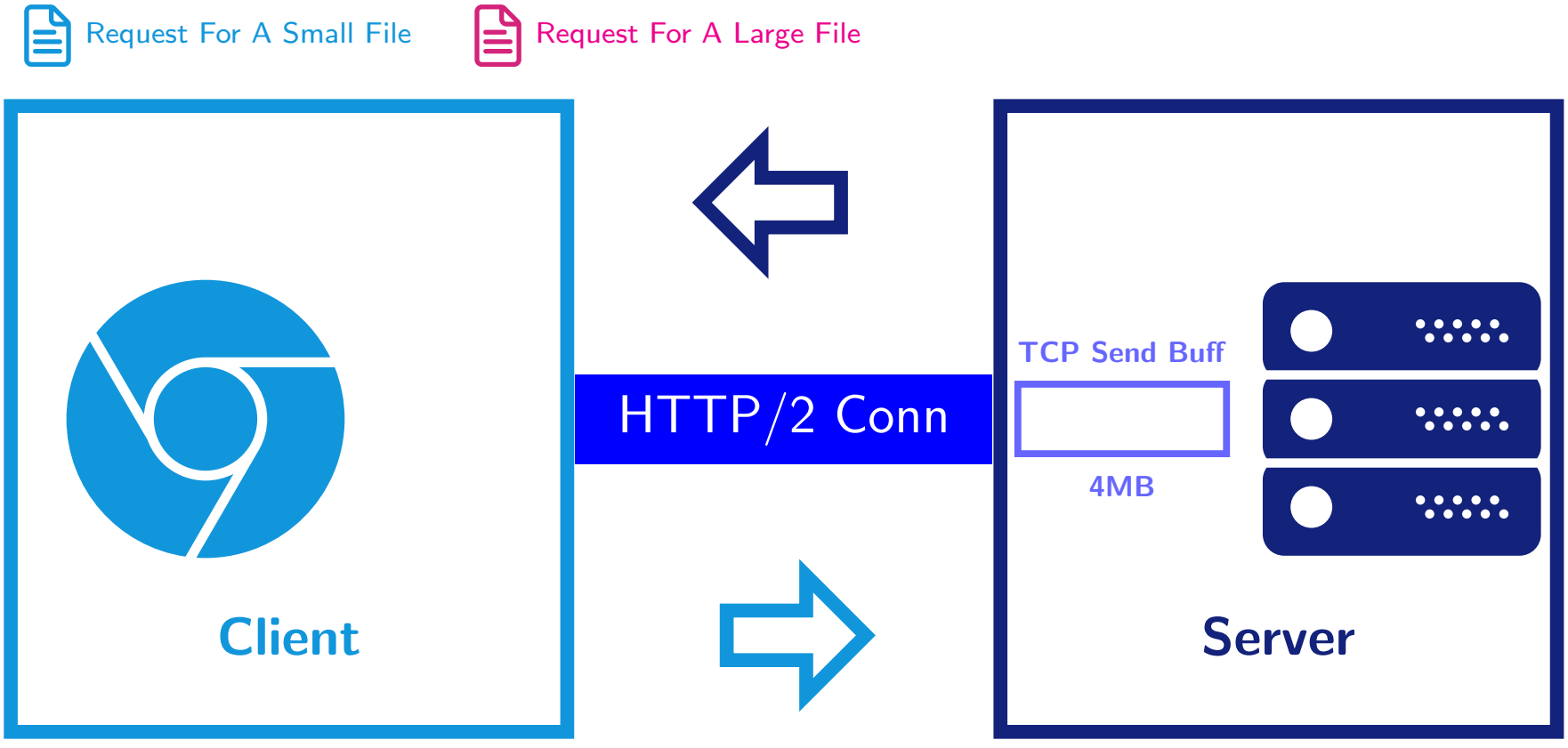
SMig: A Stream Migration Extension For HTTP/2

Xianghang Mi **Feng Qian** XiaoFeng Wang

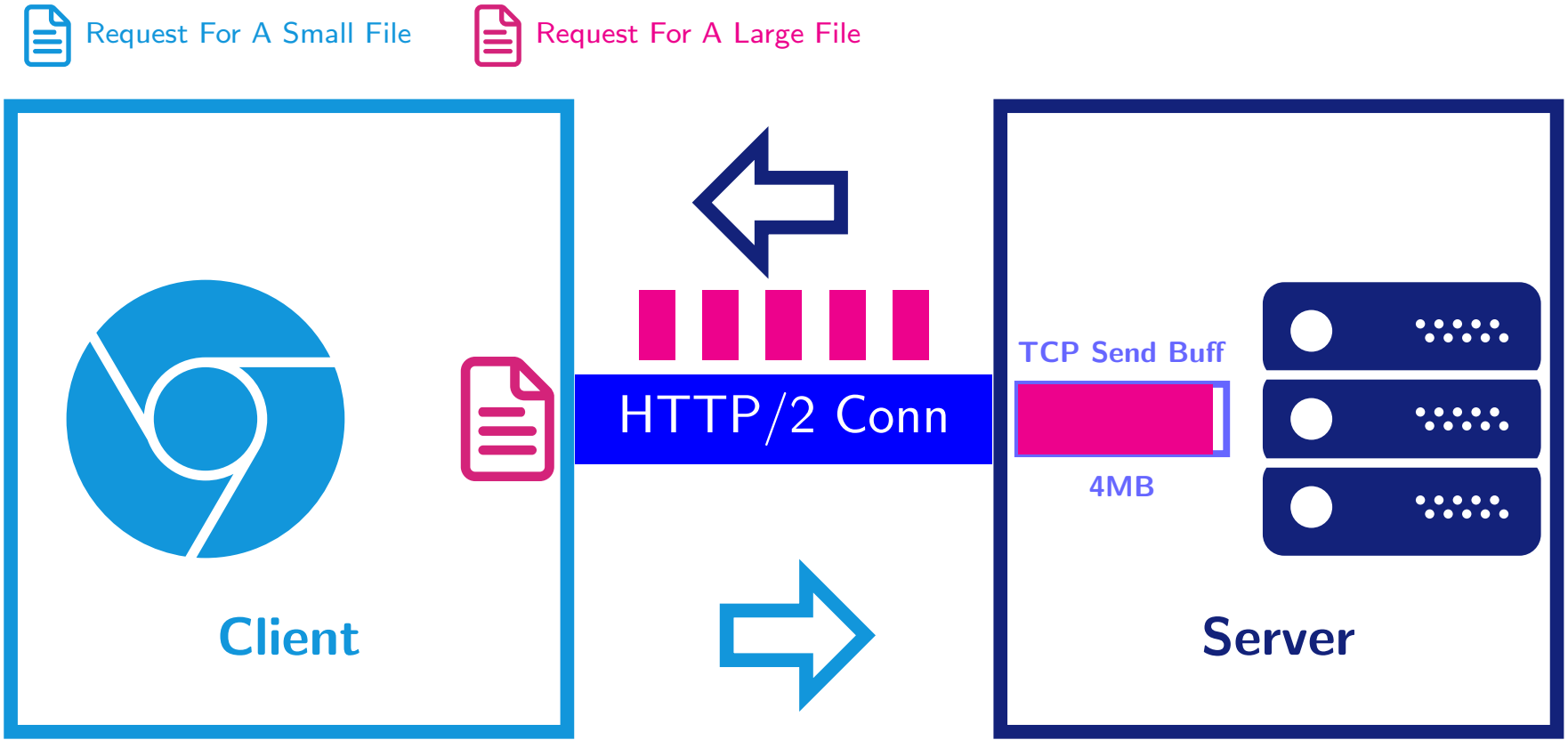
Department of Computer Science
Indiana University Bloomington

IETF 98 httpbis Meeting
Chicago IL, 3/31/2017

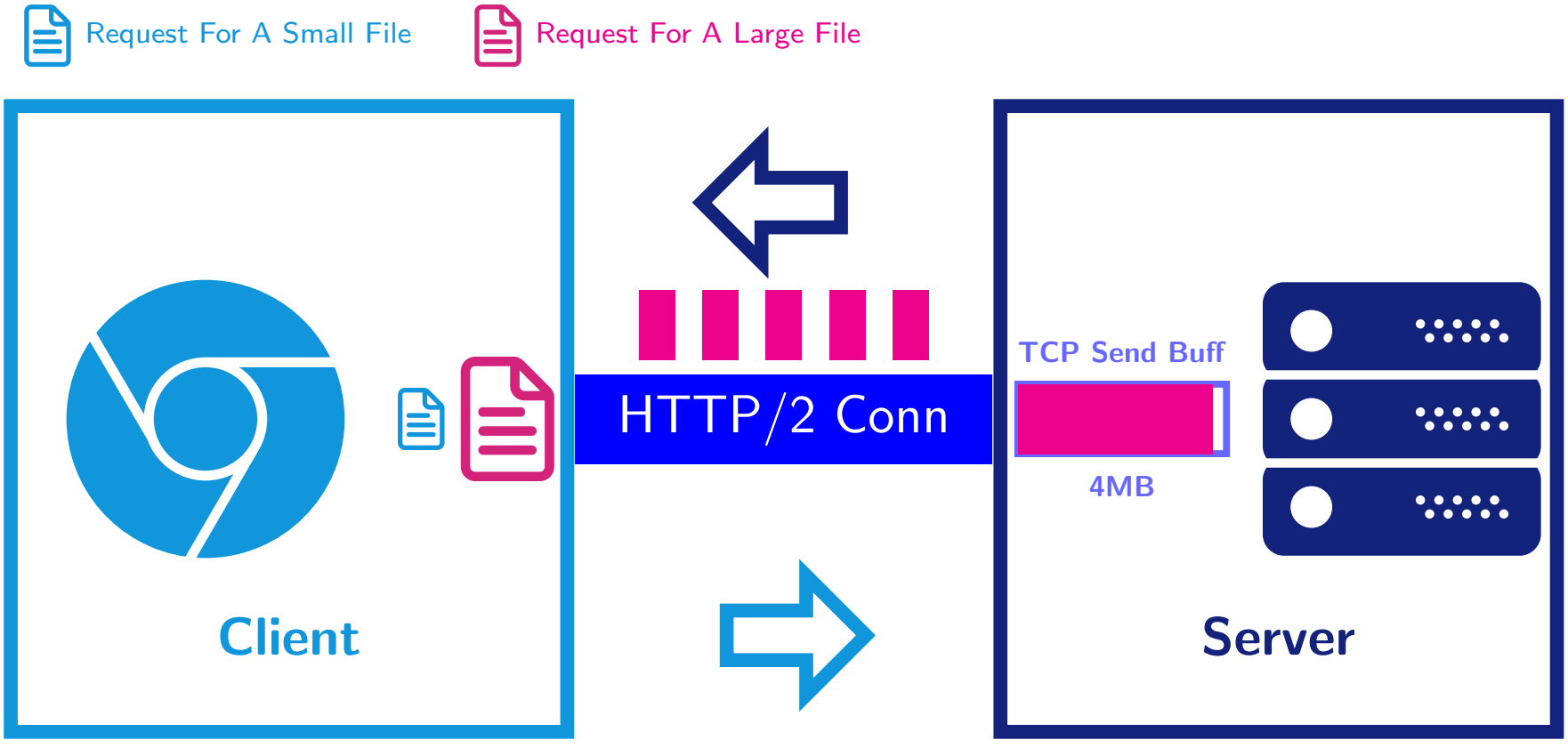
Motivations of SMig: Sender-side HoLB Problem



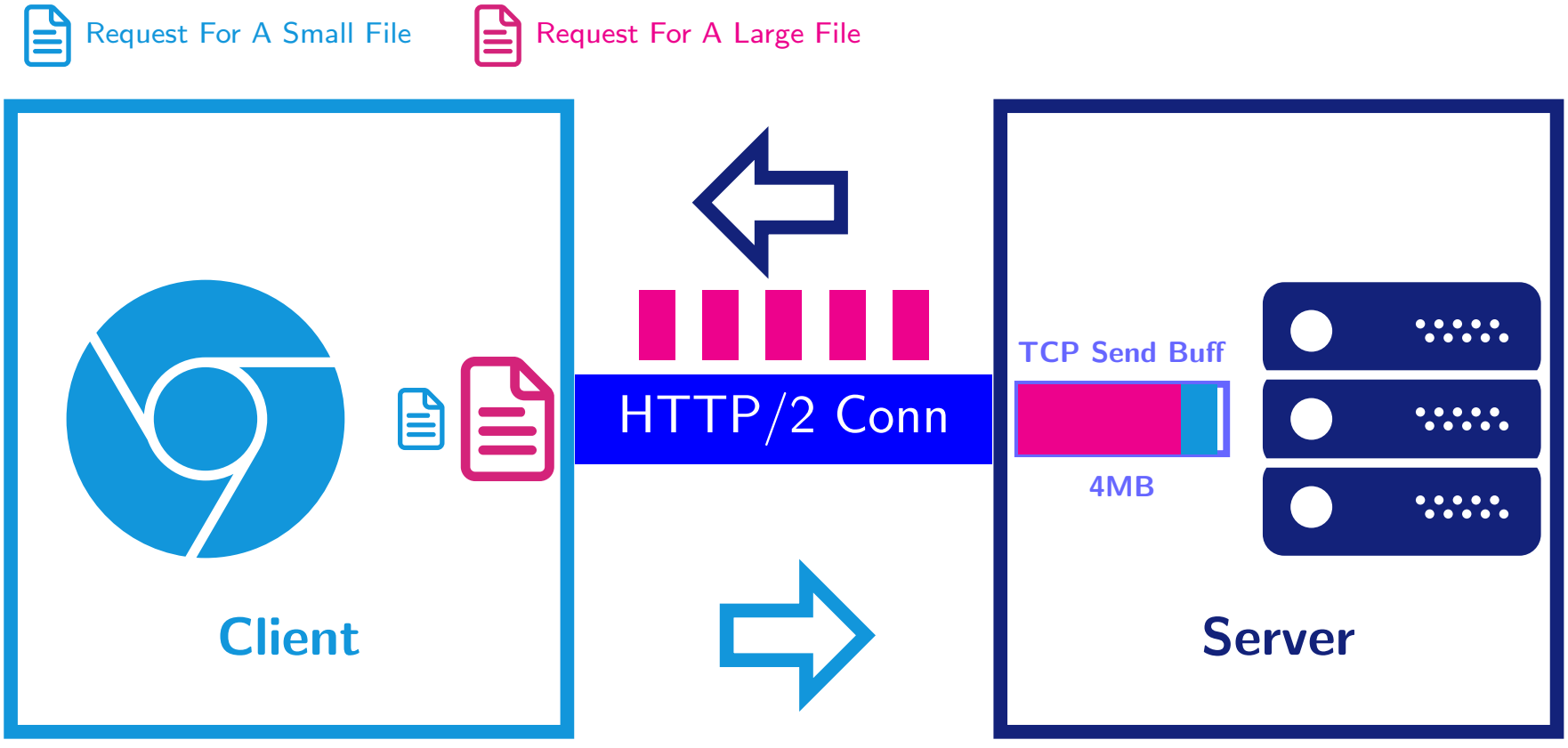
Motivations of SMig: Sender-side HoLB Problem



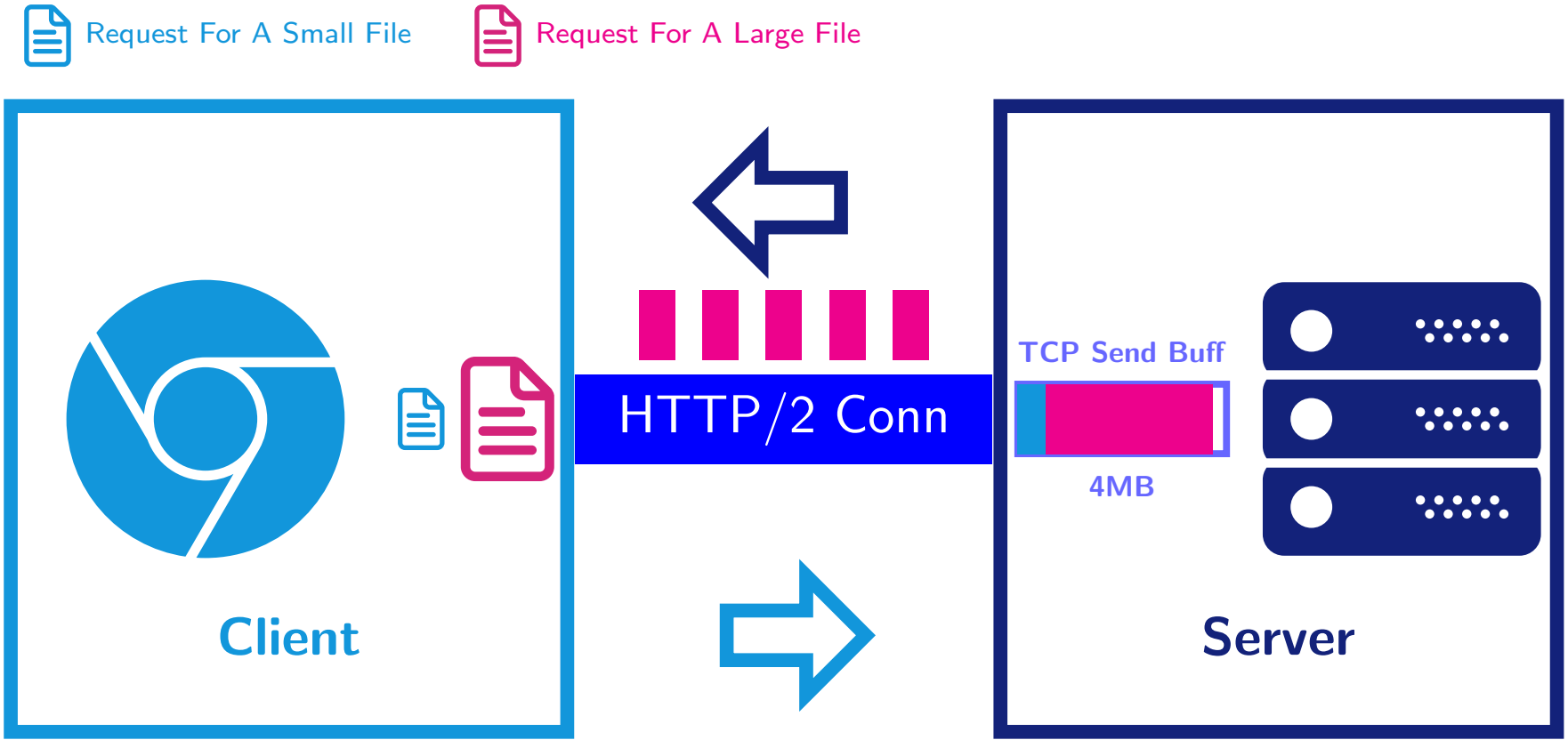
Motivations of SMig: Sender-side HoLB Problem



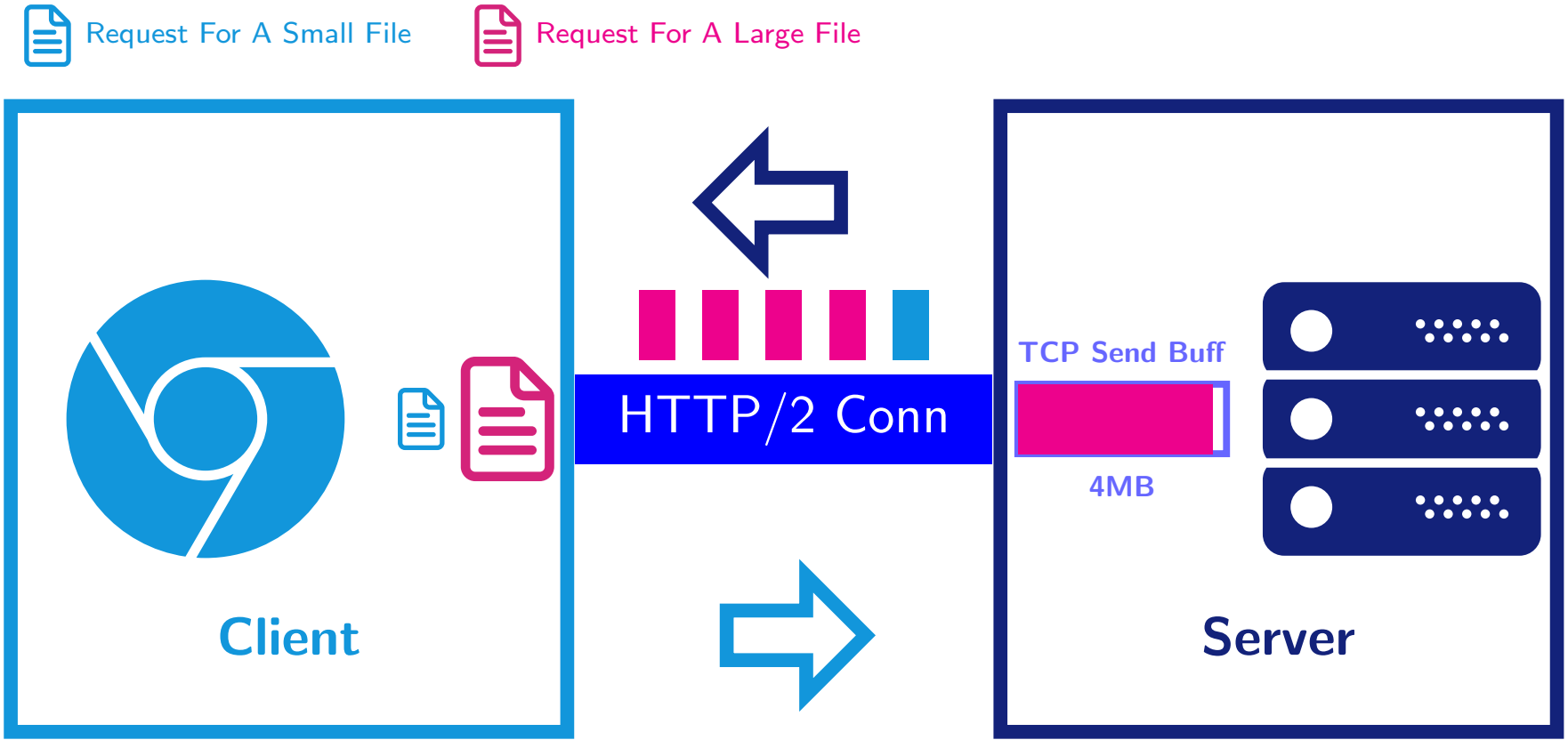
Motivations of SMig: Sender-side HoLB Problem



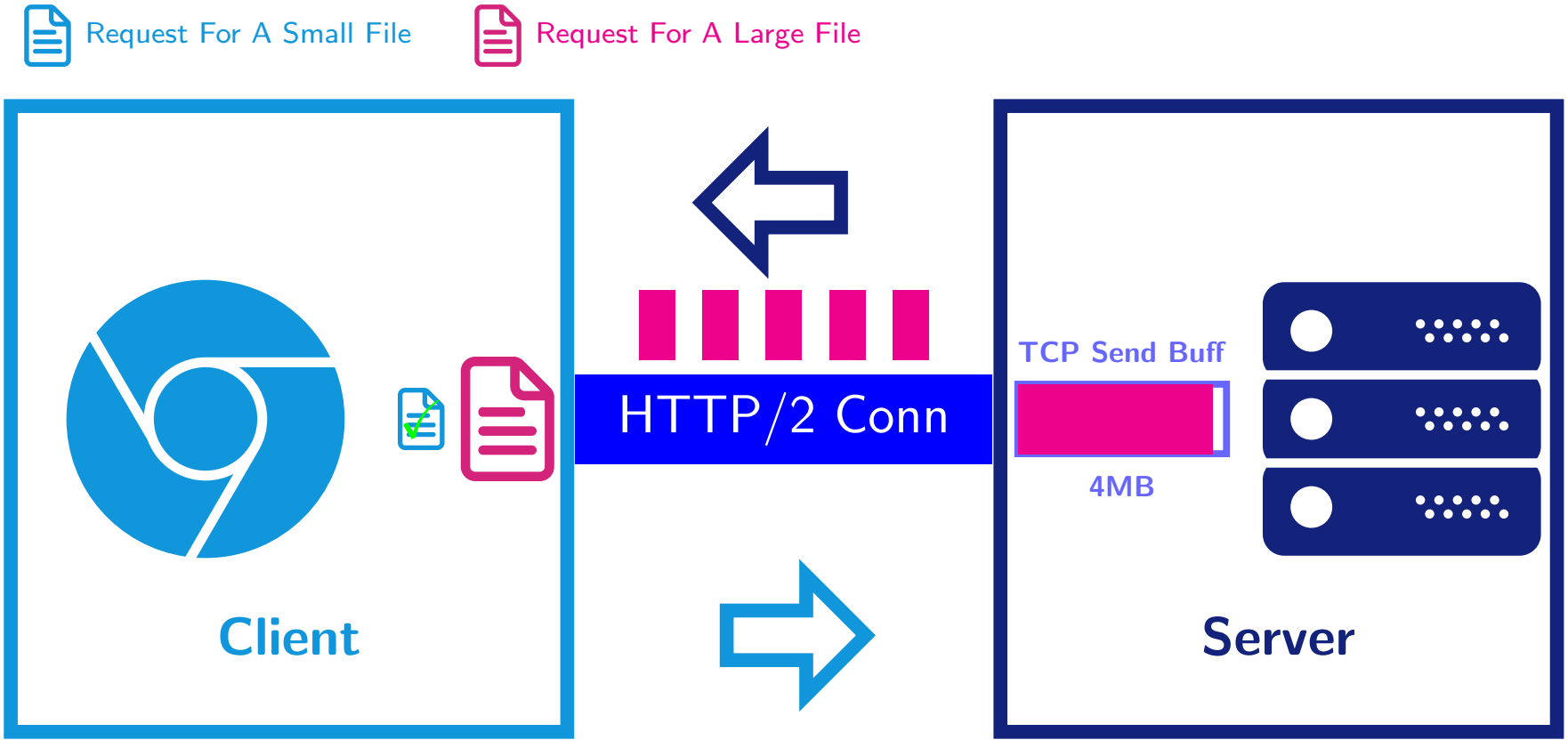
Motivations of SMig: Sender-side HoLB Problem



Motivations of SMig: Sender-side HoLB Problem



Motivations of SMig: Sender-side HoLB Problem



Motivations of SMig: Sender-side HoLB Problem

Table: Download Time for 10KB file (10Mbps BW, 50ms RTT)

Concurrent Download	HTTPS(HTTP/1.1)	HTTP/2
No	0.05	0.05
Yes	0.14	8.40

HoLB increases the small file download time by up to 70x, compared to HTTP/1.1!

Motivations of SMig: Sender-side HoLB Problem

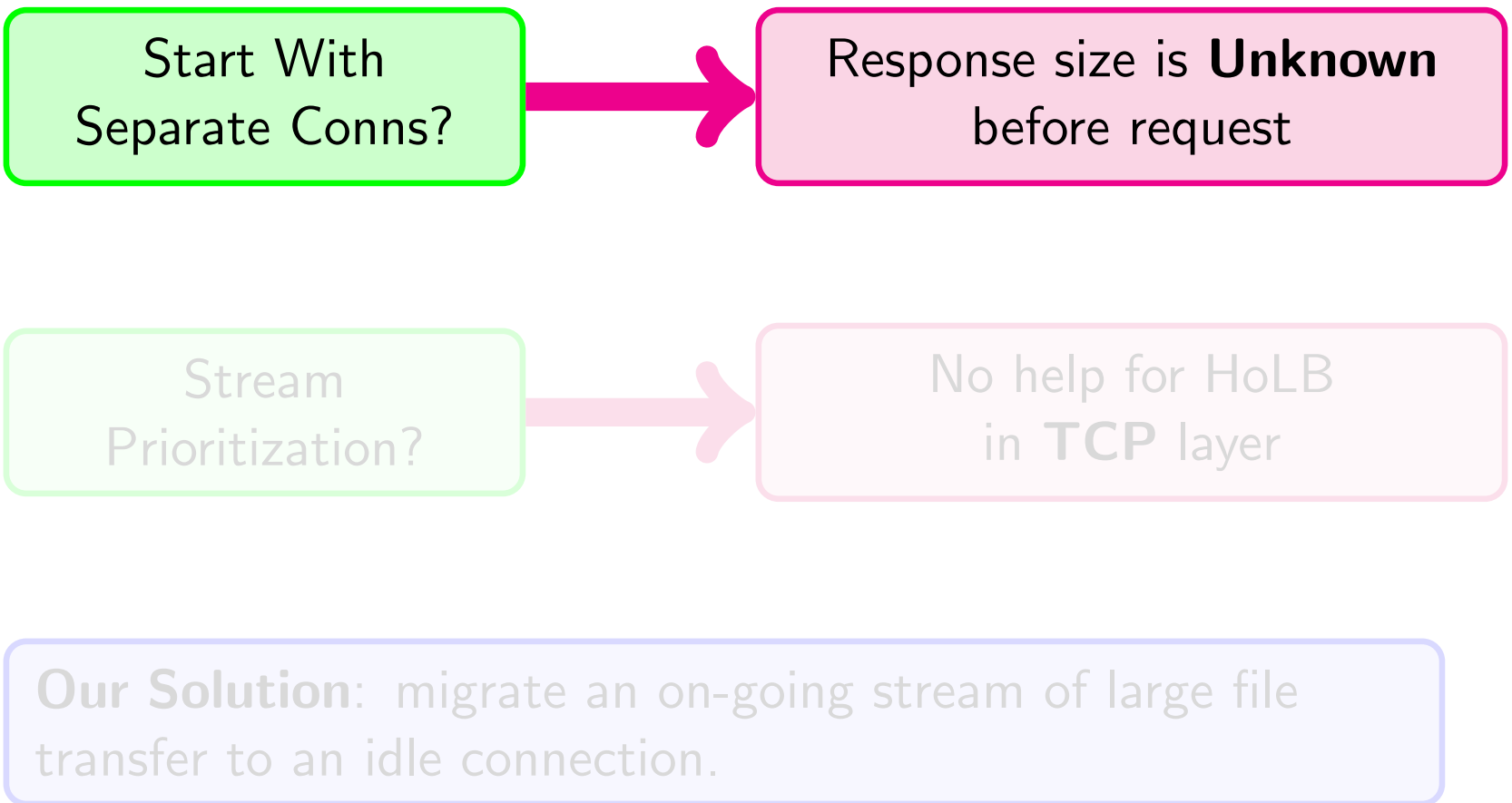
Table: Download Time for 10KB file (10Mbps BW, 50ms RTT)

Concurrent Download	HTTPS(HTTP/1.1)	HTTP/2
No	0.05	0.05
Yes	0.14	8.40

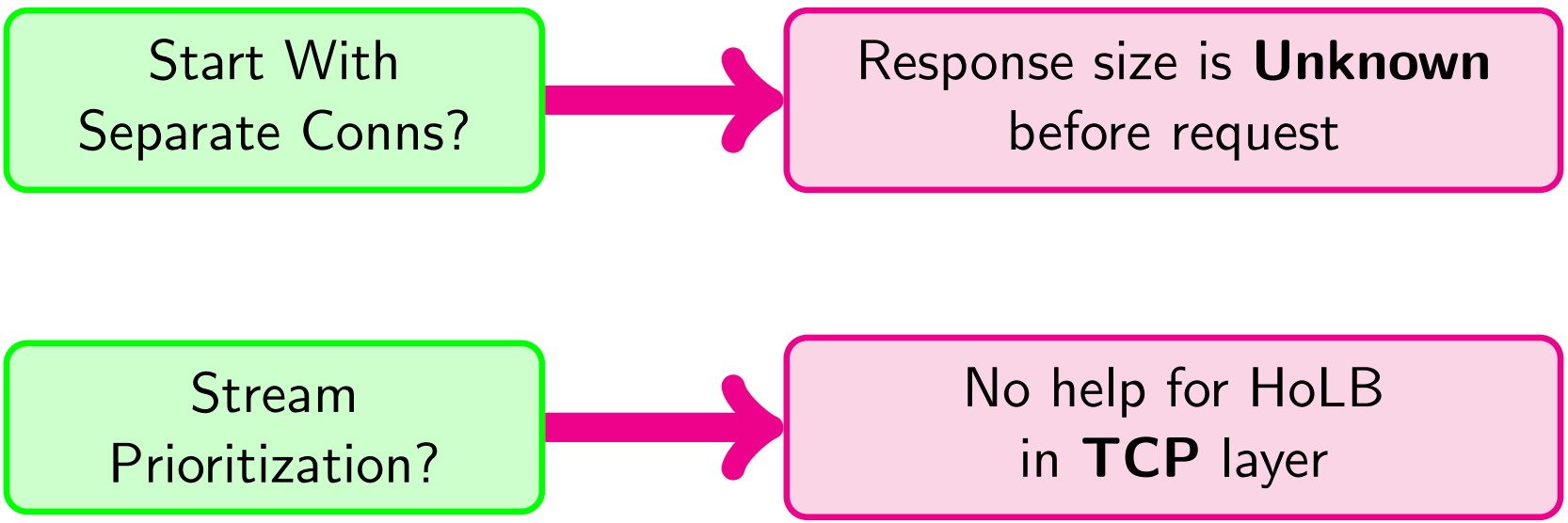
HoLB increases the small file download time by up to 70x, compared to HTTP/1.1!

HoLB may frequently happen in the real world: >14% of Alexa top 1500 websites have 1MB+ objects within the top three levels of their landing pages.

Motivations of SMig: How to Handle Sender-side HoLB?

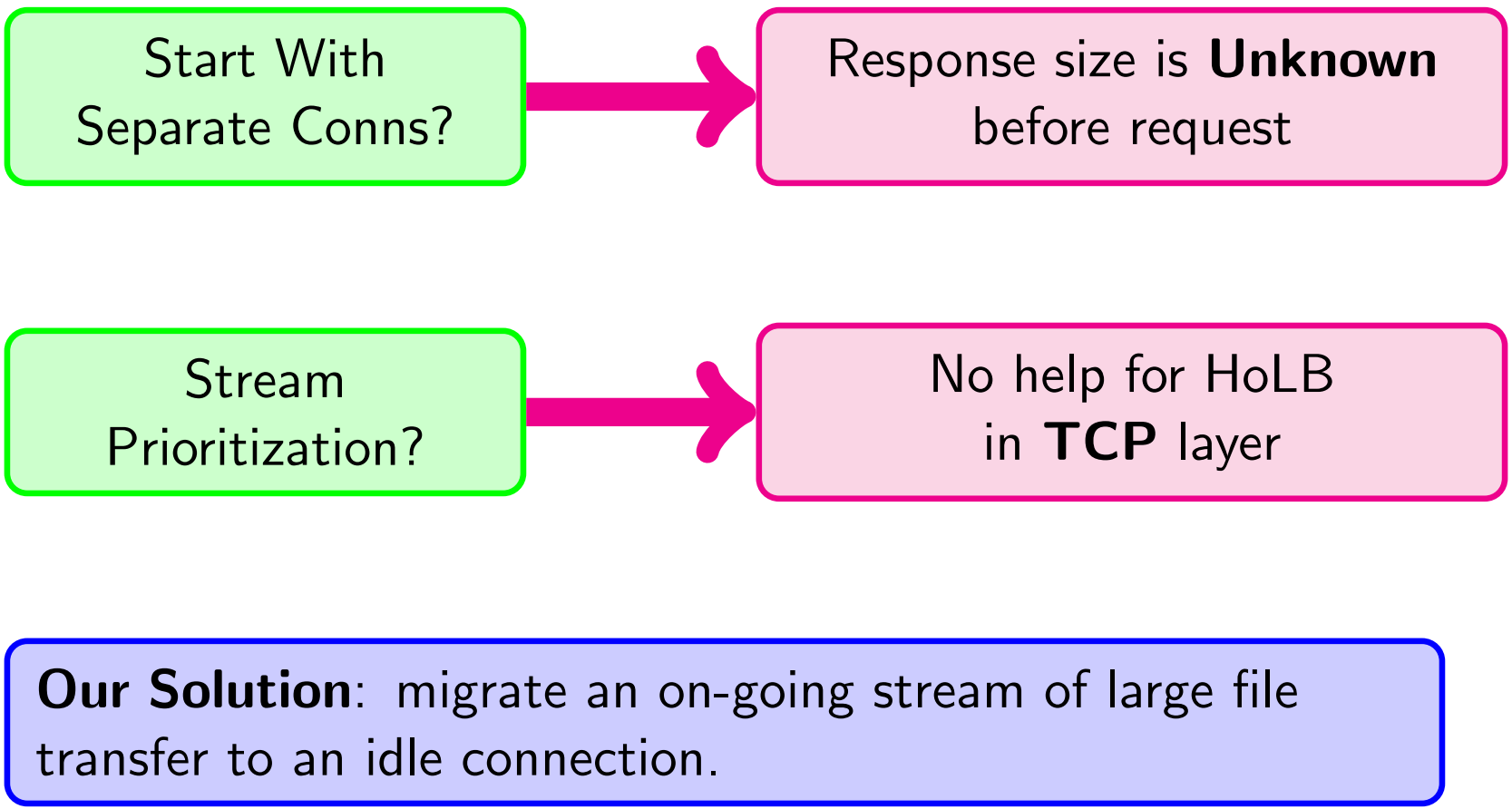


Motivations of SMig: How to Handle Sender-side HoLB?



Our Solution: migrate an on-going stream of large file transfer to an idle connection.

Motivations of SMig: How to Handle Sender-side HoLB?



SMig – Typical Usage Scenario

Client sends one or more HTTP/2 requests to server over a multiplexed connection.

If small and large objects are multiplexed together, server migrates large objects to separate connections.

Each large object uses a dedicated connection, thus eliminating sender-side HoL blocking.

If response size unknown: server initiates migration after transmitting k bytes over the multiplexed connection.

SMig – Typical Usage Scenario

Client sends one or more HTTP/2 requests to server over a multiplexed connection.

If small and large objects are multiplexed together, server migrates large objects to separate connections.

Each large object uses a dedicated connection, thus eliminating sender-side HoL blocking.

If response size unknown: server initiates migration after transmitting k bytes over the multiplexed connection.

SMig – Typical Usage Scenario

Client sends one or more HTTP/2 requests to server over a multiplexed connection.

If small and large objects are multiplexed together, server migrates large objects to separate connections.

Each large object uses a dedicated connection, thus eliminating sender-side HoL blocking.

If response size unknown: server initiates migration after transmitting k bytes over the multiplexed connection.

SMig – Typical Usage Scenario

Client sends one or more HTTP/2 requests to server over a multiplexed connection.

If small and large objects are multiplexed together, server migrates large objects to separate connections.

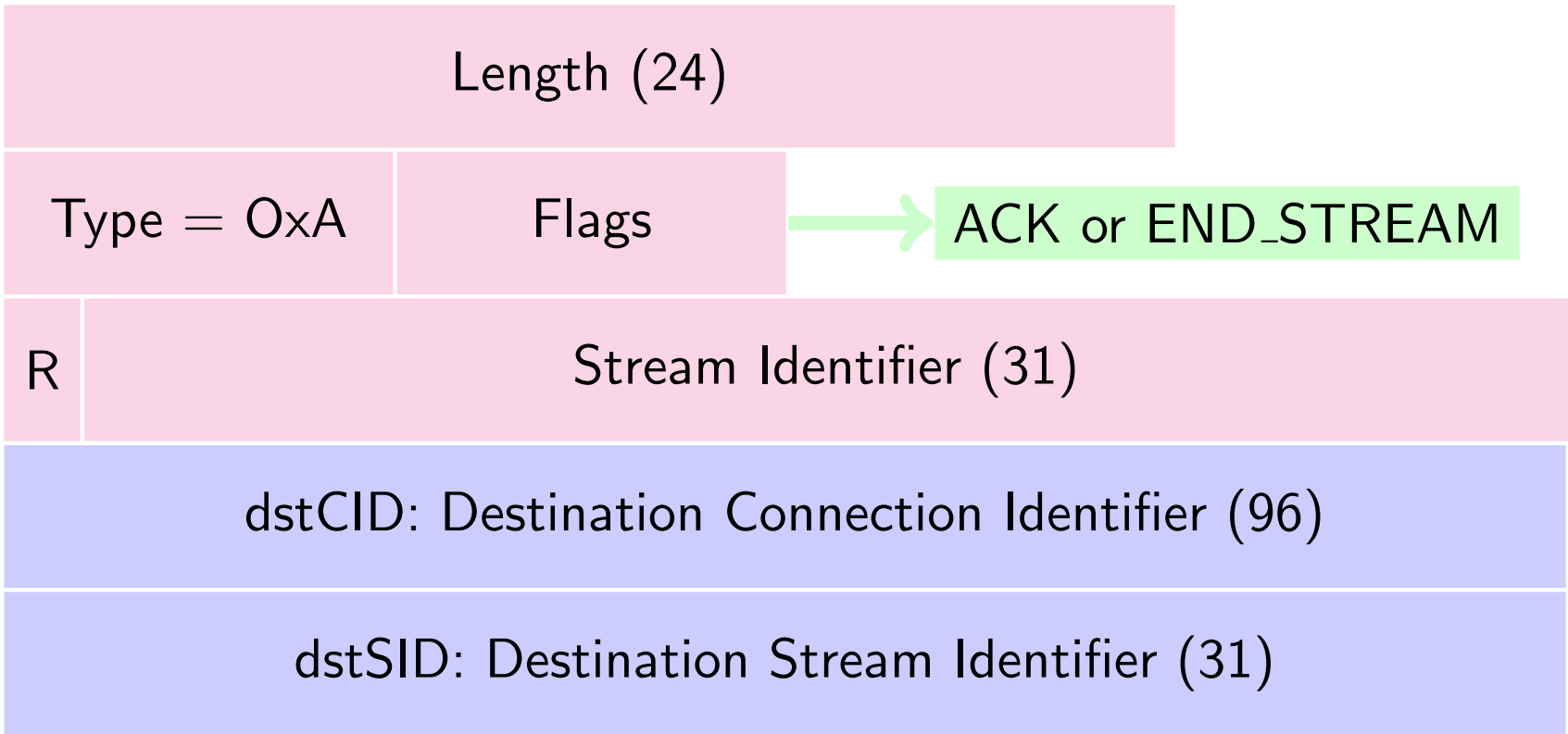
Each large object uses a dedicated connection, thus eliminating sender-side HoL blocking.

If response size unknown: server initiates migration after transmitting k bytes over the multiplexed connection.

Design of Smig: Migration Frame

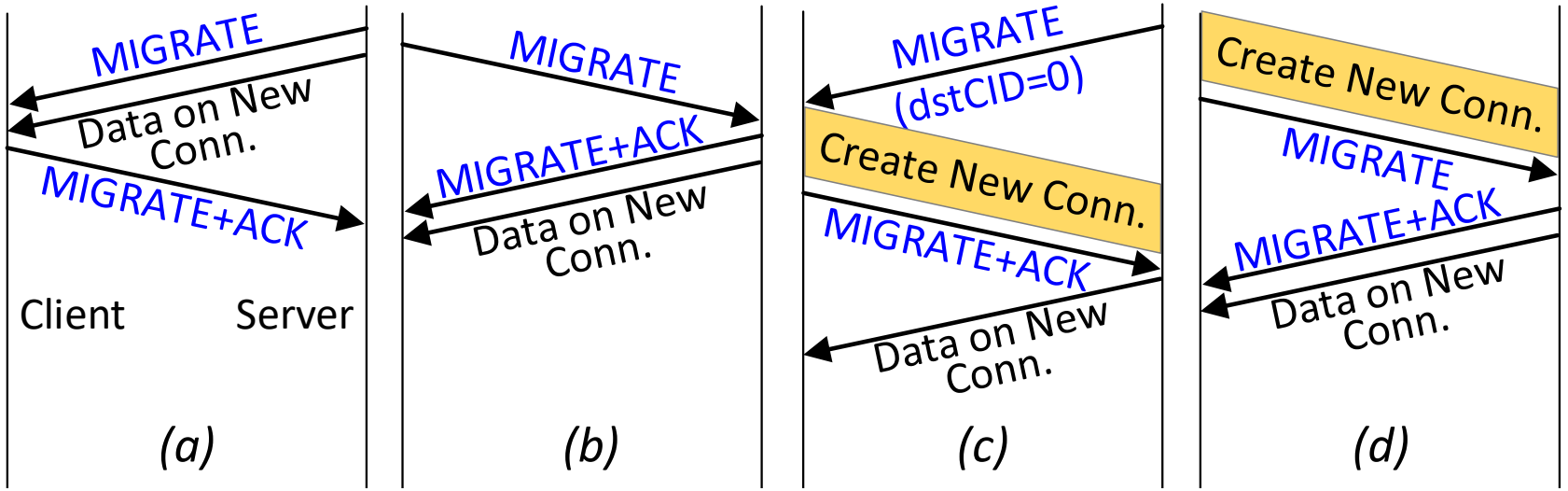
Migration Frame expresses the intent of initiating a stream migration.

The flags ensure correct cross-connection ordering of frames (details in the paper).



Design of Smig: Migration Scenarios

A migration can be initiated by either a client or server. If no idle connection exists, SMig will create a new one.



Initiated by server w/ idle conn.

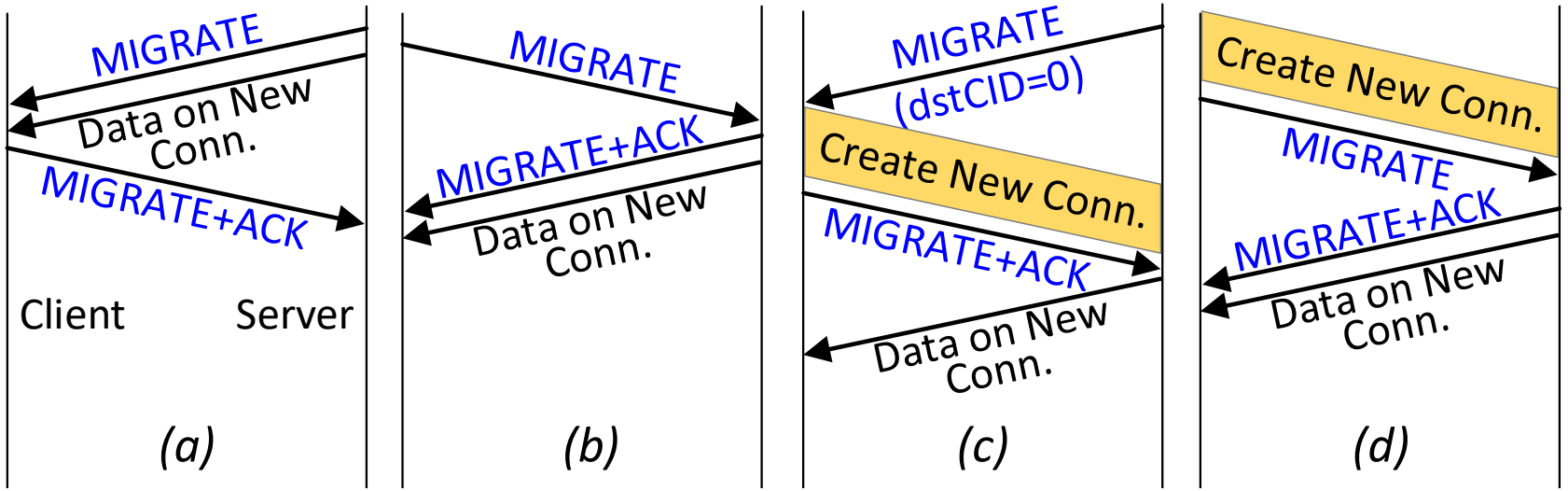
Initiated by client w/ idle conn.

Initiated by server w/o idle conn.

Initiated by client w/o idle conn.

Design of Smig: Migration Scenarios

A migration can be initiated by either a client or server. If no idle connection exists, SMig will create a new one.



Initiated by server w/ idle conn.

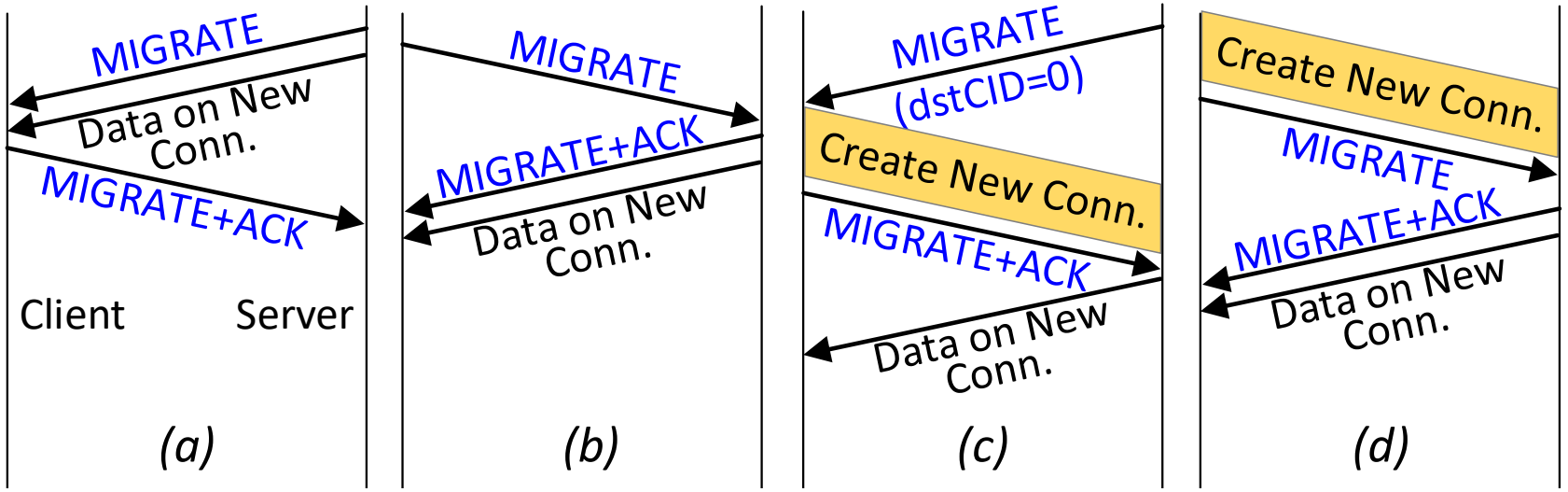
Initiated by client w/ idle conn.

Initiated by server w/o idle conn.

Initiated by client w/o idle conn.

Design of Smig: Migration Scenarios

A migration can be initiated by either a client or server. If no idle connection exists, SMig will create a new one.



Initiated by server w/ idle conn.

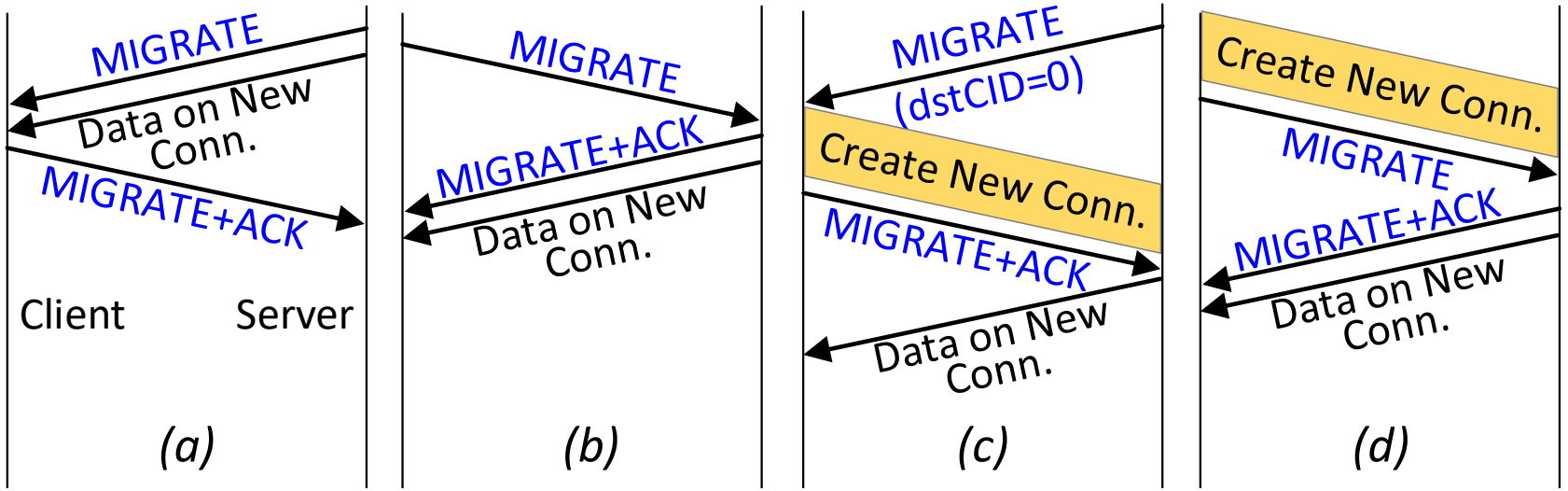
Initiated by client w/ idle conn.

Initiated by server w/o idle conn.

Initiated by client w/o idle conn.

Design of Smig: Migration Scenarios

A migration can be initiated by either a client or server. If no idle connection exists, SMig will create a new one.



Initiated by server w/ idle conn.	Initiated by client w/ idle conn.	Initiated by server w/o idle conn.	Initiated by client w/o idle conn.

Other Design Considerations (Details in the Paper)

SMig incurs low overhead for migration in common usage scenarios.

SMig strategically manages idle connections to strike a balance between resource usage and performance.

Various migration policies can be applied (examples shown soon).

SMig can work with HTTP/2 server push.

No new security vulnerability is introduced by SMig.

Implementation of SMig

Component	PL	LOC	OS Platforms
HTTP/2 Client and Server	C++	7.5K	Linux/OS X

Component	PL	LOC	OS Platforms
SMig extension	C++	1K	Linux/OS X

Evaluation of SMig: Experimental Setup

Client & Server Setting

Node	OS	CPU	Memory
Client	OS X 10.10	2.7GHz Intel Core i5 CPU	8GB
Server	Ubuntu 14.04	3GHz Intel Core2 Duo E8400 CPU	4GB

Network Setting

Type	Network Type
Wired	An emulated 10Mbps link with 50ms RTT
Cellular	A commercial LTE network

Evaluation of SMig: Experimental Setup

Client & Server Setting

Node	OS	CPU	Memory
Client	OS X 10.10	2.7GHz Intel Core i5 CPU	8GB
Server	Ubuntu 14.04	3GHz Intel Core2 Duo E8400 CPU	4GB

Network Setting

Type	Network Type
Wired	An emulated 10Mbps link with 50ms RTT
Cellular	A commercial LTE network

Evaluation Methodology

Workload: concurrent small & large file downloads (10 KB vs. 50 MB) in four scenarios. SMig migrates the large file.

NoMig: SMig is disabled

MigSW: server initiates the migration for the large file once it receives its request.

MigSP: server initiates the migration after sending 100KB response data (for chunked mode encoding).

MigCP: client initiates the migration once it receives the response header.

Evaluation: Small File Download Time over Wired Network

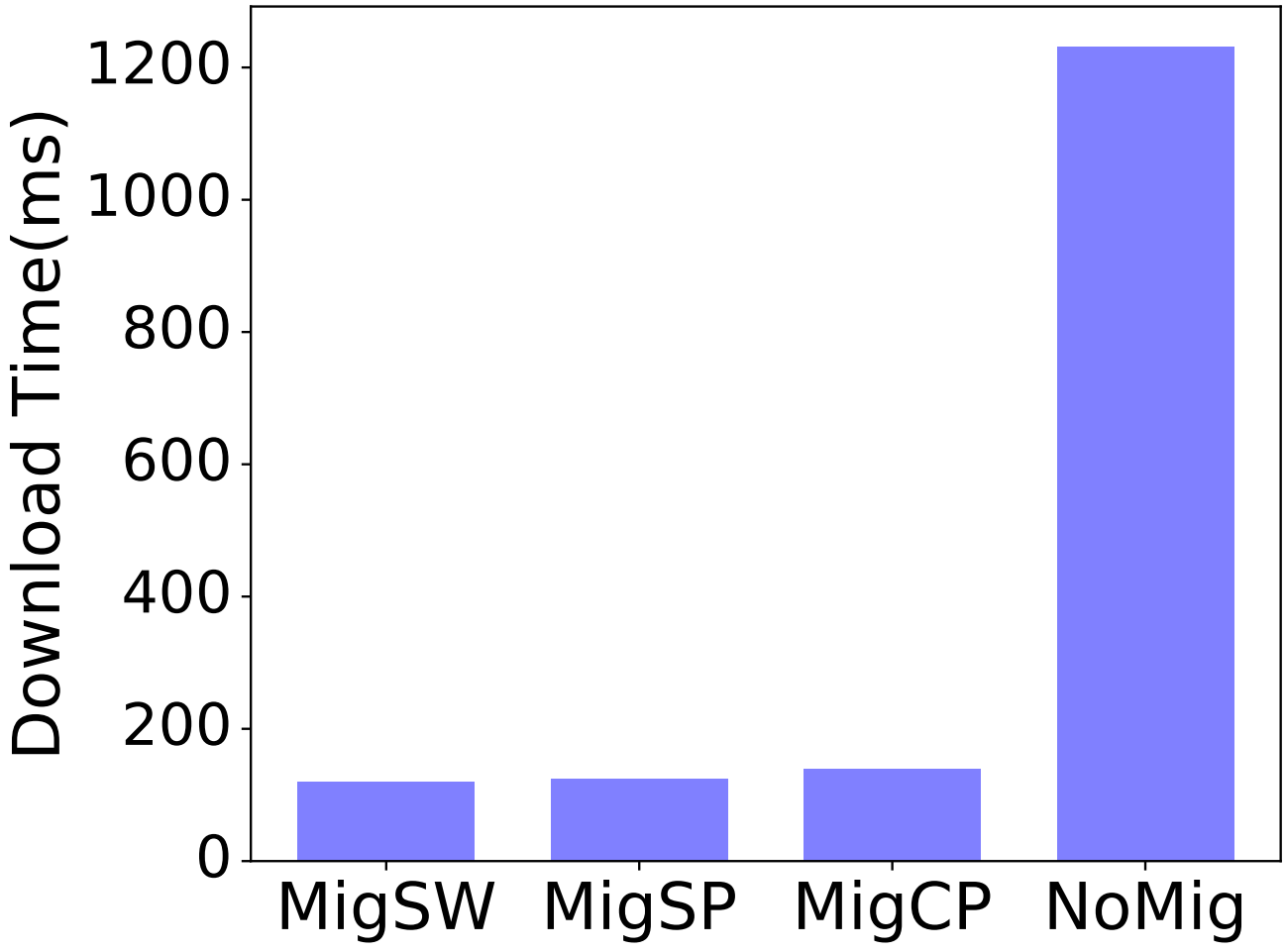


Figure: SMig's Impact on Small File Download (Wired)

Evaluation: Small File Download Time over LTE

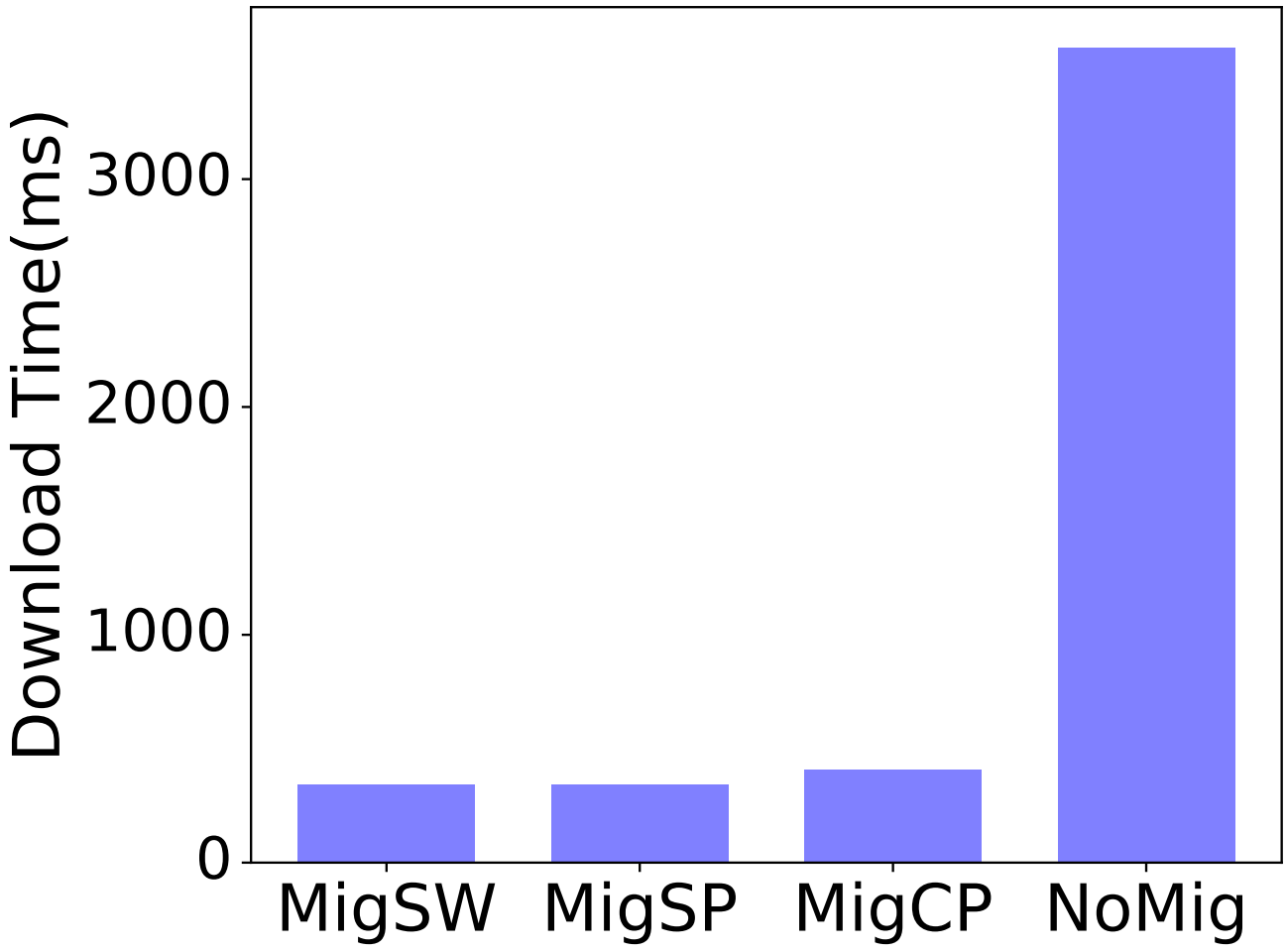


Figure: SMig's Impact on Small File Download (Cellular)

Evaluation: Impact of Migration on Large File Download Time

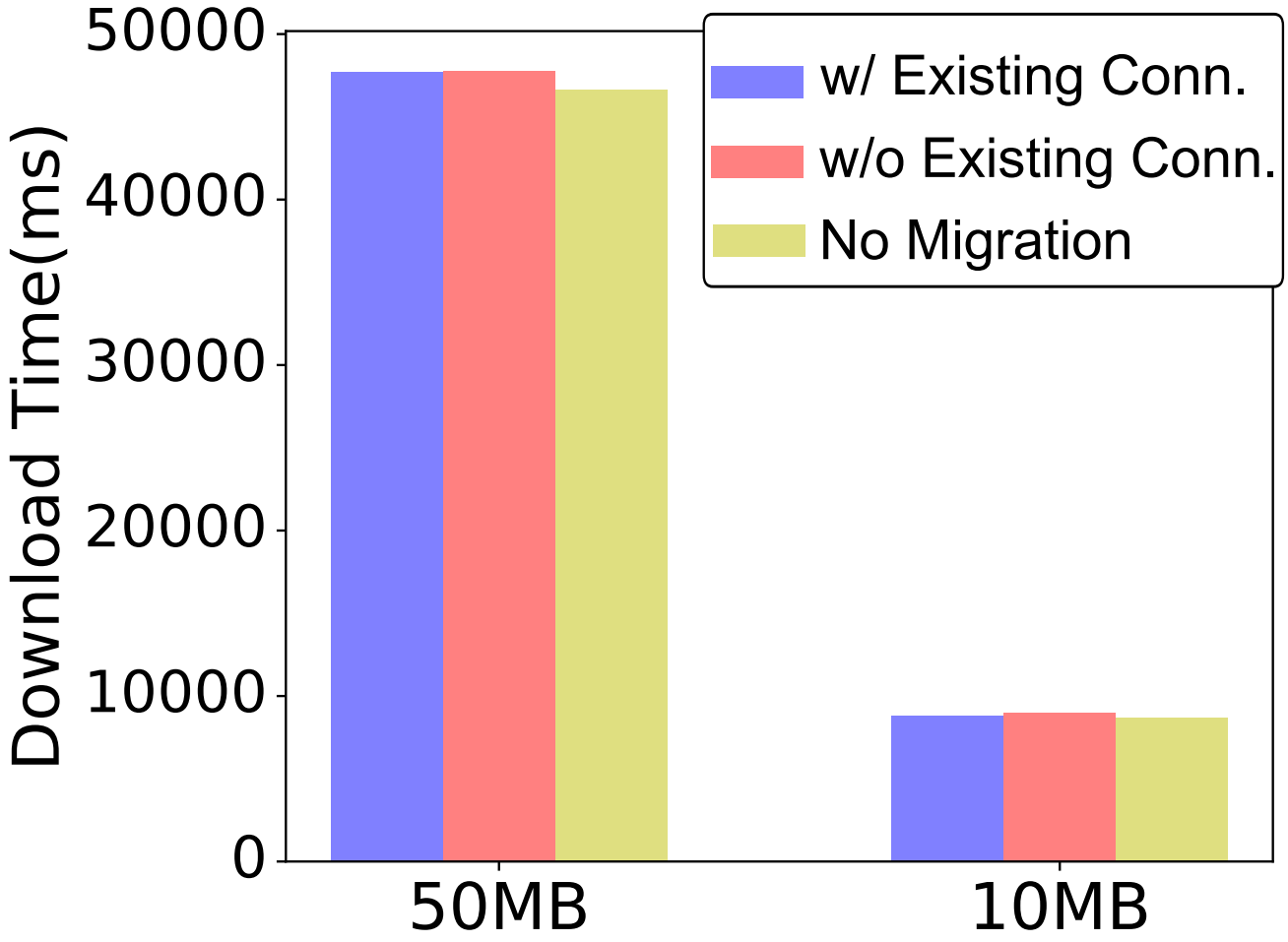


Figure: SMig's Impact on Large File Download(Wired)

Summary

SMig: an HTTP/2 extension allowing a client or server to migrate an on-going HTTP/2 stream from one connection to another.

SMig eliminates sender-side HoLB. It reduces the delay-sensitive file download time by up to 99% when concurrent transfers occur.

SMig brings other benefits and usage scenarios (see the paper for details).

An IETF Draft of SMig is working-in-progress!

Summary

SMig: an HTTP/2 extension allowing a client or server to migrate an on-going HTTP/2 stream from one connection to another.

SMig eliminates sender-side HoLB. It reduces the delay-sensitive file download time by up to 99% when concurrent transfers occur.

SMig brings other benefits and usage scenarios (see the paper for details).

An IETF Draft of SMig is working-in-progress!

Summary

SMig: an HTTP/2 extension allowing a client or server to migrate an on-going HTTP/2 stream from one connection to another.

SMig eliminates sender-side HoLB. It reduces the delay-sensitive file download time by up to 99% when concurrent transfers occur.

SMig brings other benefits and usage scenarios (see the paper for details).

An IETF Draft of SMig is working-in-progress!

Summary

SMig: an HTTP/2 extension allowing a client or server to migrate an on-going HTTP/2 stream from one connection to another.

SMig eliminates sender-side HoLB. It reduces the delay-sensitive file download time by up to 99% when concurrent transfers occur.

SMig brings other benefits and usage scenarios (see the paper for details).

An IETF Draft of SMig is working-in-progress!

Q&A

Contact: fengqian@indiana.edu