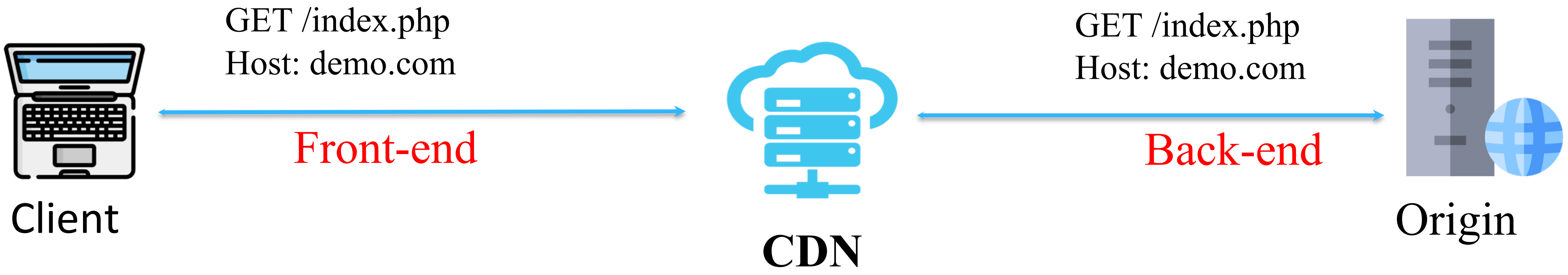
# Content Delivery Network

- Infrastructure for access acceleration and DoS defense
  - 38.98% of top 10K websites use CDN [Your Remnant Tells Secret-DSN’18]
  - We find CDN itself can be abuse to break its DoS protection



# CDN Forwarding Process

**End-to-end connection** ⇒ **Front-end and back-end connections**



```
GET /index.php
Host: demo.com
```

```
GET /index.php
Host: demo.com
```

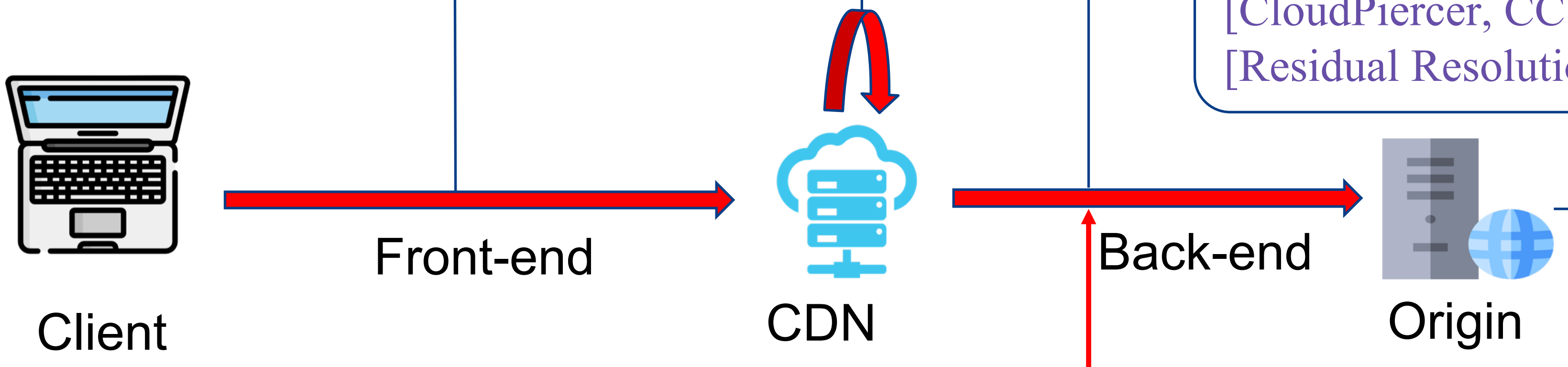
# Previous Works

**Front-end connection security**
[HTTPS meet CDN, IEEE S&P ’14]
[TLS private key sharing, CCS ’16]
[Host of troubles, CCS ’16]
[Cache fallen, CCS ’19]
[End user maneuvered, USENIX security ’18]
[Cached and Confused, USENIX security ’20]

**CDN internal security**
[Forwarding loop attack, NDSS ’16]

**Back-end connection security**
[Protection or Threat, ESORICS ’09]

**Origin IP exposure**
[CloudPiercer, CCS ’15]
[Residual Resolution, DSN ’18]



Our work: abuse CDN-forwarded requests to attack the origin.

# Our Work

- Exploiting CDN forwarding features to attack the origin

| | |
|---|---|
| Attack-1 | HTTP/2 amplification attack |
| Attack-2 | Pre-POST slow HTTP attack |
| Attack-3 | Egress IP blocking attack |

- Performed real-world evaluations on six vendors

amazon web services

CLOUDFLARE

CDNsun

fastly

keycdn

maxcdn

Attack-1

# HTTP/2 Amplification Attack

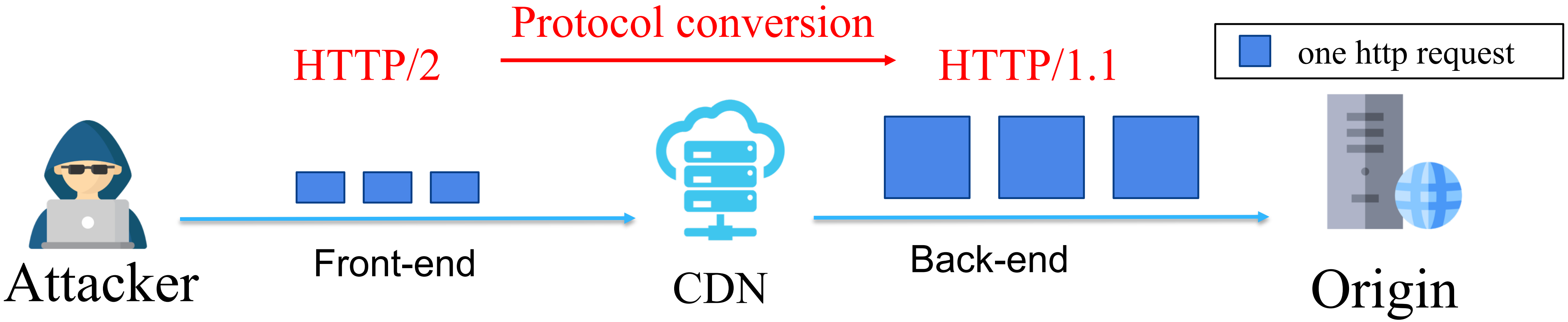
# HTTP/2 Protocol

- Designed to improve HTTP performance
  - RFC7540, released in 2015
- Compression (to reduce header redundancy)
  - Binary protocol, HPACK header compression
- Connection reuse (to reduce TCP connections)
  - Request -> Stream
  - Streams are multiplexed

❑ *Deployment: Over 43.2% of Alexa top 1M websites (w3techs.com, 12 Feb 2020)*

# Concept of HTTP/2 Amplification attack

- Our study
  - Identify that HTTP/2-1.1 conversion of CDN will cause amplification attack.
  - Improve the attack with the feature of Huffman encoding.
  - Real-world measurement and evaluation



- [HTTP/2 Tsunami Attack, EST ’17]

  Show bandwidth amplification attack in local proxies built with Nginx and Nghttp2.

# CDN Vendors Claim to Support HTTP/2

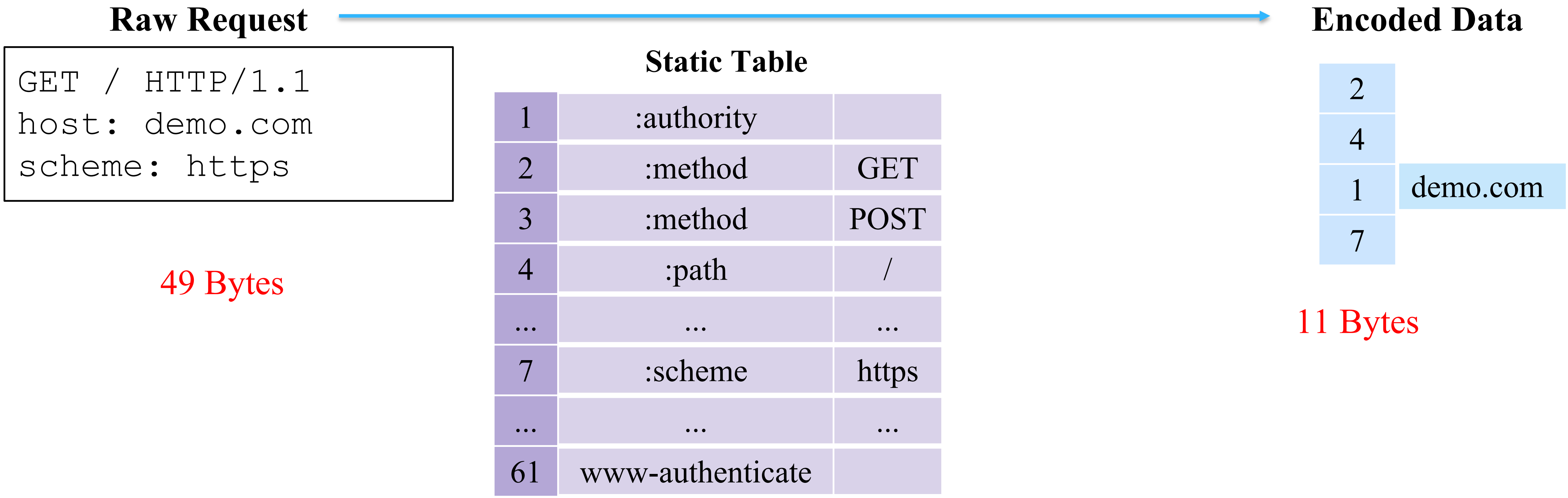
- HTTP/2 is supported by most major CDNs
- The backend connection still uses HTTP/1.1

| | CloudFront | Cloudflare | CDNSun | Fastly | KeyCDN | MaxCDN |
|---|---|---|---|---|---|---|
| **Frontend Connection** | Default on<br>Configurable | Default on | Default on | Default off<br>Configurable | Default on | Default on<br>Configurable |
| **Backend Connection** | Only support HTTP/1.1 | | | | | |

Next we describe three amplification attack techniques.

# HPACK Static Table

- An indexed table of common header fields
- pre-defined in both HTTP/2 client and server.



**Raw Request**

```
GET / HTTP/1.1
host: demo.com
scheme: https
```

49 Bytes

**Static Table**

| | | |
|---|---|---|
| 1 | :authority | |
| 2 | :method | GET |
| 3 | :method | POST |
| 4 | :path | / |
| ... | ... | ... |
| 7 | :scheme | https |
| ... | ... | ... |
| 61 | www-authenticate | |

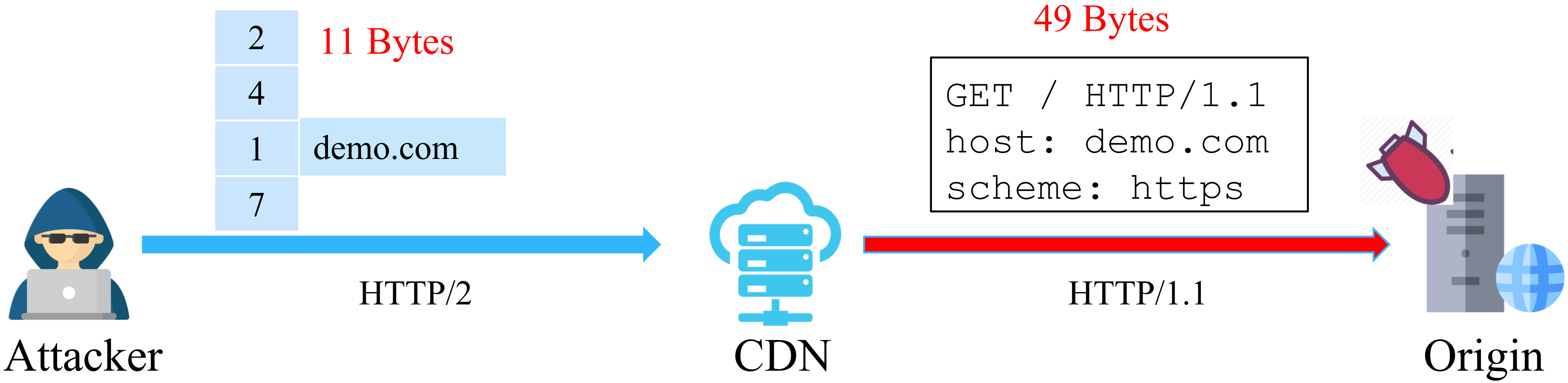
**Encoded Data**

| | |
|---|---|
| 2 | |
| 4 | |
| 1 | demo.com |
| 7 | |

11 Bytes

# Attack-1.1: Using HPACK Static Table

- HTTP/2-1.1 conversion of CDN causes a bandwidth amplification.



11 Bytes

| | |
|---|---|
| 2 | |
| 4 | |
| 1 | demo.com |
| 7 | |

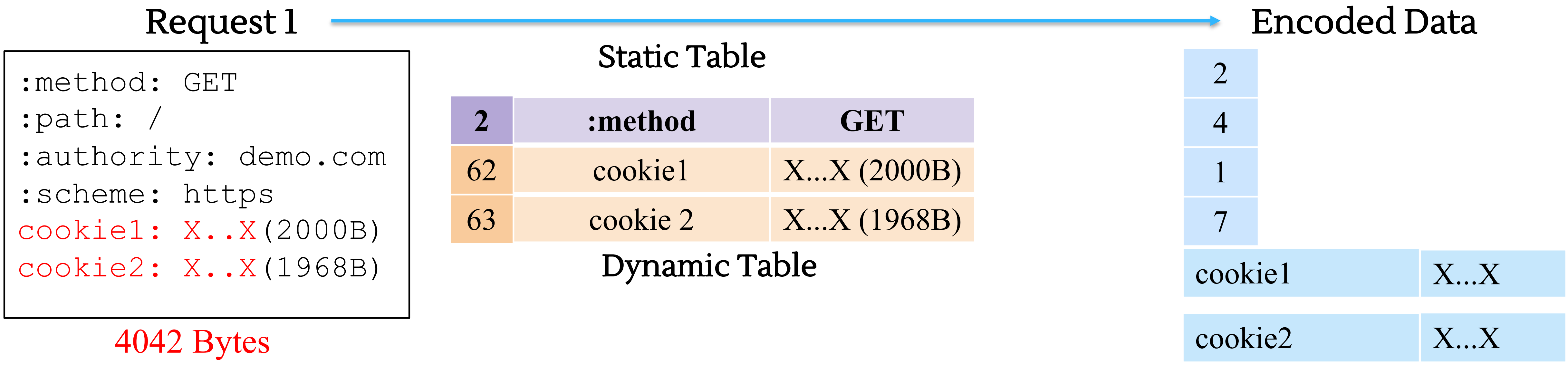
49 Bytes

```
GET / HTTP/1.1
host: demo.com
scheme: https
```

Bandwidth amplification factor: 49B / 11B = 4.45

# HPACK Dynamic Table (1/2)

- An indexed table of previously seen headers to avoid repeatedly transferring headers.
  - Step 1: The firstly seen headers will be inserted into the dynamic table.



Request 1

```
:method: GET
:path: /
:authority: demo.com
:scheme: https
cookie1: X..X(2000B)
cookie2: X..X(1968B)
```

4042 Bytes

Static Table

| 2 | :method | GET |
|---|---|---|
| 62 | cookie1 | X...X (2000B) |
| 63 | cookie 2 | X...X (1968B) |

Dynamic Table

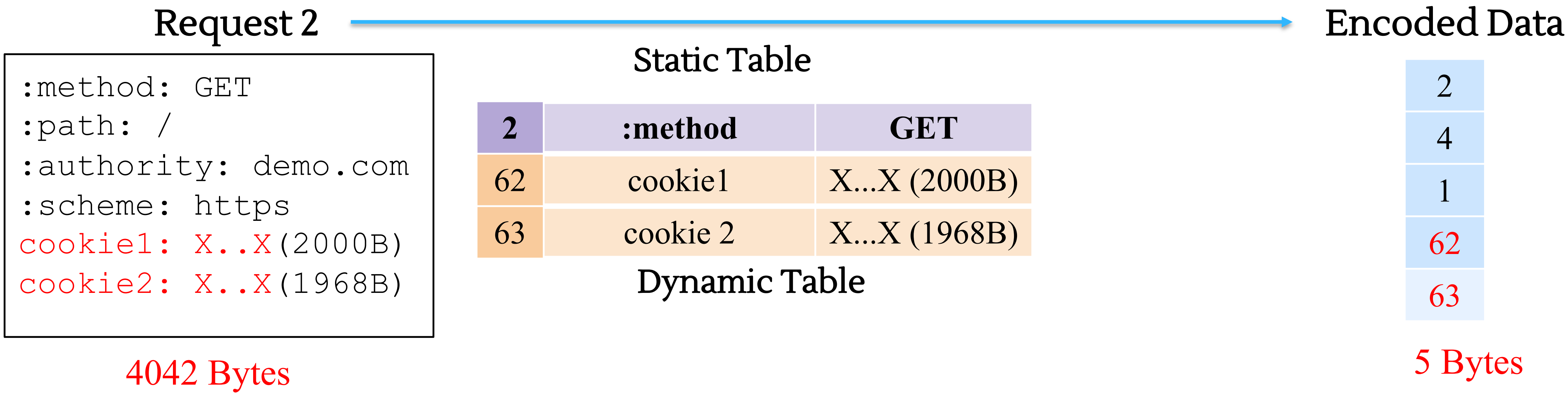
Encoded Data

| | |
|---|---|
| 2 | |
| 4 | |
| 1 | |
| 7 | |
| cookie1 | X...X |
| cookie2 | X...X |

3999 Bytes

# HPACK Dynamic Table (2/2)

- An indexed table of previously seen headers to avoid repeatedly transferring headers.
  - Step 2: The subsequently repeated headers will be substituted as an index.



Request 2

```
:method: GET
:path: /
:authority: demo.com
:scheme: https
cookie1: X..X(2000B)
cookie2: X..X(1968B)
```

4042 Bytes

Static Table

| | | |
|---|---|---|
| **2** | **:method** | **GET** |

| | | |
|---|---|---|
| 62 | cookie1 | X...X (2000B) |
| 63 | cookie 2 | X...X (1968B) |

Dynamic Table

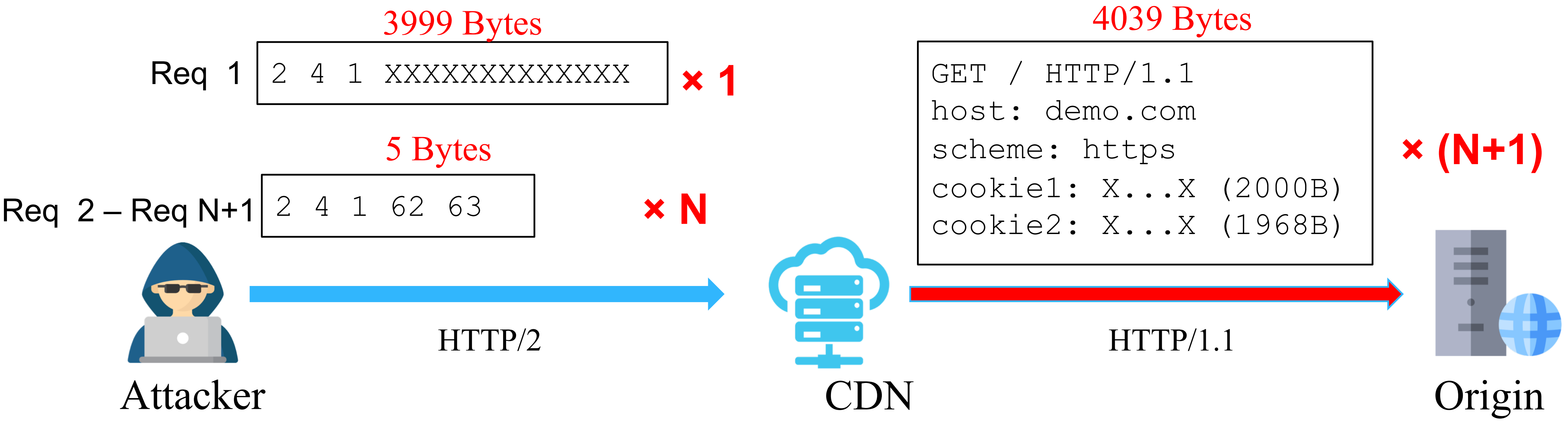
Encoded Data

2<br>
4<br>
1<br>
62<br>
63

5 Bytes

# Attack-1.2: Using HPACK Dynamic Table

- The dynamic table enhances this kind of bandwidth amplification.



3999 Bytes

Req 1

```
2 4 1 XXXXXXXXXXXXX
```

× 1

5 Bytes

Req 2 – Req N+1

```
2 4 1 62 63
```

× N

4039 Bytes

```
GET / HTTP/1.1
host: demo.com
scheme: https
cookie1: X...X (2000B)
cookie2: X...X (1968B)
```

× (N+1)

Attacker

HTTP/2

CDN

HTTP/1.1

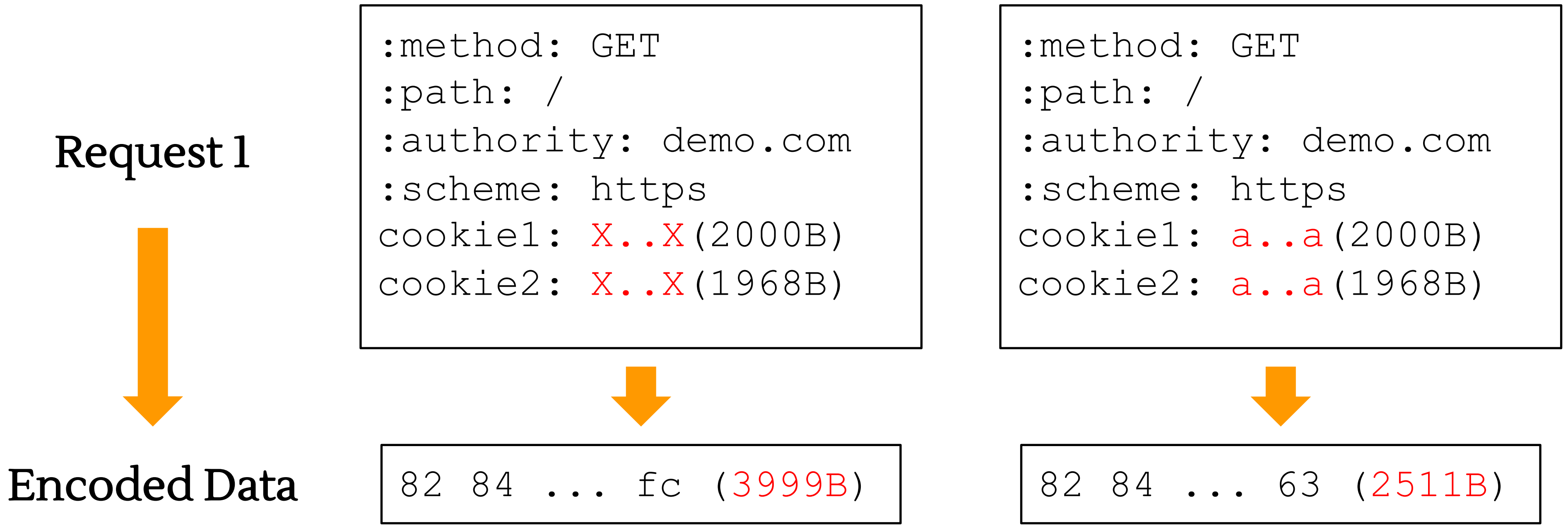
Origin

Bandwidth amplification factor: 4039B × **(N+1)** / 3999B + 5B × **N** = $\frac{4039 + 4039N}{3999 + 5N}$

For example, when N is 100, the factor is 88.70.

# Attack-1.3: Improve with Huffman Encoding

- Some special characters can have short Huffman encodings
  - ➢The Huffman encoding of ‘X’ is 8 bits in length.
  - ➢Characters {0, 1, 2, a, c, e, i, o, s, t} have the shortest Huffman encoding (5 bits).



Request 1

```
:method: GET
:path: /
:authority: demo.com
:scheme: https
cookie1: X..X(2000B)
cookie2: X..X(1968B)
```

```
:method: GET
:path: /
:authority: demo.com
:scheme: https
cookie1: a..a(2000B)
cookie2: a..a(1968B)
```

Encoded Data

```
82 84 ... fc (3999B)
```

```
82 84 ... 63 (2511B)
```

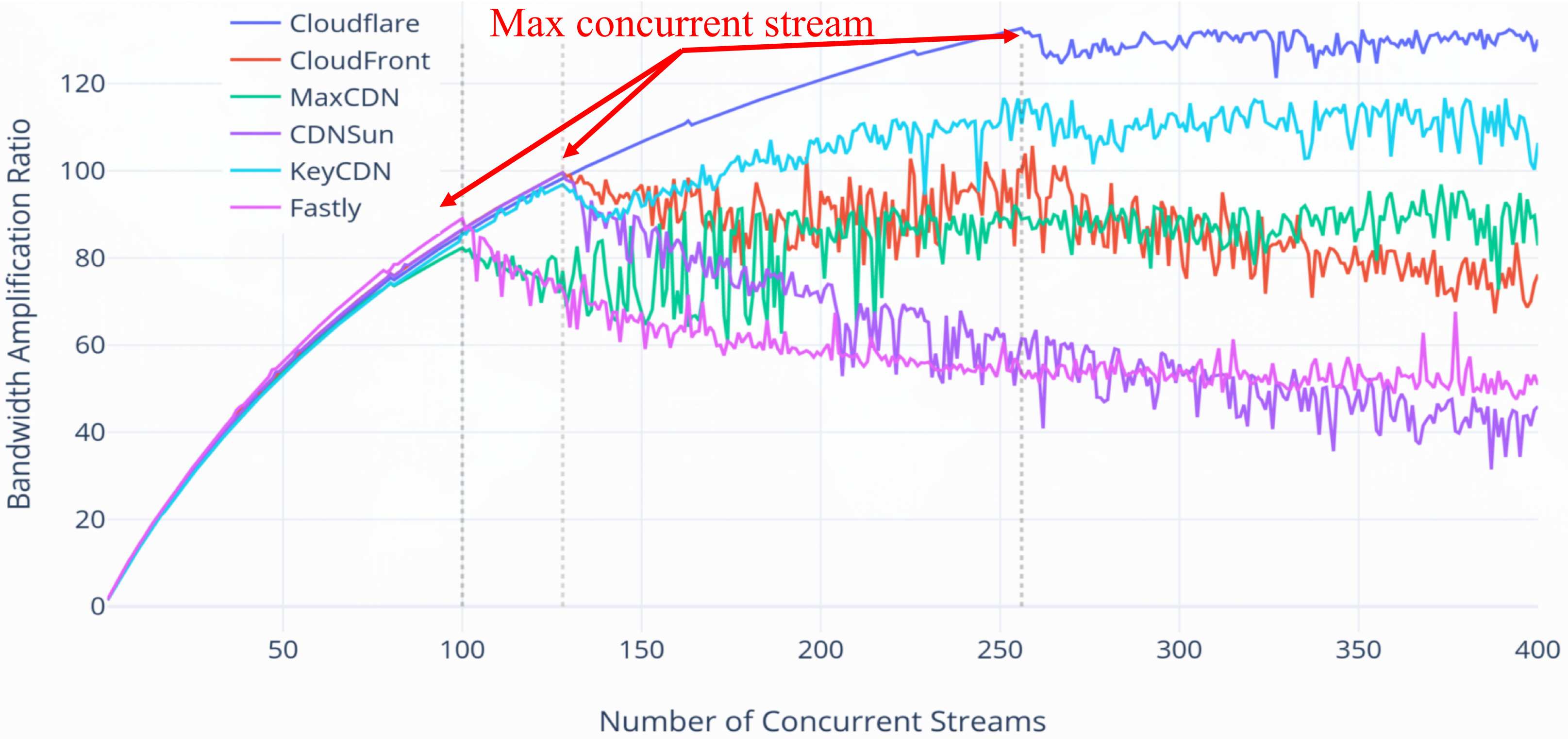
# Attack-1.3: Improve with Huffman Encoding

- The shorter the Huffman encoding, the larger the amplification factor.

| | **Huffman Encoding Length** | **Amplification Factor** | |
|---|---|---|---|
| **Character ‘X’** | 8 bits | $\frac{4039 + 4039\mathbf{N}}{3999 + 5N}$ | 88.70<br>when N is 100 |
| **Character ‘a’** | 5 bits | $\frac{4039 + 4039\mathbf{N}}{2511 + 5N}$ | 131.13<br>when N is 100 |
| Note: N is the concurrent streams in the same HTTP/2 connection. | | | |

# Bandwidth Amplification Evaluation

- Create multiple concurrent requests in one HTTP/2 connection.
  - ➢The amplification factor grows with the number of concurrent streams.
  - ➢The max factor is got at the position of the max concurrent streams.



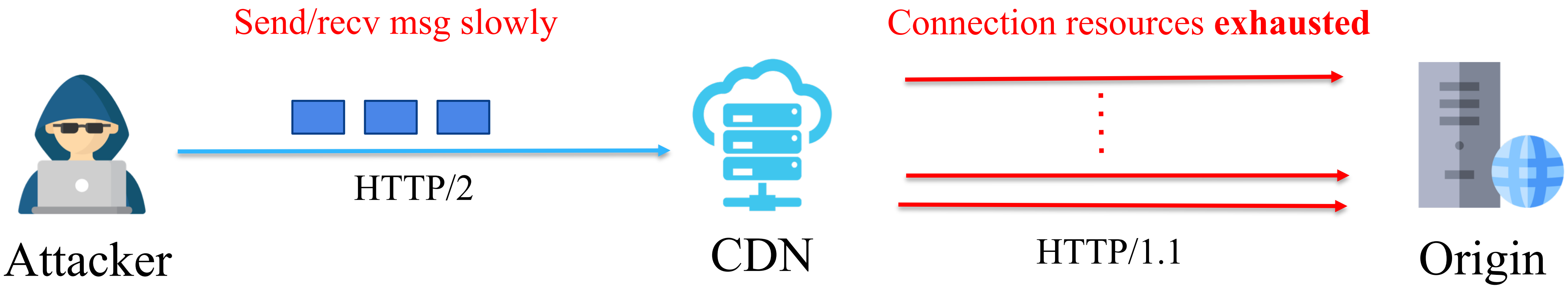
# Comparison with previous work

- Our work achieved larger amplification factors than previous work.

| | **Max Streams** | **100** | | **128** | | | **256** |
|---|---|---|---|---|---|---|---|
| **Our Attack** | Evaluation Platform | MaxCDN | Fastly | CDNsun | CloudFront | KeyCDN | Cloudflare |
| | Amplification Factor | 94.7 | 97.9 | 118.7 | 116.9 | 105.5 | 166.1 |
| **HTTP/2 Tsunami Attack** | Evaluation Platform | HTTP/2 Proxies built with Nginx and Nghttp2 | | | | | |
| | Amplification Factor | 79.2 | | 94.4 | | | 140.6 |

# HTTP/2 Connection Amplification Attack

- concurrent streams in one HTTP/2 connection → multiple HTTP/1.1 connections



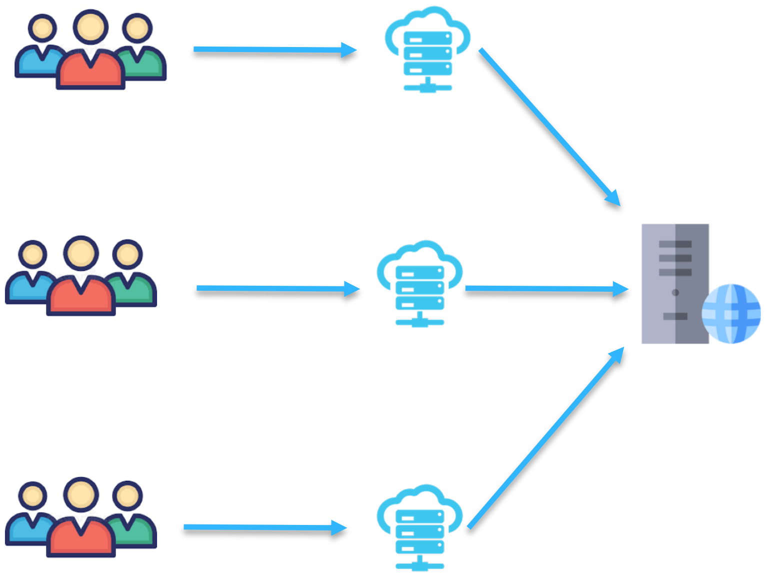
| | CloudFront | Cloudflare | CDNSun | Fastly | KeyCDN | MaxCDN |
|---|---|---|---|---|---|---|
| **Max concurrent streams per HTTP/2 connection** | 128 | 256 | 128 | 100 | 128 | 100 |
| **Connection Amplification** | Yes | Yes | - | - | - | Yes |

Attack-3

# Egress IP Blocking Attack

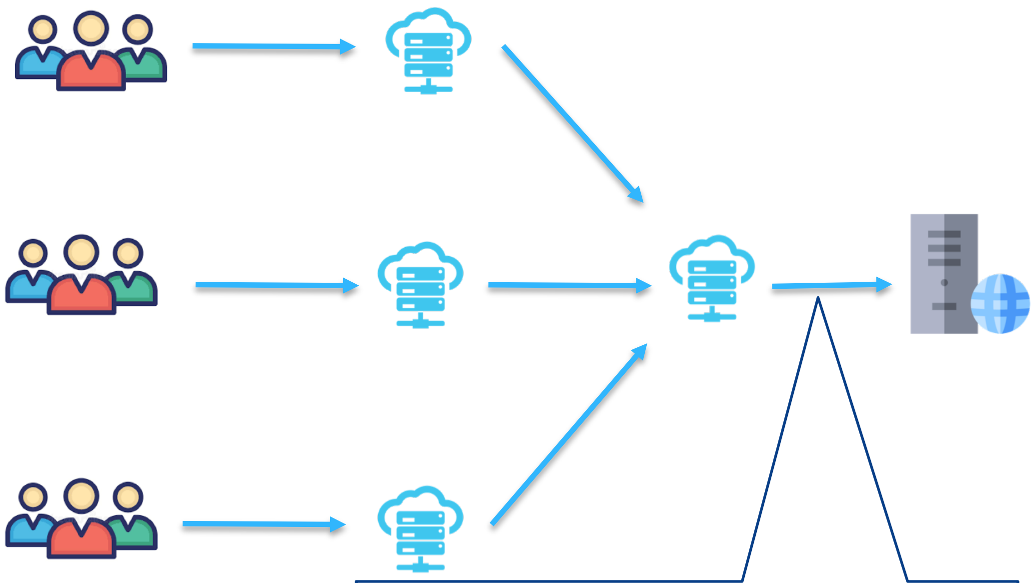
# Origin Shield

## Without Origin Shield



## With Origin Shield

- reduce origin workload
- speed up cache-miss responses

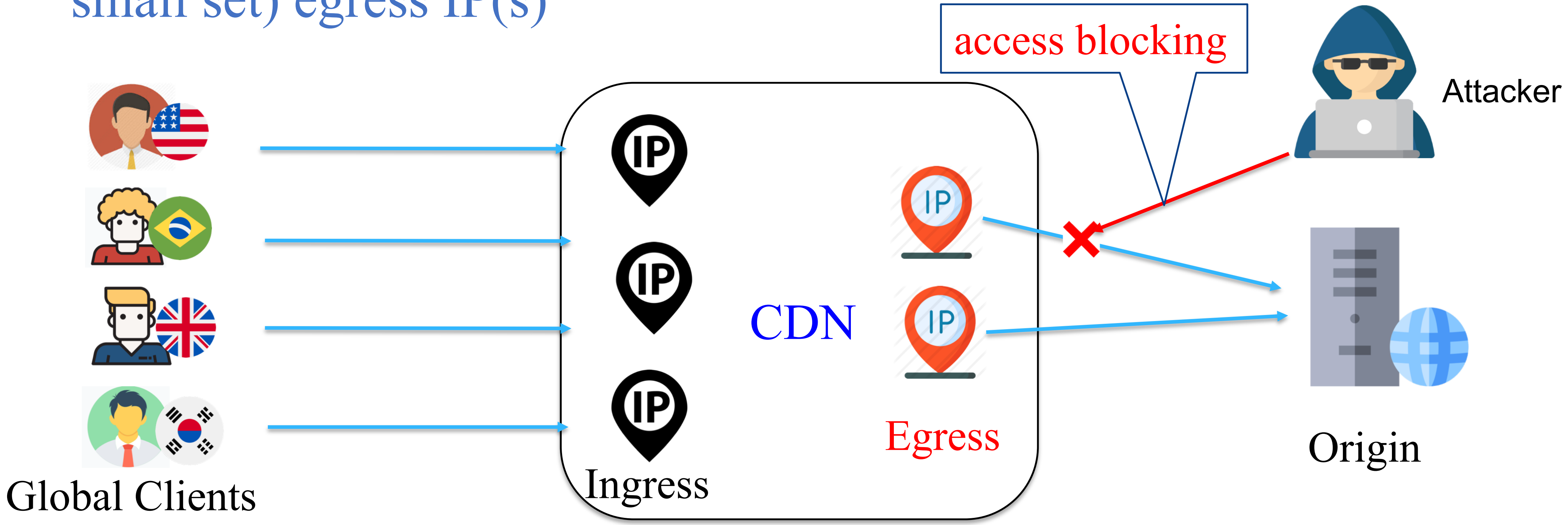


backend connections originated from less egress IPs.

❑ *https://docs.fastly.com/en/guides/shielding*

# Threat Model

- Global clients will be affected when an attacker just block one (or a small set) egress IP(s)



Next we describe our measurement of CDN IP distribution, and evaluation experiments.

# Characteristics of Egress IP distribution

- Observation 1: Fewer egress IPs than ingress IPs

| | Ingress IPs | Egress IPs | Egress/Ingress |
|---|---|---|---|
| CloudFront | 128,906 | 862 | 0.67% |
| Cloudflare | 490,309 | 242 | 0.05% |
| Fastly | 64,659 | 1,136 | 1.7% |
| MaxCDN | 300 | 12 | 4% |

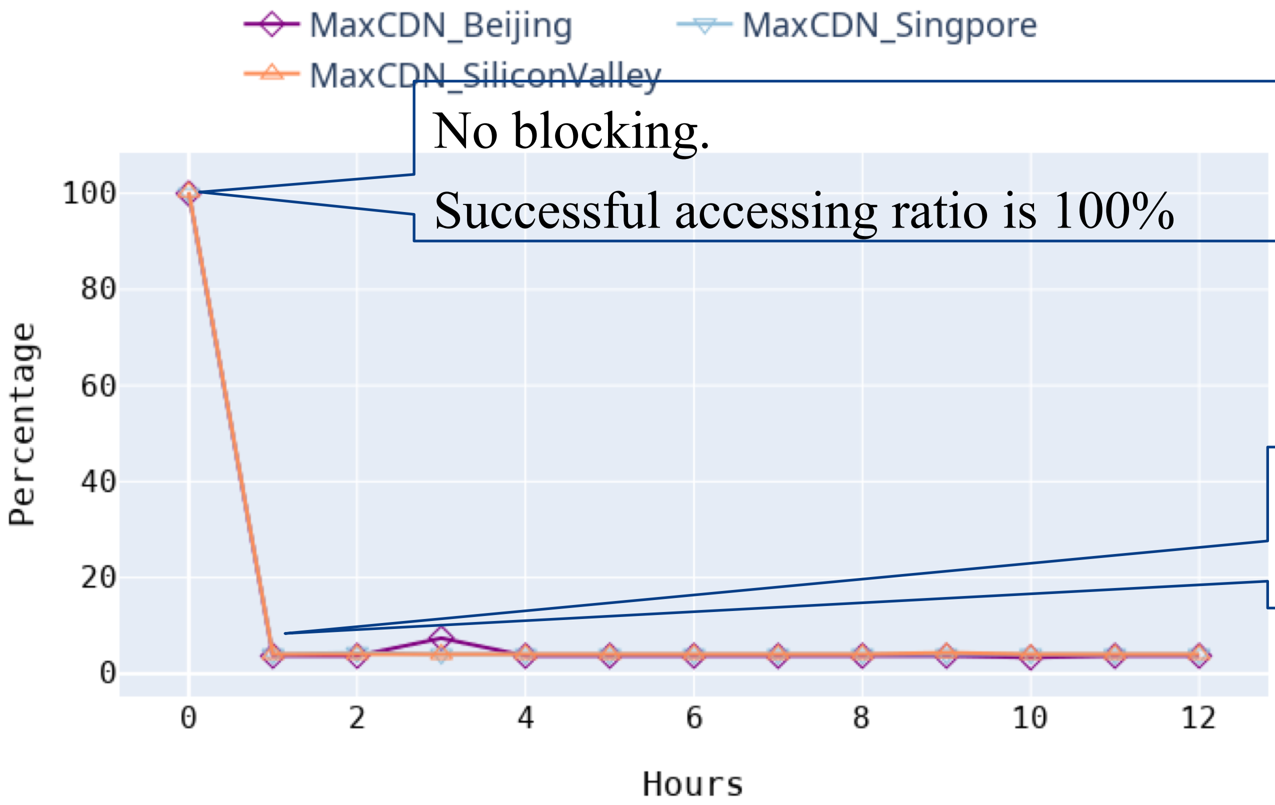
- Observation 2: Churning rate of egress IPs are low
  - MaxCDN: 96.32% of the backend connections originated from the same egress IP.
  - Other CDNs churn egress IPs more fast, < 10% of the backend connections originated form the same egress IP.

❑ *Results are consistent with [Unveil the hidden presence, ICNP ’19]*

# Egress IP Blocking Evaluation

MaxCDN

- We block one single egress IP at our origin for 12 hours
- Access the website from global ingress IPs



No blocking.

Successful accessing ratio is 100%

Block one egress IP.

Successful accessing ratio drops below 10%.

# Real-world Case Study

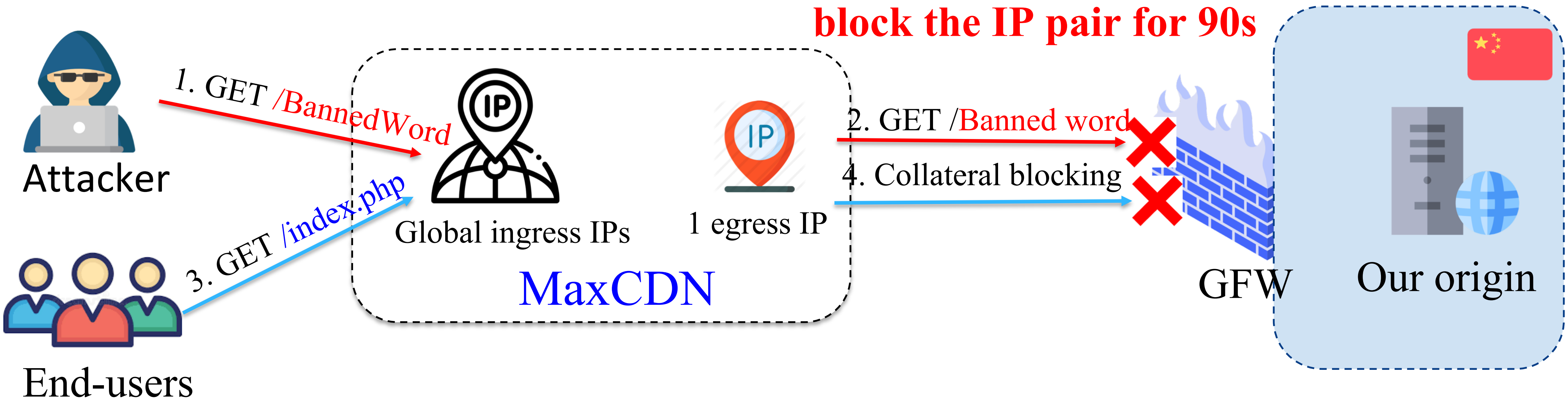
**Censorship (e.g., Great Firewall of China)**

- locate between CDN and origin
- inspect censored bad words
- block the IP pair for 90s

⇨

**Collateral blocking**

- Attacker sends requests to ingress IPs
- Global end-users are collaterally blocked



# Mitigation

| Threats | Recommendation |
|---|---|
| HTTP/2 attack | HTTP/2 support for back-end connection<br>limit the back-end network traffic. |
| Pre-POST attack | limit the number of CDN back-to-origin connections<br>enforce strict forwarding (store-then-forward). |
| Egress IP blocking | apply unpredictable egress IP churning strategy. |

# Responsible Disclosure

- **Cloudflare:** reproduced HTTP/2 amplification with 126x and rewarded us $200 bonus.
- **Fastly:** confirmed our report and offered us T-shirts.
- **CloudFront:** suggested HTTP/2 amplification is a feature of HTTP/2 standard, and would like to use rate-based WAF rules to mitigate the attack.
- **MaxCDN:** stated the egress IP blocking is out of scope as it involves with additional GFW infrastructure.
- **CDNSun** and **KeyCDN:** received our report but no further comments so far.

# Summary

- A empirical security study on CDN back-end connections
  - HTTP/2 amplification attack
  - pre-POST slow HTTP attack
  - Egress IP blocking attack
- Real-world evaluation on six CDN vendors
  - Received positive feedback from some CDNs
- How to balance performance and security

# Thank you!