# Network support for TCP Fast Open

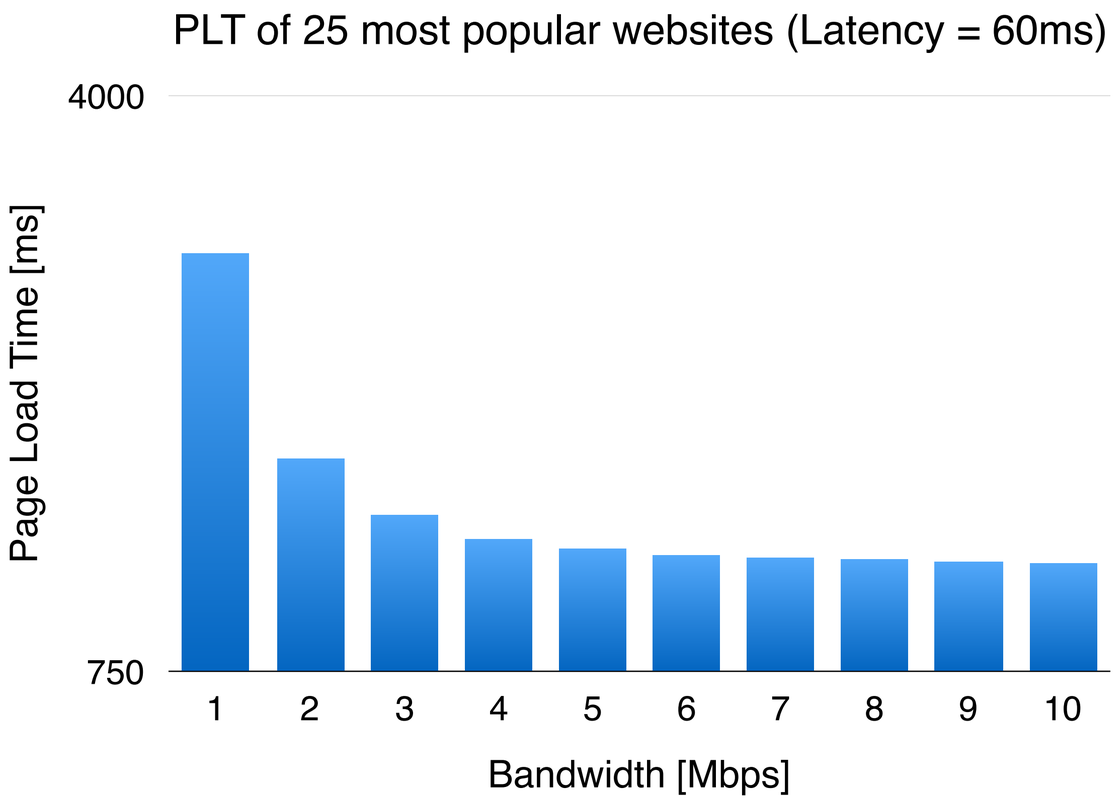
Christoph Paasch <cpaasch@apple.com>

# Outline

- TCP Fast Open allows to reduce latency and significantly improve user-experience
- However, naive firewalls and bad Intrusion Detection Systems got in our way

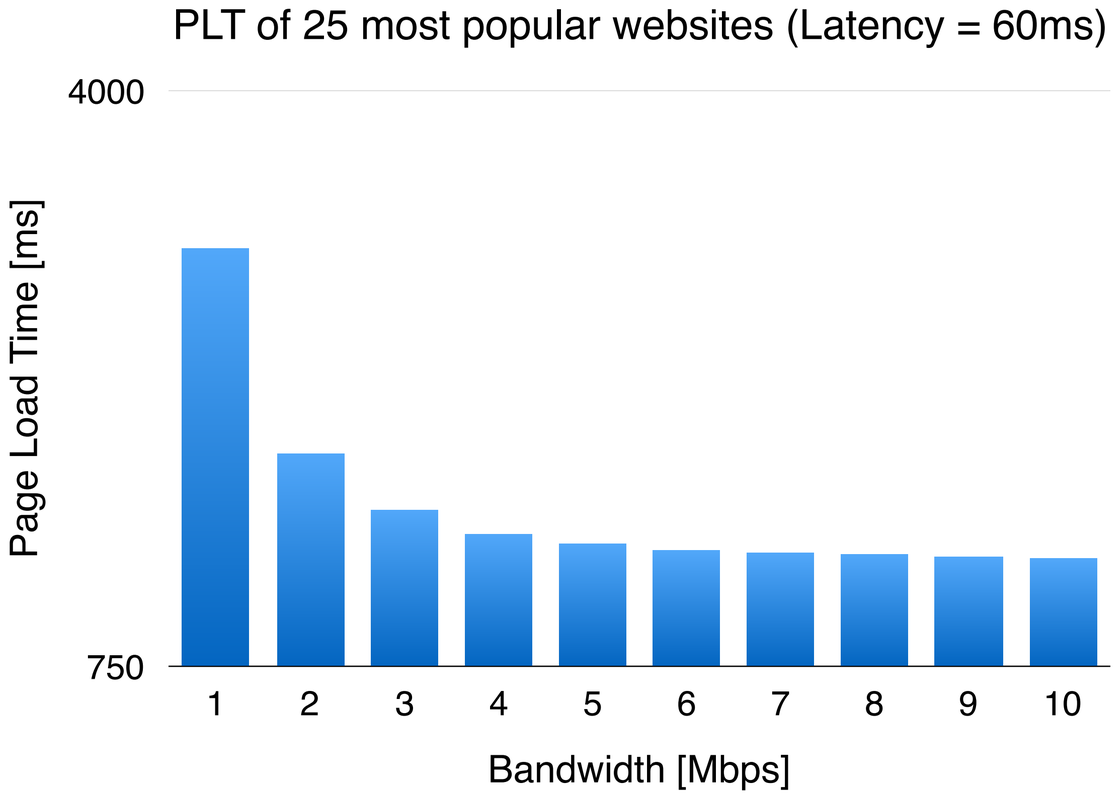
**We should change that!**

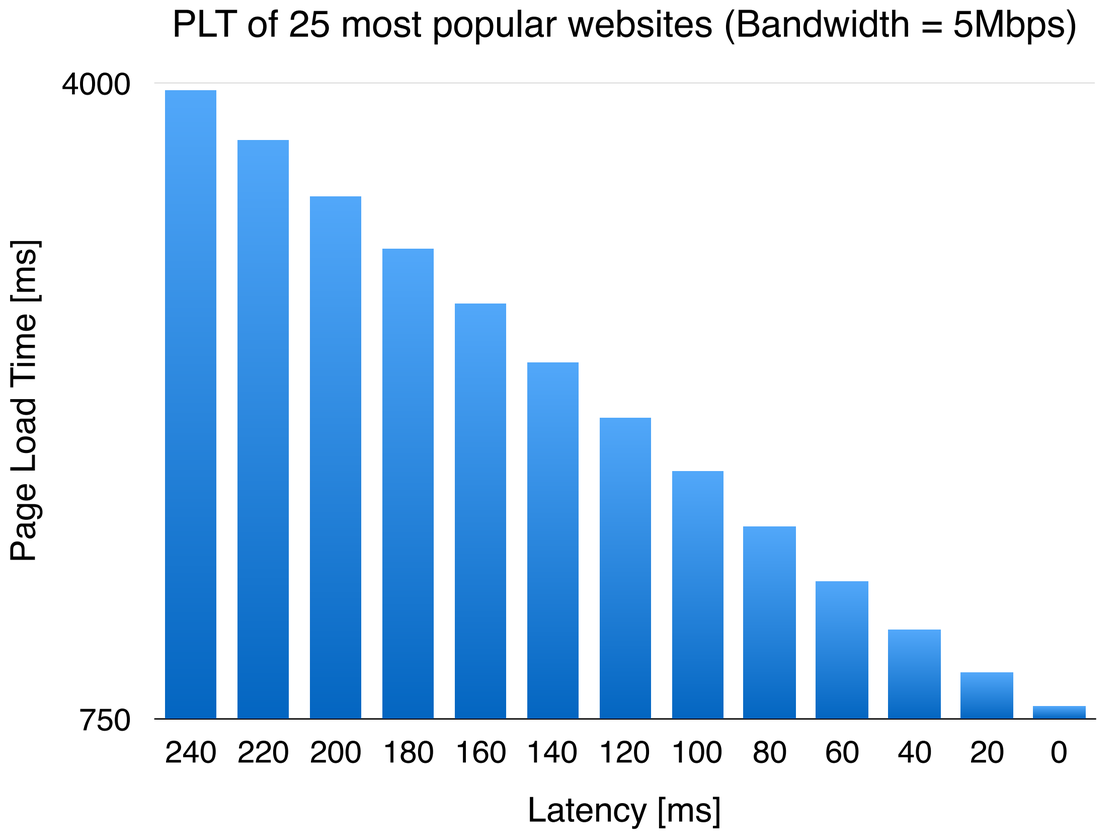
# Latency matters




*[1] "More Bandwidth Doesn't Matter (much)"*. M. Belshe. 2010 (https://goo.gl/X8rE6Q).

# Latency matters




*[1] "More Bandwidth Doesn't Matter (much)"*. M. Belshe. 2010 (https://goo.gl/X8rE6Q).

# Latency matters

- [2] measured impact of latency on service revenue
- Direct correlation between latency and revenue:
  - 100ms of additional delay has significant impact on the revenue and customer satisfaction

*[2] "Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks"*. S. Sundaresan, et al. ACM IMC 2013.

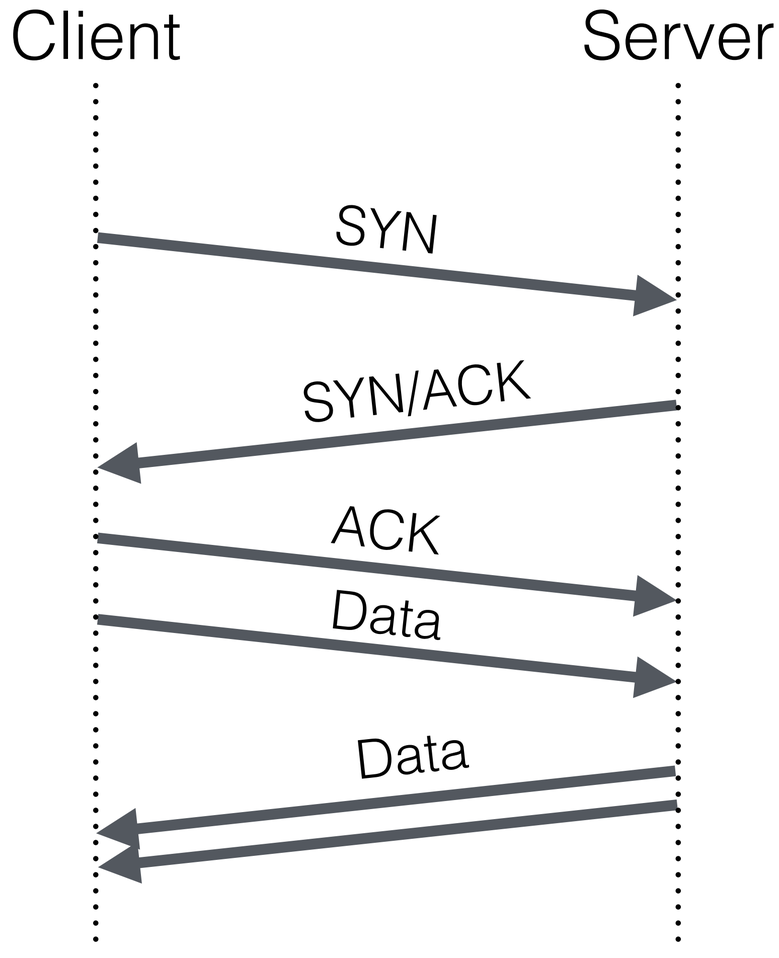
# Transmission Control Protocol

- Used for 95% of the Internet's traffic
- Provides a reliable and in-order byte-stream service
- 3-way handshake to establish the connection

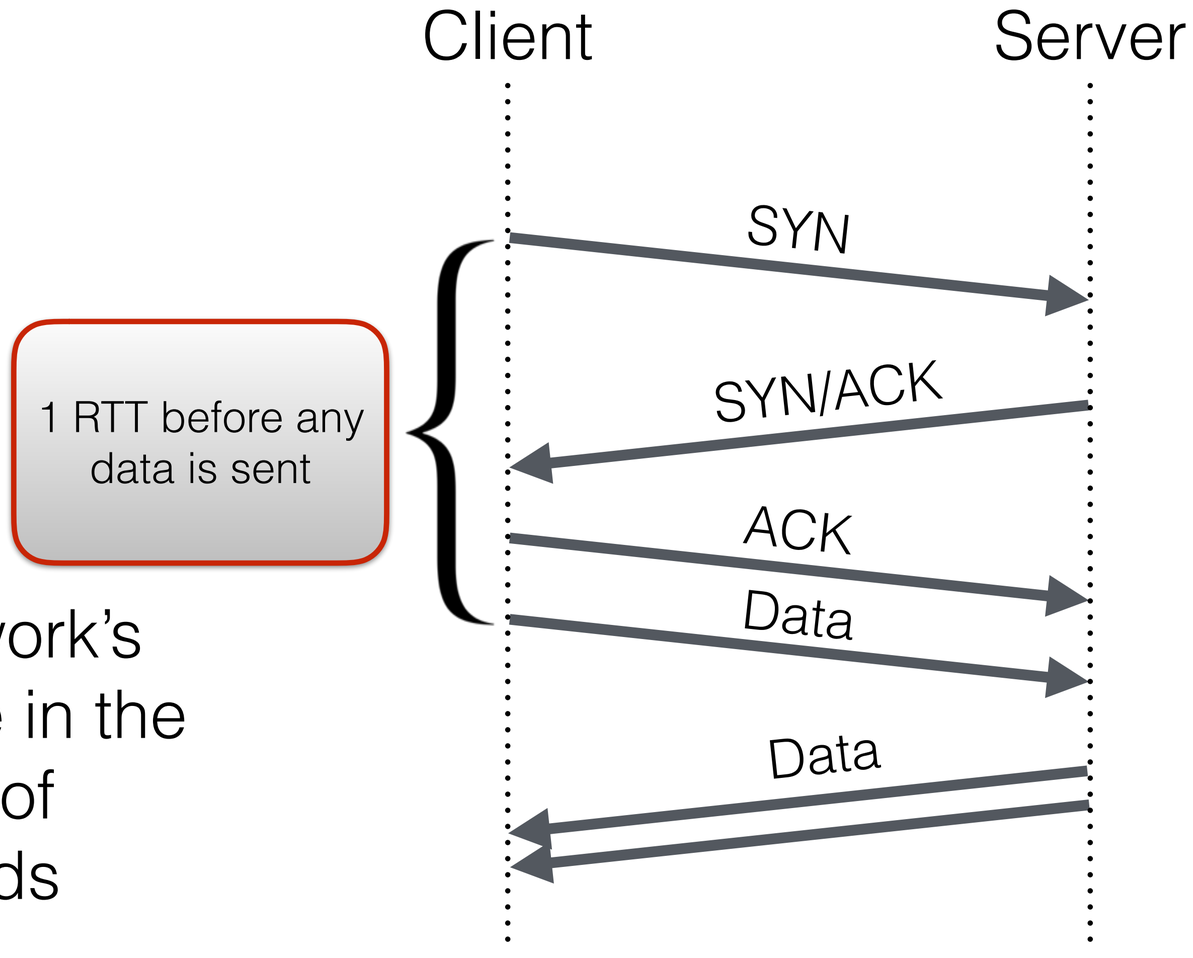
# The TCP joke

- ***“Hi, I’d like to hear a TCP joke.”***
- ***“Hello, would you like to hear a TCP joke?”***
- ***“Yes, I’d like to hear a TCP joke.”***
- ***“Ok, I’ll tell you a TCP joke.”***
- ***…***

# TCP Handshake is expensive



# TCP Handshake is expensive




Cellular Network's RTT can range in the hundreds of milliseconds

# TCP Fast Open (RFC 7413)

*Accelerating the TCP Handshake*

# TCP Fast Open (TFO)

- Allows clients to send SYN with data
- Enables servers to reply right away with the response
- Protects itself against DoS through a cookie, unique for each client-IP
- Standardized at the IETF - RFC 7413

*[3] "TCP Fast Open"*. Y. Cheng, J. Chu, S. Radhakrishnan, A. Jain. IETF RFC 7413. 2014

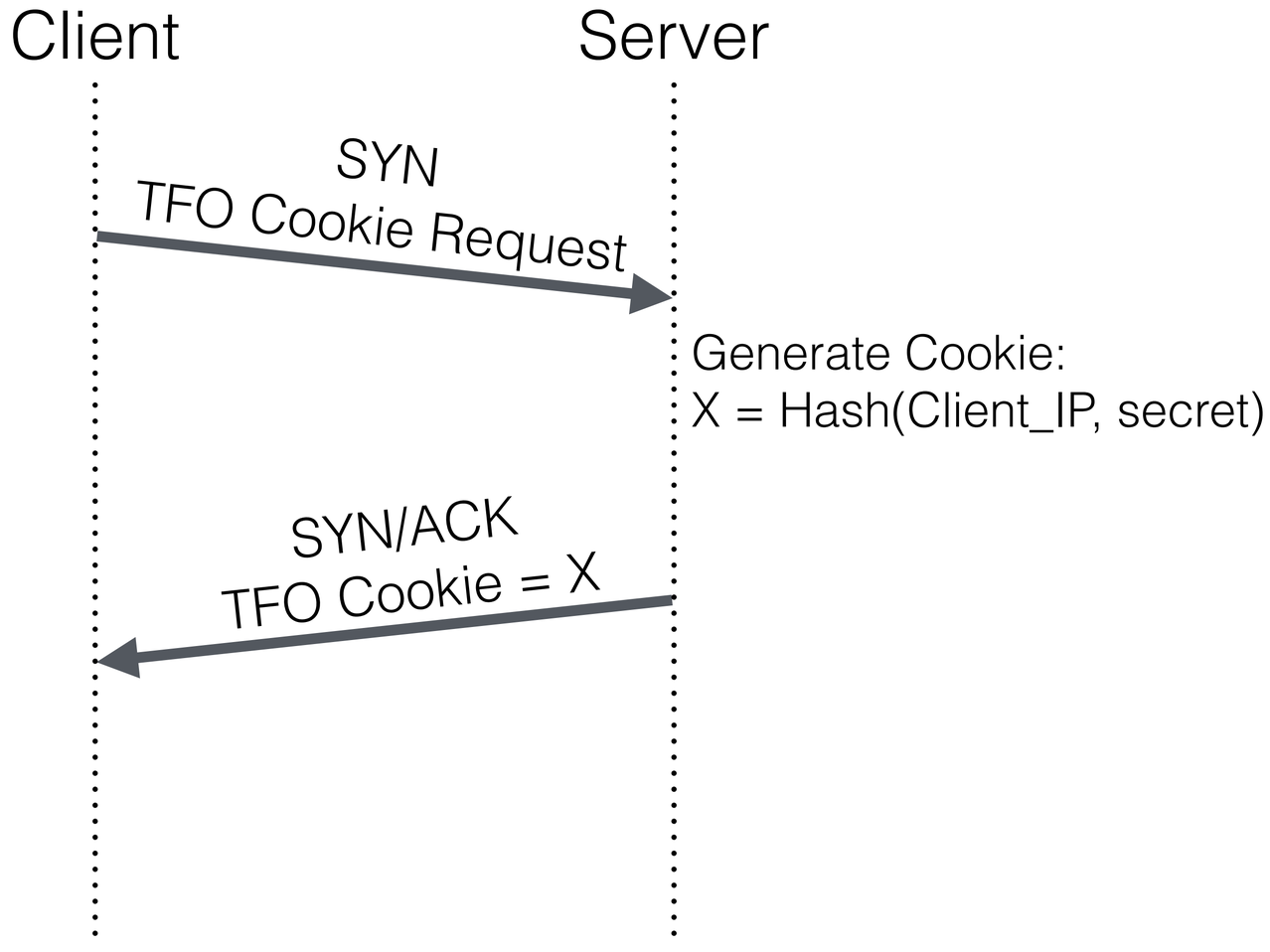
# TFO at Apple

- TCP Fast Open in iOS 9 and OS X 10.11 (and later)
- Used for an Apple Service on all iOS and OS X devices
- Public API by using `connectx(2)`
- Overall, very beneficial

**But, Firewalls got in our way**

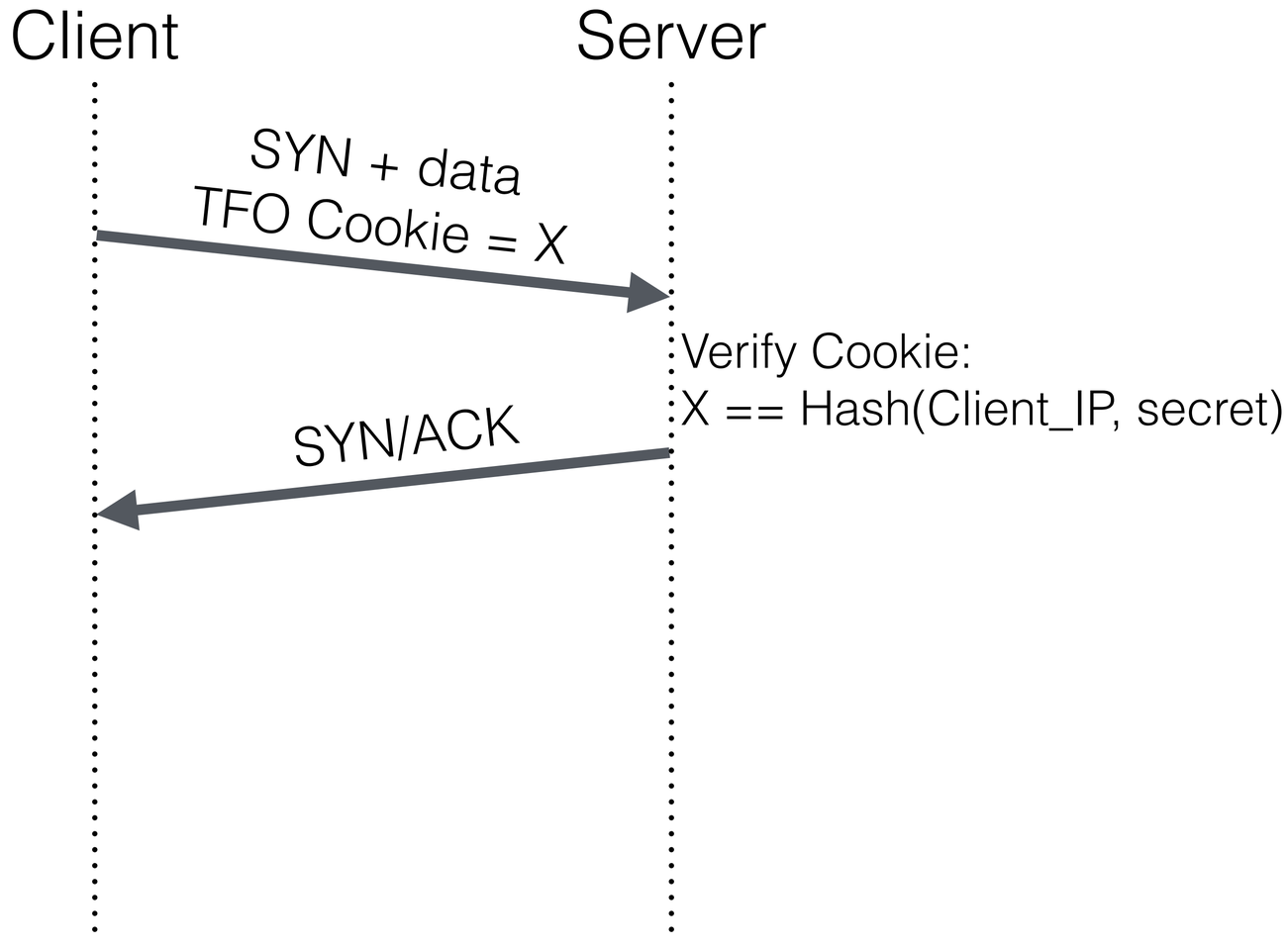
# TFO in details

1. Cookie Exchange



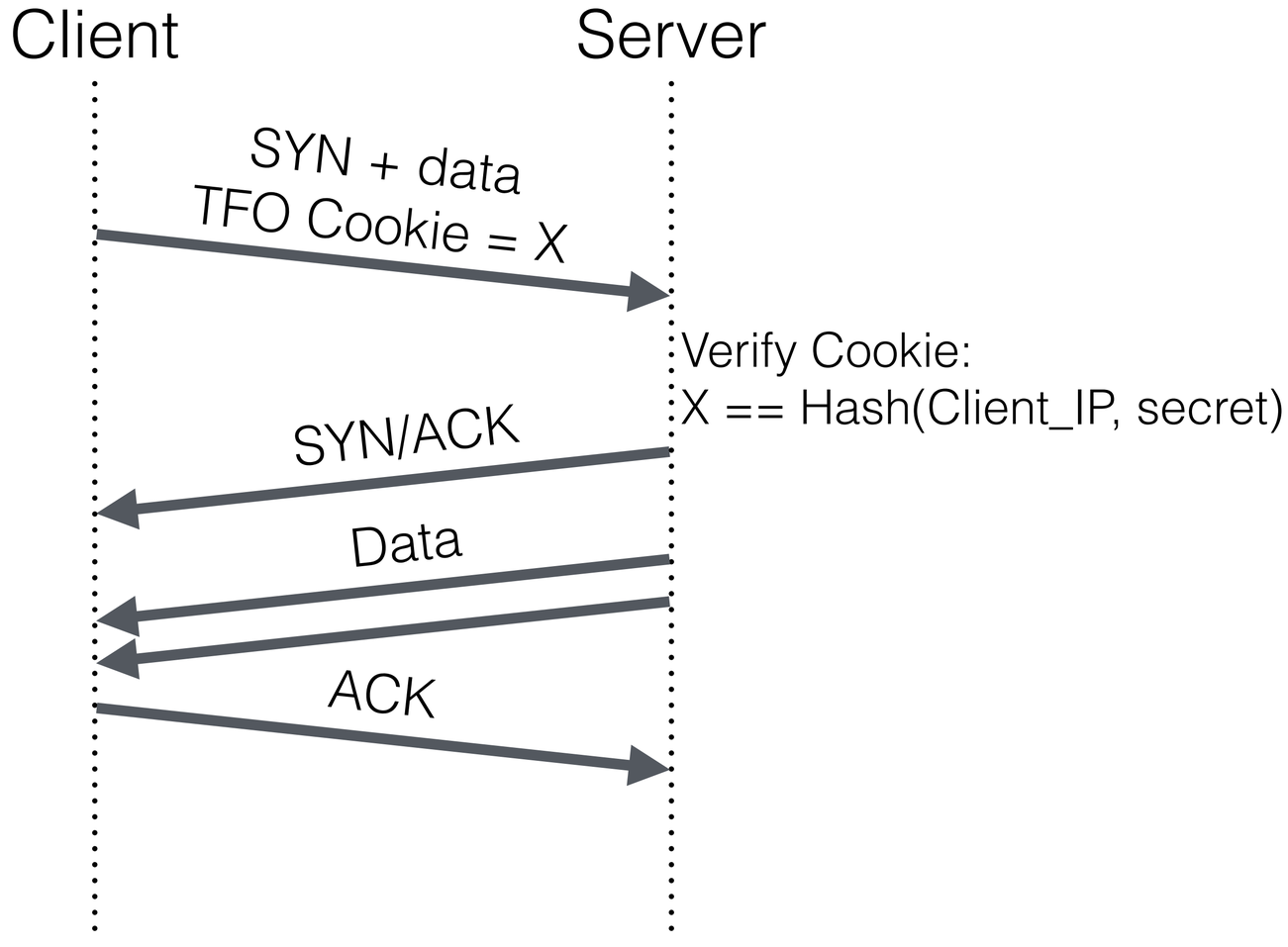
# TFO in details

2. Sending SYN + data



# TFO in details

3. Server replies with data



# Middlebox issues with TCP Fast Open

*… and their negative impact*

# Middlebox issues

- Bad Middleboxes and Firewalls respond badly to TCP Fast Open
  - **Suppress** TCP options
  - **Drop** packets
  - Mark entire connection as “**invalid**”
  - **Blackhole** the clients

# Using a new TCP option

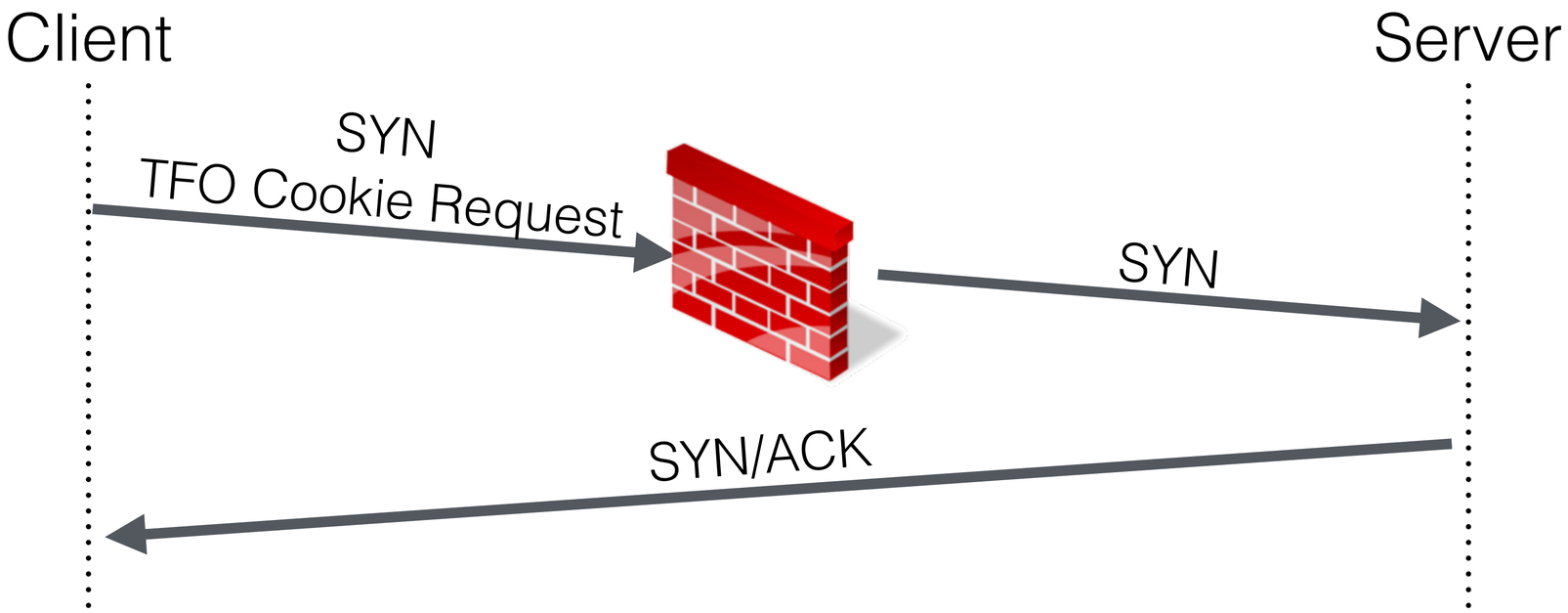
*Issue*

Simplistic middleboxes **remove** unknown TCP options

# Using a new TCP option

*Issue*

Simplistic middleboxes **remove** unknown TCP options



# Using a new TCP option

*Issue*

Simplistic middleboxes **remove** unknown TCP options

*Impact*

Clients **cannot use TFO**, and thus pay a latency cost compared to well-behaving networks

# Using a new TCP option

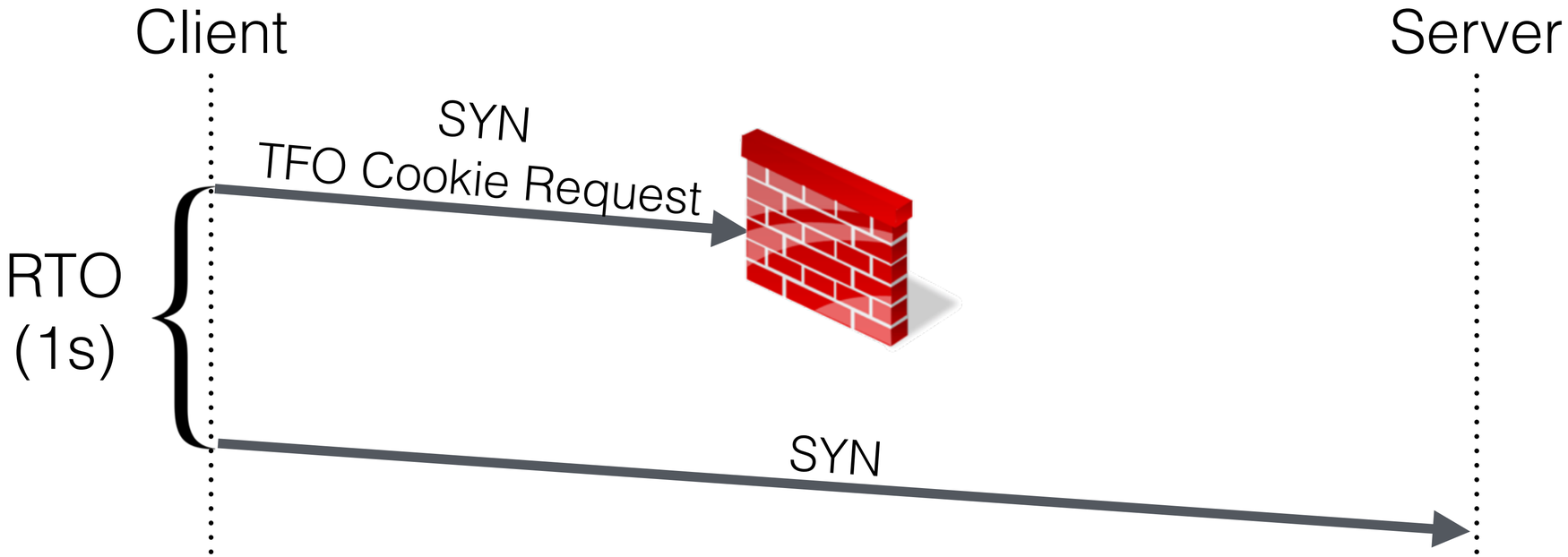
*Issue*

Simplistic middleboxes **drop segments with** unknown TCP options

# Using a new TCP option

*Issue*

Simplistic middleboxes **drop segments with** unknown TCP options



# Using a new TCP option

*Issue*

Simplistic middleboxes **drop segments with** unknown TCP options

*Impact*

Client **has to retransmit the SYN-segment** without the TCP option. The user experiences a **high page-load-time**.

# Sending SYN+data

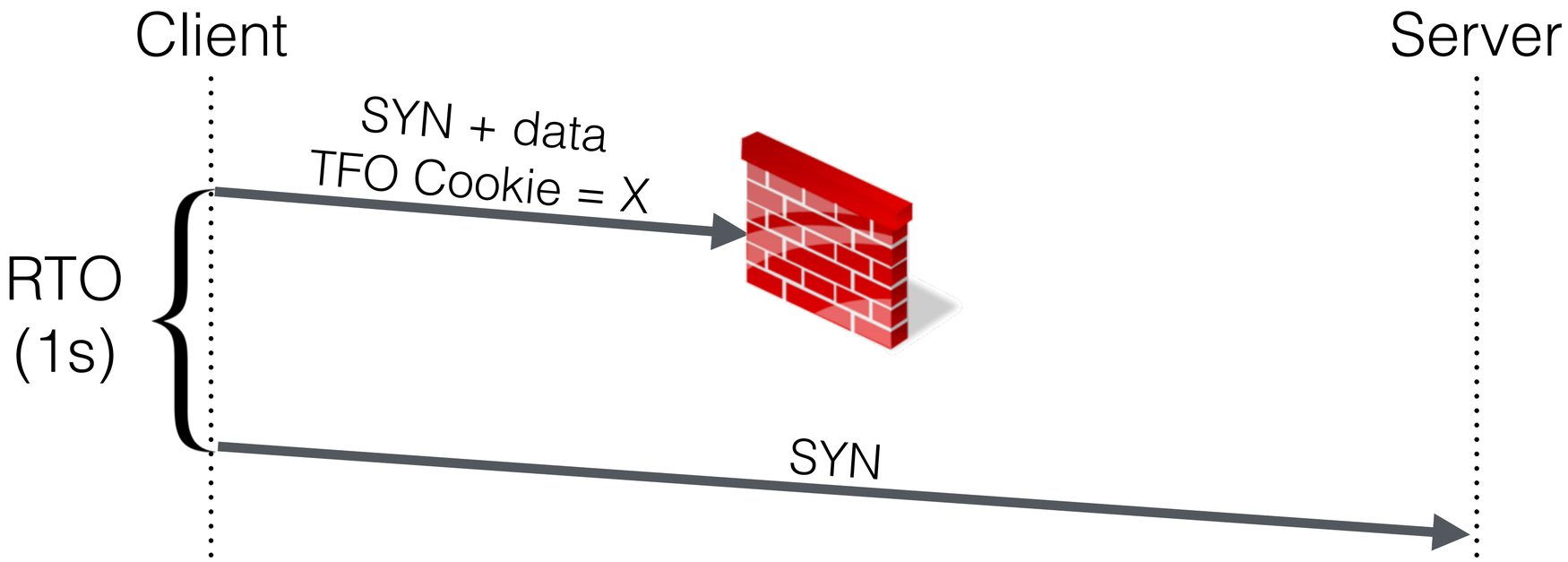
*Issue*

Naive middleboxes **drop SYN segments** with data

# Sending SYN+data

*Issue*

Naive middleboxes **drop SYN segments** with data



# Sending SYN+data

*Issue*

Naive middleboxes **drop SYN segments** with data

*Impact*

Clients **has to retransmit the SYN-segment** without the TCP option. The user experiences a **high page-load-time**.

# Acknowledging SYN+data

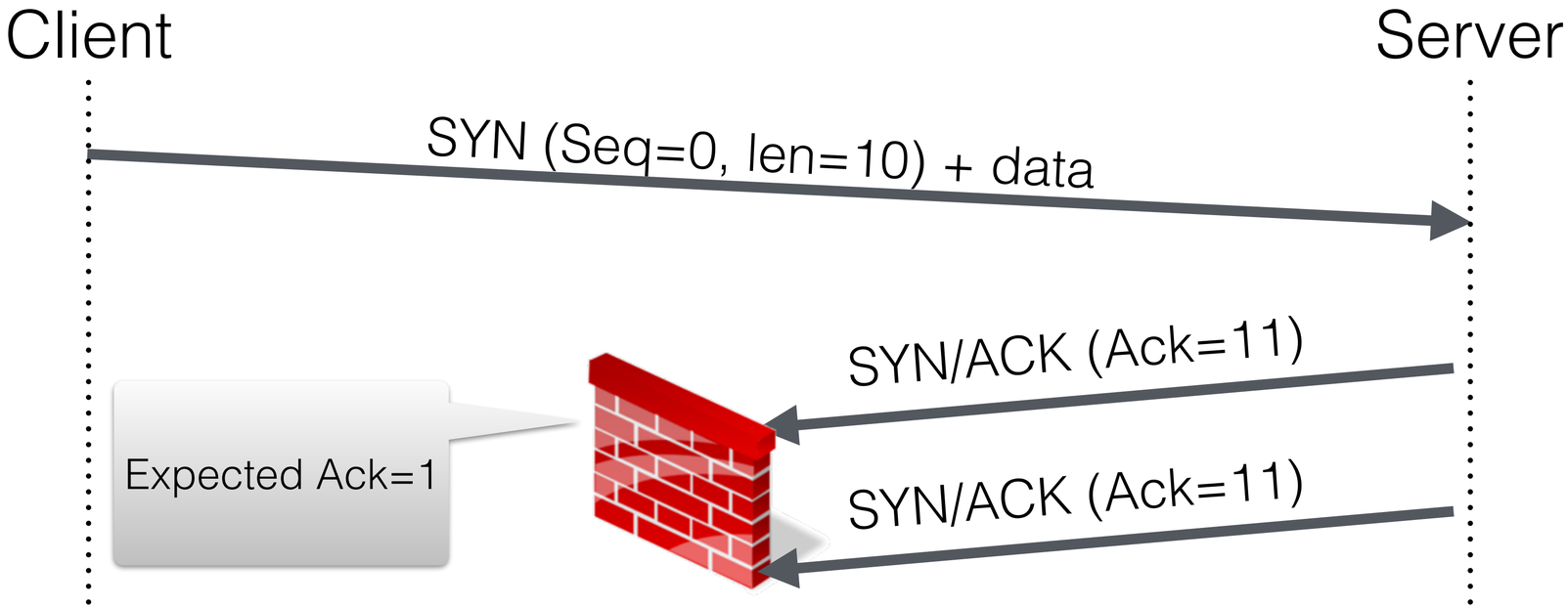
*Issue*

The server **acknowledges the SYN+data**, thus more than the initial sequence number. Middleboxes might **drop the SYN/ACK**.

# Acknowledging SYN+data

*Issue*

The server **acknowledges the SYN+data**, thus more than the initial sequence number. Middleboxes might **drop the SYN/ACK**.



# Acknowledging SYN+data

*Issue*

The server **acknowledges the SYN+data**, thus more than the initial sequence number. Middleboxes might **drop the SYN/ACK**.

*Impact*

The middlebox keeps on blocking the server's SYN/ACK. The session **never becomes established**.

# Server sends data right before 3-way handshake completes

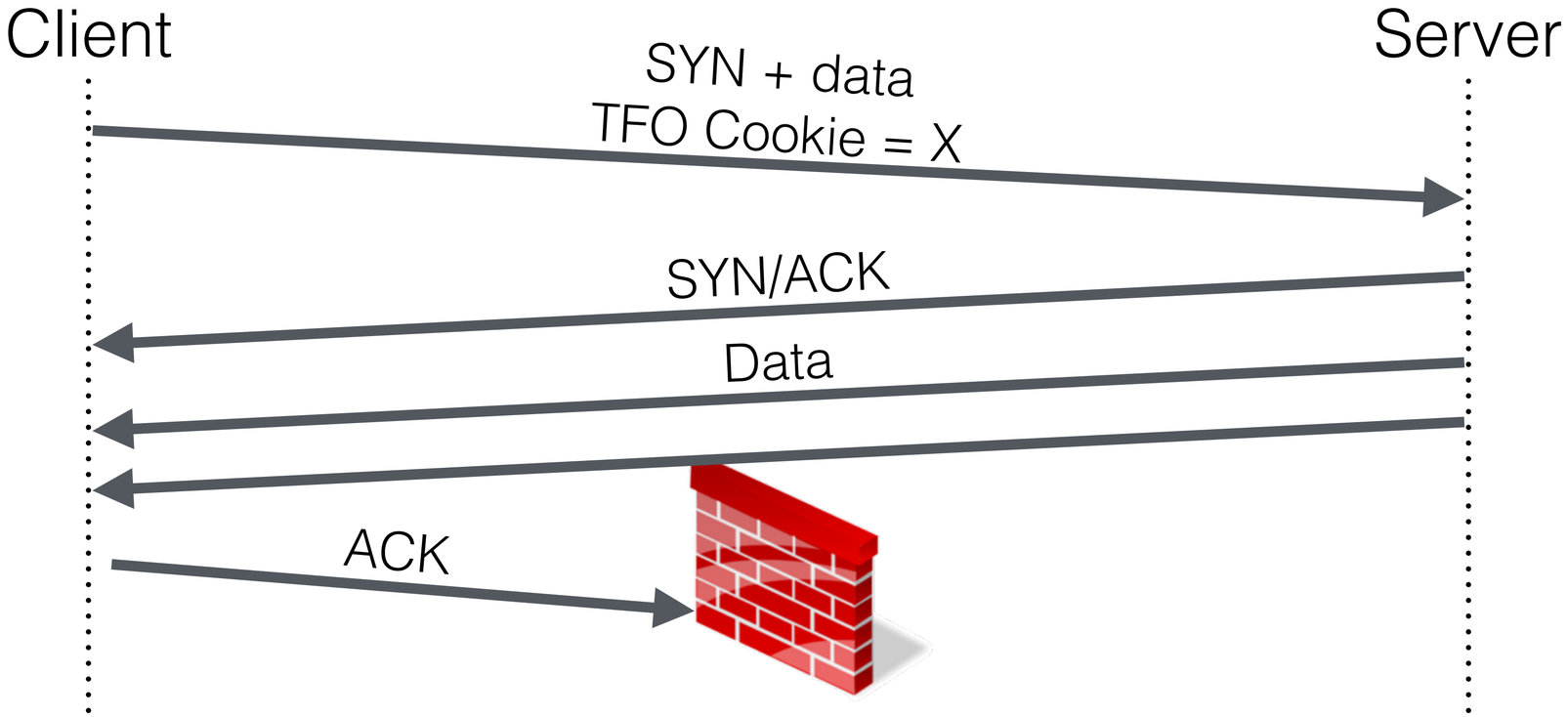
*Issue*

Bad Intrusion Detection Systems (IDS) start
**blackholing** the client

# Server sends data right before 3-way handshake completes

*Issue*

Bad Intrusion Detection Systems (IDS) start **blackholing** the client



# Server sends data right before 3-way handshake completes

*Issue*

Bad Intrusion Detection Systems (IDS) start **blackholing** the client

*Impact*

Client **loses connectivity** to the server. Subsequent connections (non-TFO) also might be blocked by the IDS.

# How common is this?

Mostly, TFO works successfully (~80% success-rate).

But…

# How common is this?

Mostly, TFO works successfully (~80% success-rate).

But…

100% of the users of the affected networks are penalized

# Conclusion

- Latency has a direct impact on user-experience
- TCP Fast Open allows to significantly reduce latency
- Bad middleboxes are interfering with TCP Fast Open

*Vendors and operators:*
***Take TFO into account for a better user-experience***

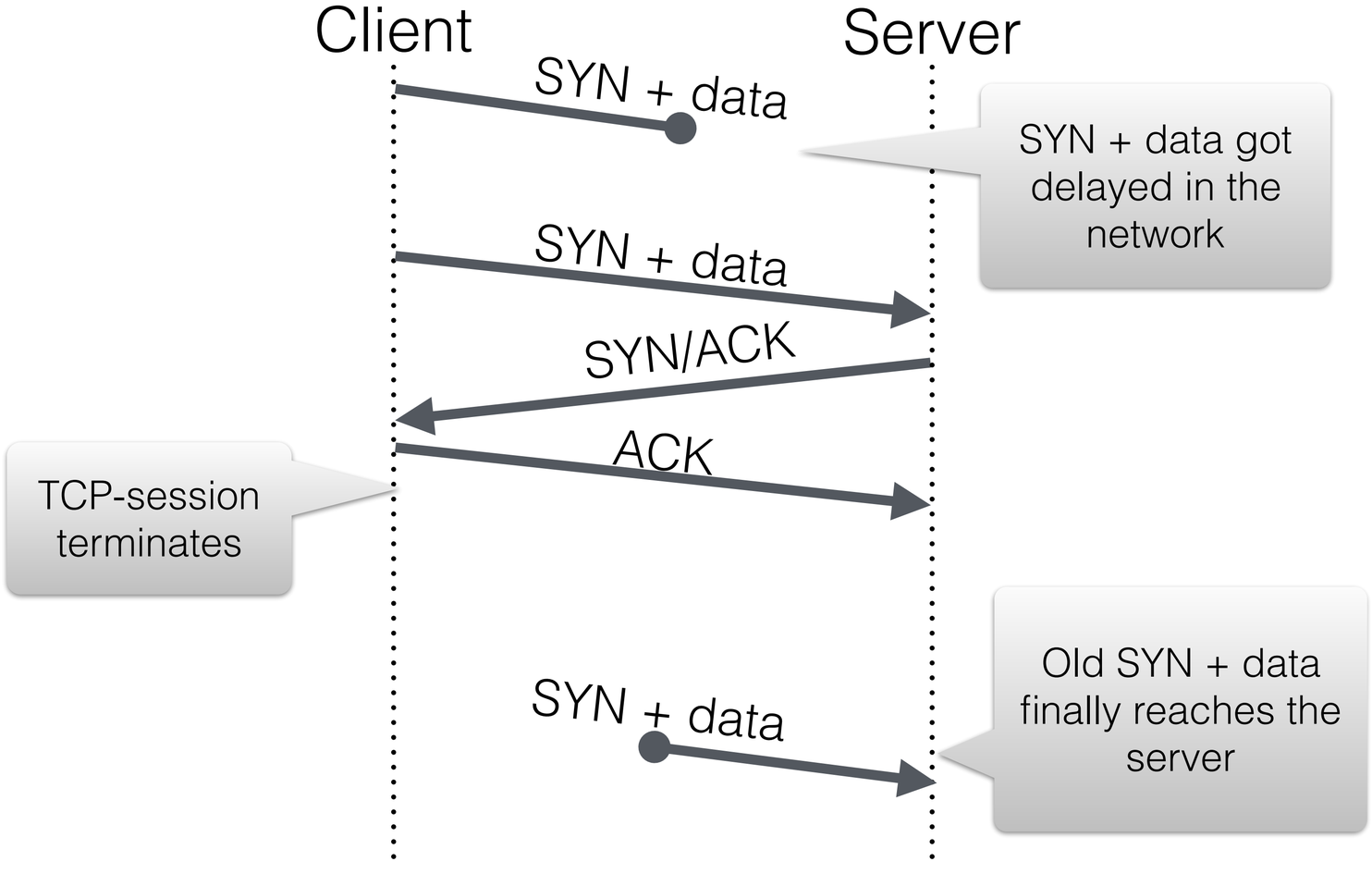
# References

[1] *“More Bandwidth Doesn’t Matter (much)”*. M. Belshe. 2010 (https://goo.gl/X8rE6Q).
[2] *“Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks”*. S. Sundaresan, et al. ACM IMC 2013.
[3] *“TCP Fast Open”*. Y. Cheng, J. Chu, S. Radhakrishnan, A. Jain. IETF RFC 7413. 2014

# Backup-slides

# TFO and idempotency

- Data sent in a SYN might reach the server twice



# TFO and idempotency

- Use TFO only with **“idempotent”** data (aka., data that can be received twice by the server)

  E.g.,:

  - TLS (ClientHello)
  - HTTP-Requests