
AWS
re:Invent

NET311

How AWS Global Accelerator improves performance

Dennis Marinus

Senior Systems Development Engineer

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Up front: 300 level – advanced technical



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Learning objectives



Understand how networking impacts application performance

Learning objectives



Understand how networking impacts application performance



Learn how AWS Global Accelerator improves performance and availability

Learning objectives



Understand how networking impacts application performance



Learn how AWS Global Accelerator improves performance and availability



Ways that you can mitigate network challenges yourself

AWS Global Accelerator

A networking service that improves availability
and performance of applications with global reach



Key features



Static IP addresses as fixed entry points to apps in single or multiple AWS Regions

Key features



Static IP addresses as fixed entry points to apps in single or multiple AWS Regions



Improved performance for TCP and UDP traffic targeting ALBs, NLBs, or Amazon EC2 instances

Key features



Static IP addresses as fixed entry points to apps in single or multiple AWS Regions



Improved performance for TCP and UDP traffic targeting ALBs, NLBs, or Amazon EC2 instances



Traffic failover in less than 30 seconds from detection of unhealthy endpoint

Key features



Static IP addresses as fixed entry points to apps in single or multiple AWS Regions



Improved performance for TCP and UDP traffic targeting ALBs, NLBs, or Amazon EC2 instances



Traffic failover in less than 30 seconds from detection of unhealthy endpoint



Move endpoints between Availability Zones and Regions with no change to clients or Domain Name System (DNS) configuration

Key features



Static IP addresses as fixed entry points to apps in single or multiple AWS Regions



Improved performance for TCP and UDP traffic targeting ALBs, NLBs, or Amazon EC2 instances



Traffic failover in less than 30 seconds from detection of unhealthy endpoint



Move endpoints between Availability Zones and Regions with no change to clients or Domain Name System (DNS) configuration

All traffic traverses the backbone* and is protected from DDoS attacks

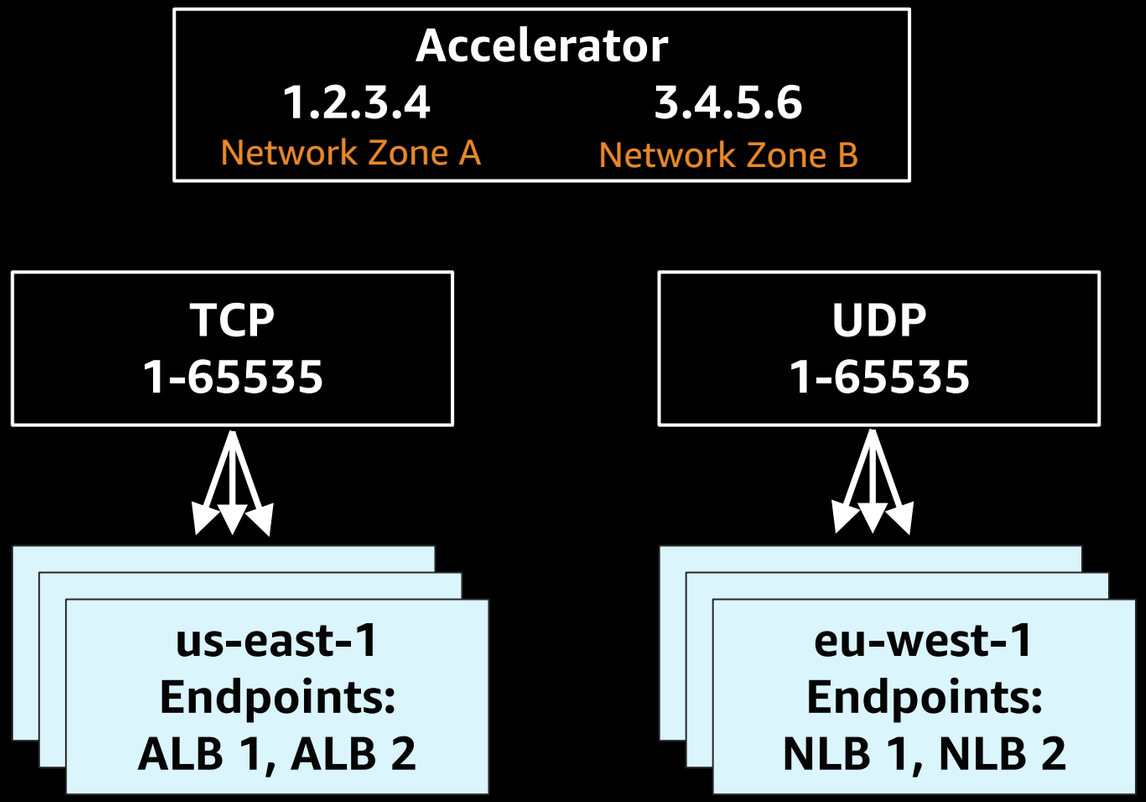
*except within the People's Republic of China

Accelerator configuration

Accelerator IPs

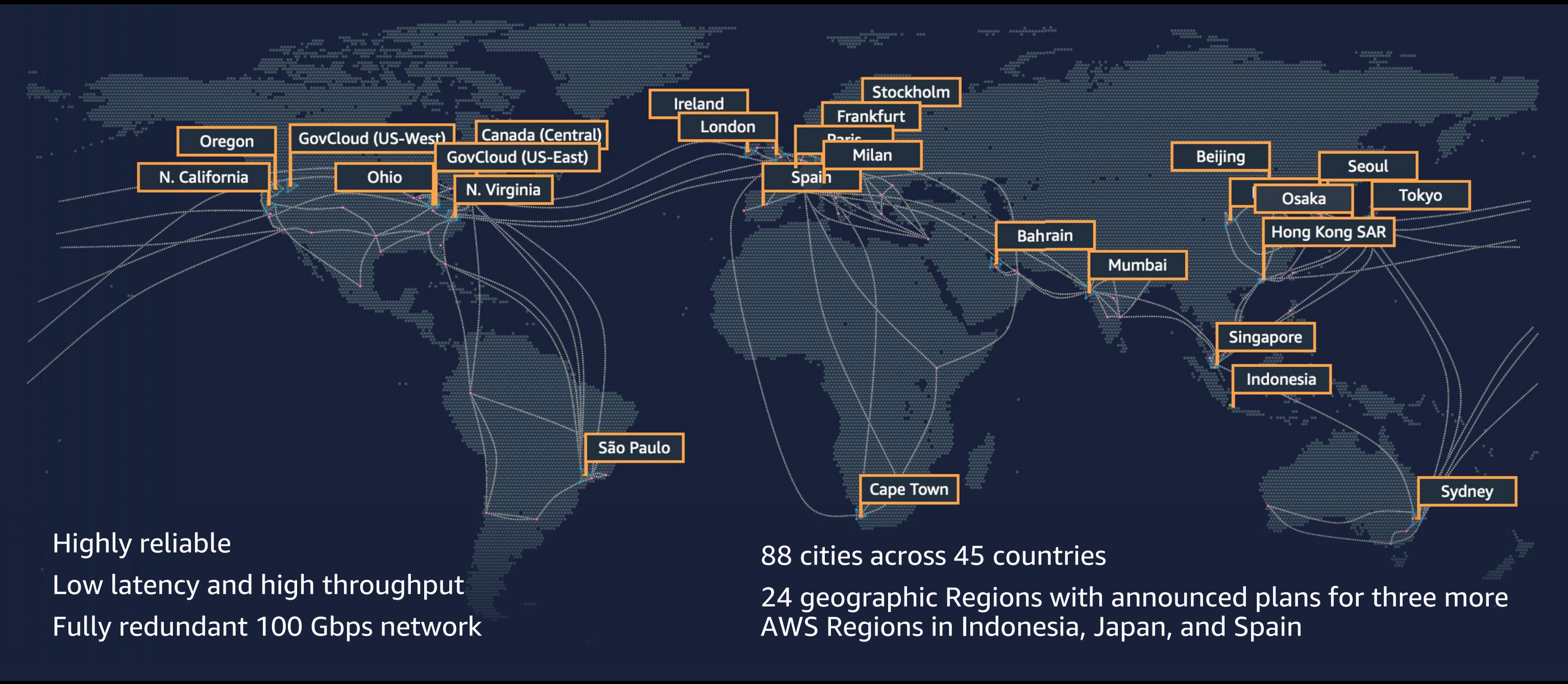
Listeners

Endpoint groups



AWS Global Cloud Infrastructure

infrastructure.aws



Highly reliable

Low latency and high throughput

Fully redundant 100 Gbps network

88 cities across 45 countries

24 geographic Regions with announced plans for three more AWS Regions in Indonesia, Japan, and Spain

Our story



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How our story starts



How our story starts





Amazon CloudWatch

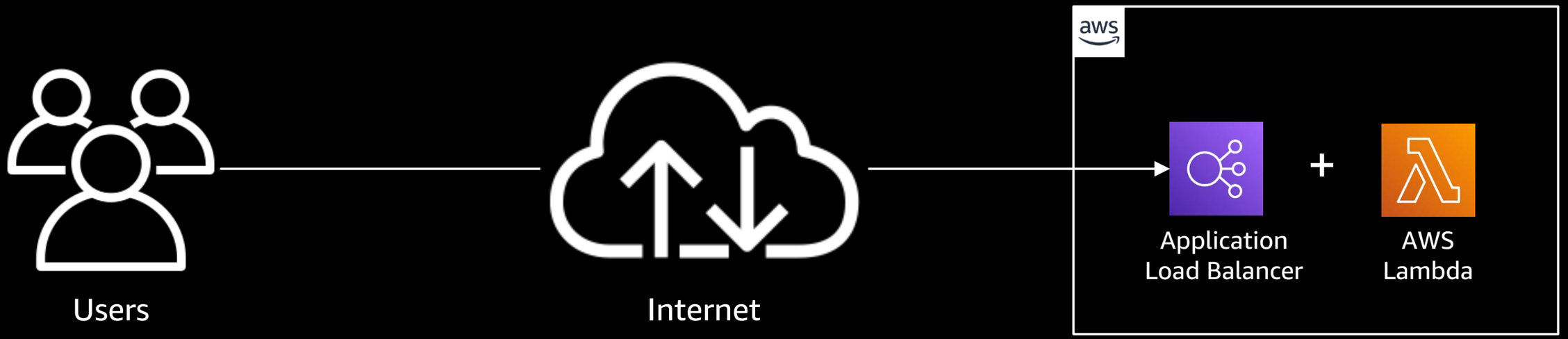
How our story starts



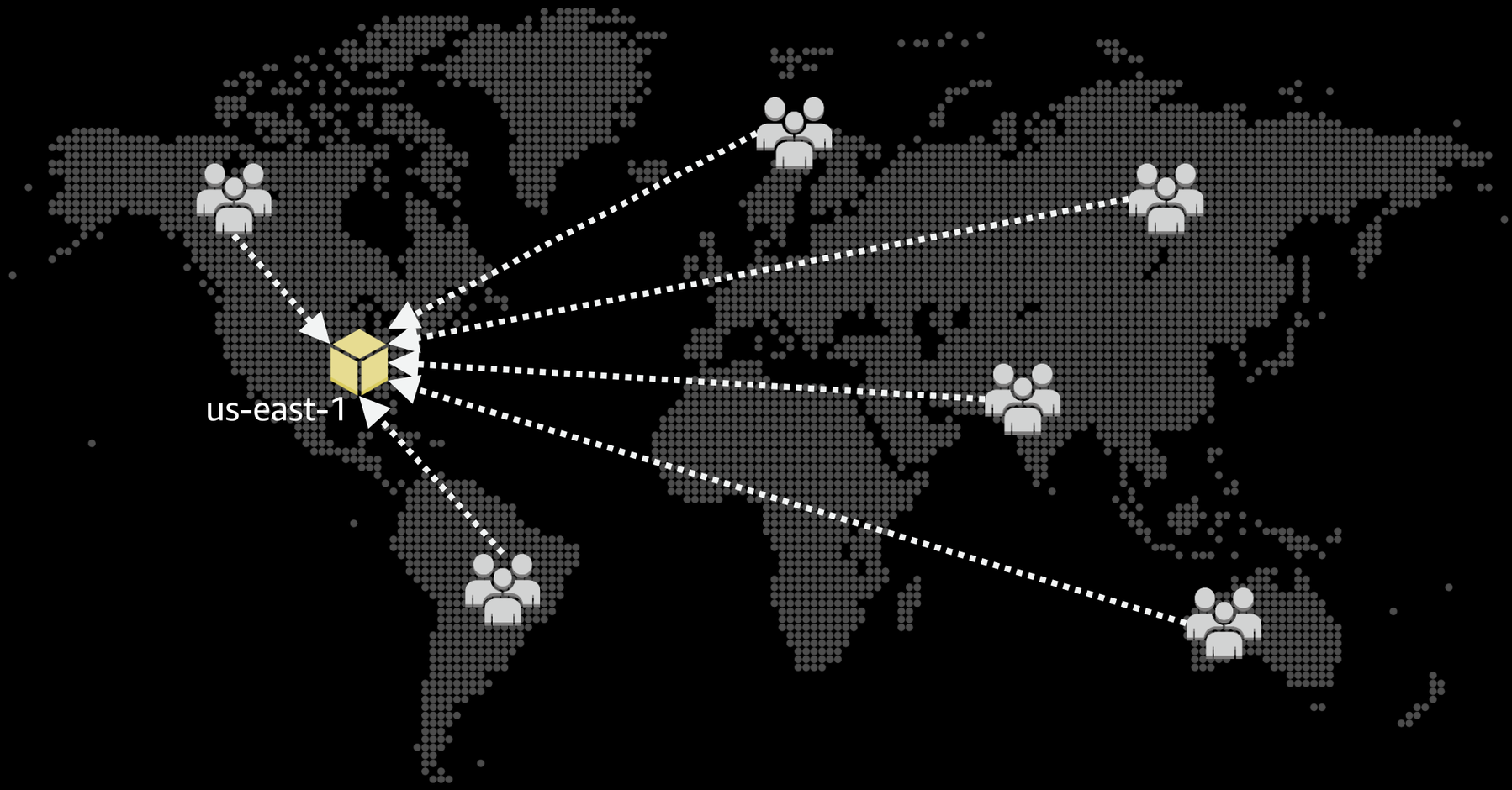


“I love your app, but it loads so slow. Can you please make it faster?”
– Rufus

Our simple app architecture



A simple view of the internet



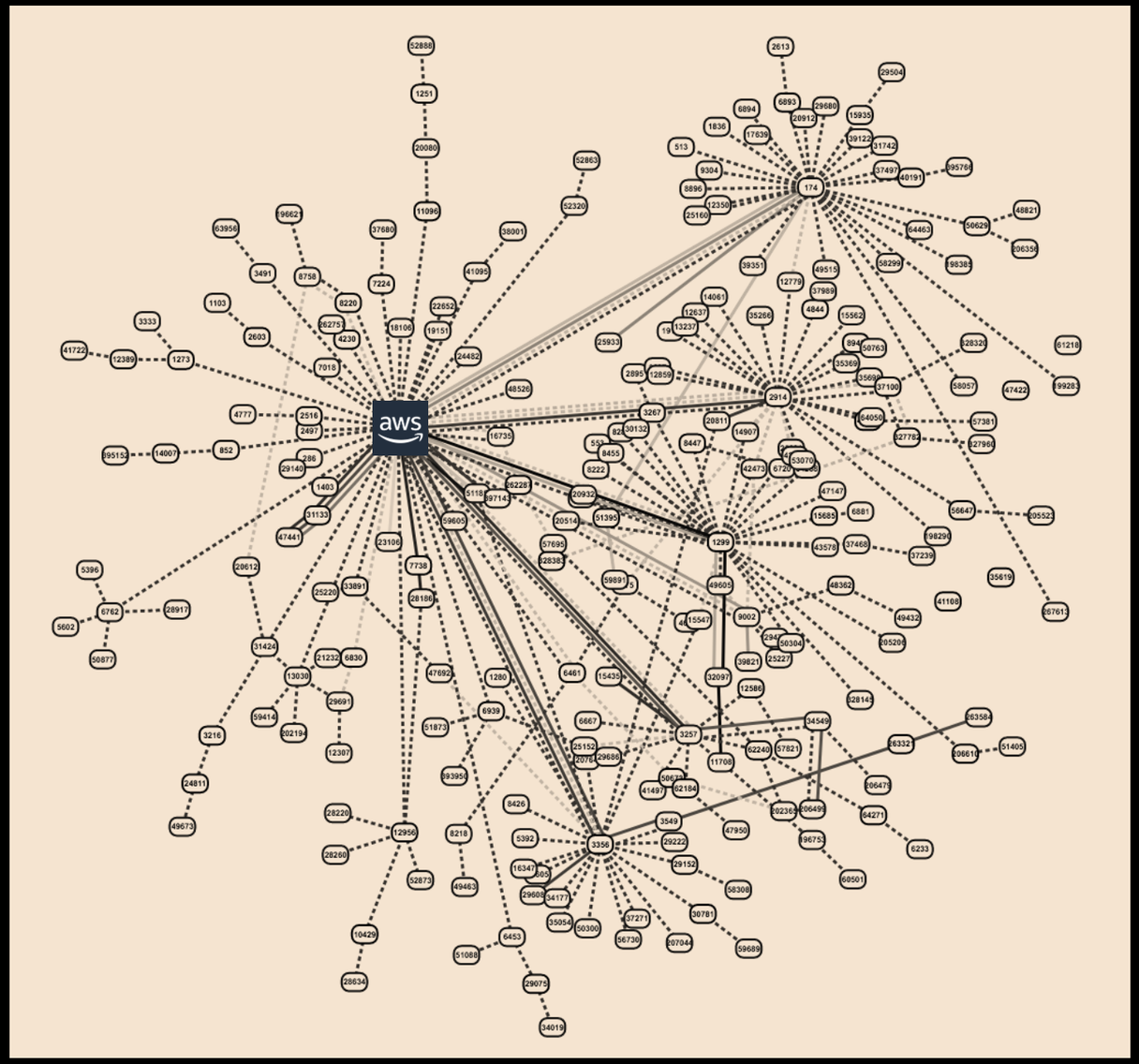
The real internet



Internet view of an AWS Region

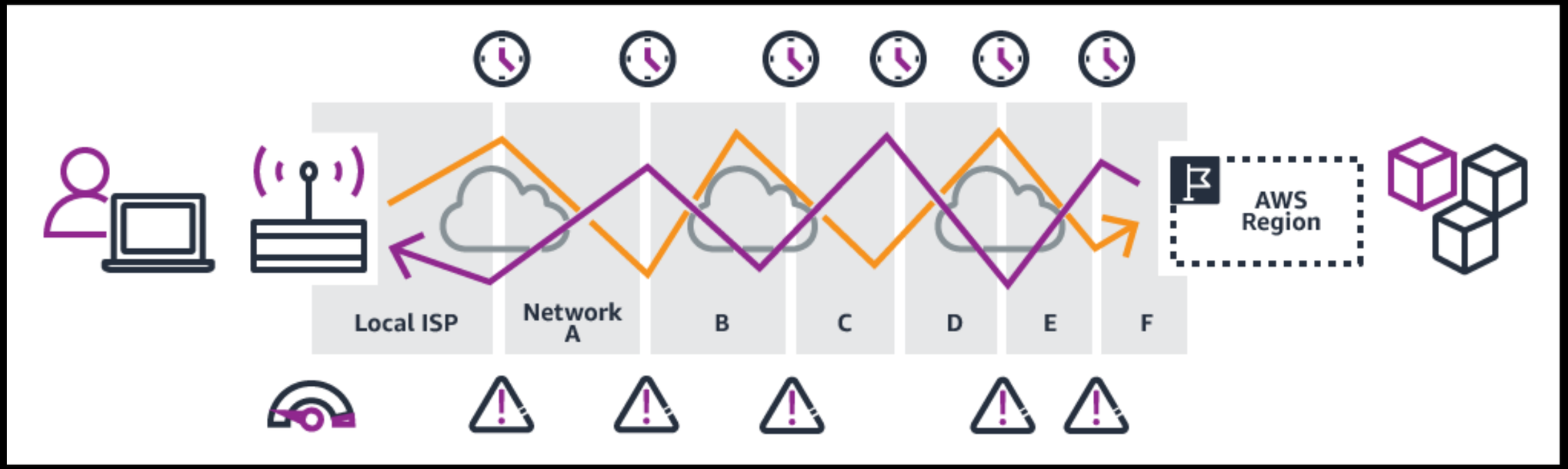
The internet consists of

- About 70 thousand networks
- About 850 thousand prefixes

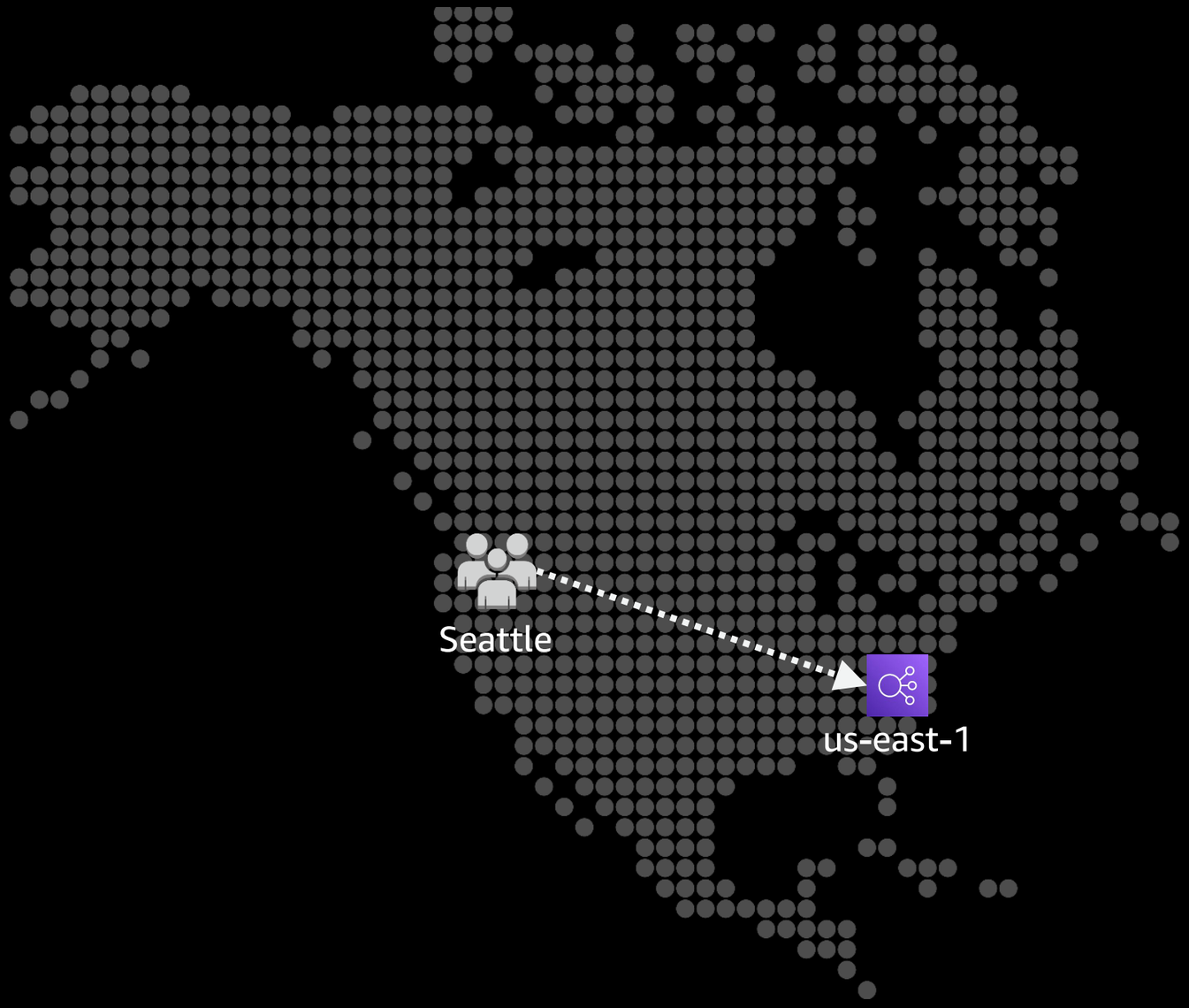


How distance affects performance





How distance affects performance



Data transfer



Users

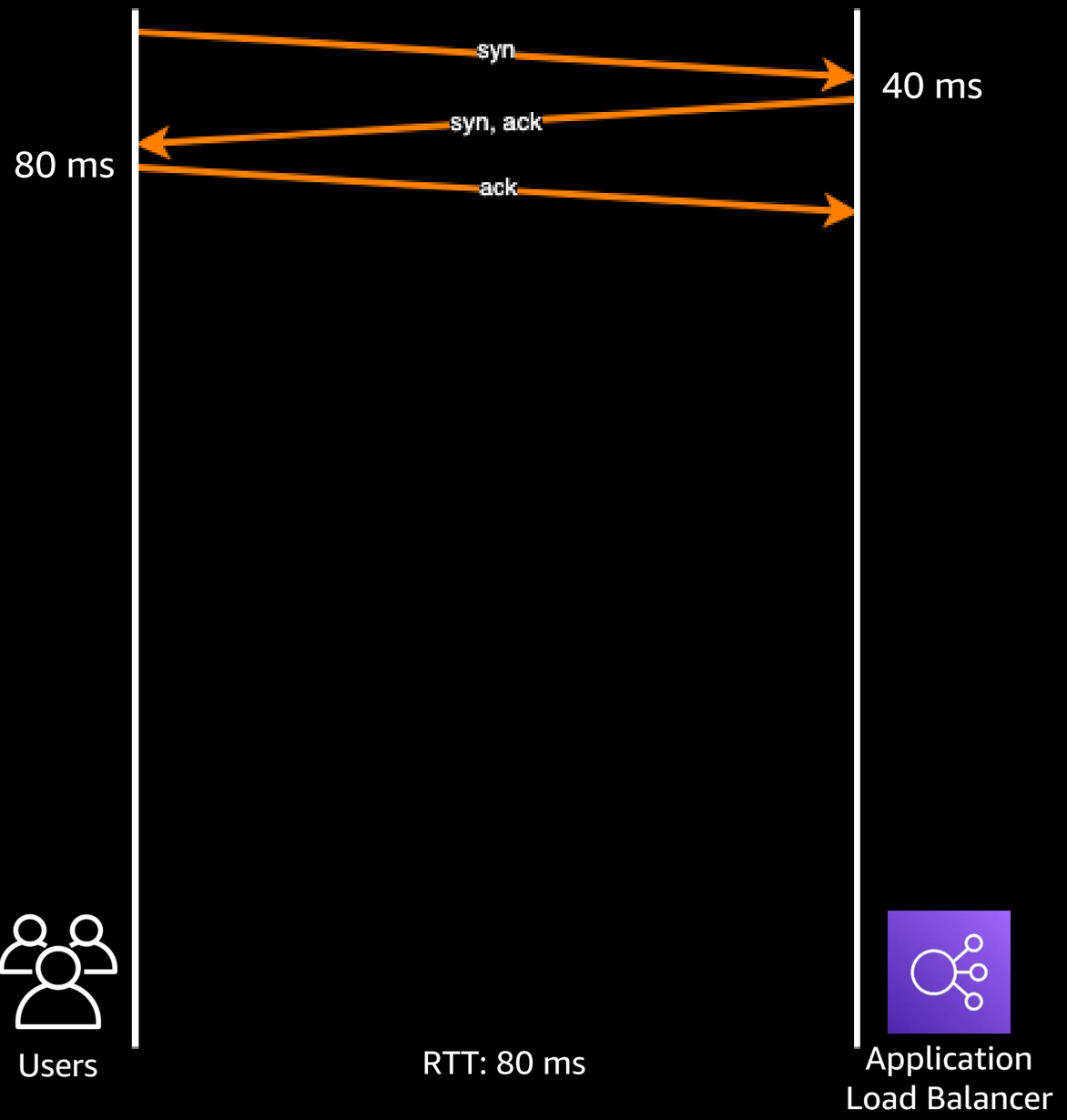
RTT: 80 ms



Application
Load Balancer

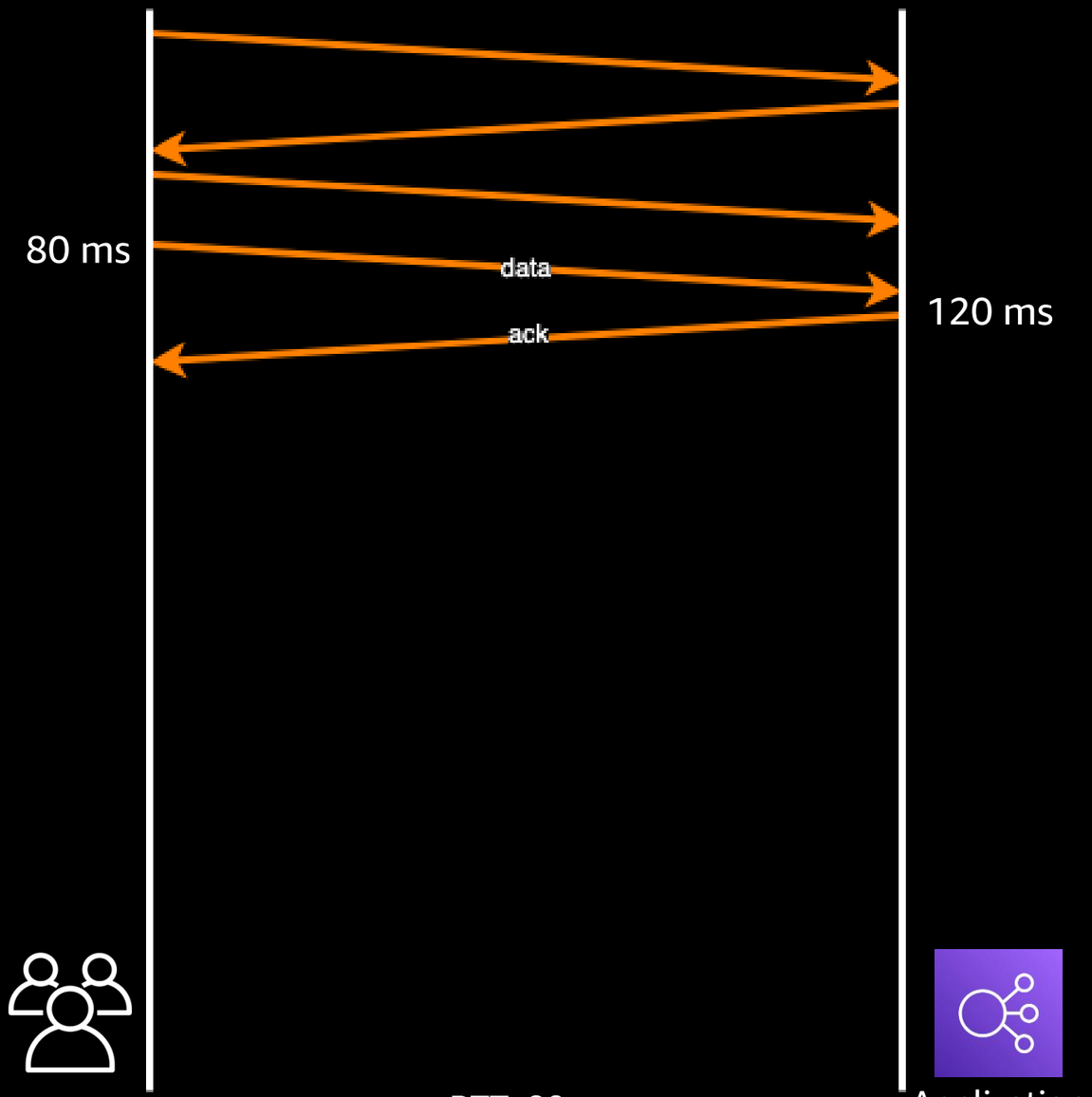
Data transfer

Handshake

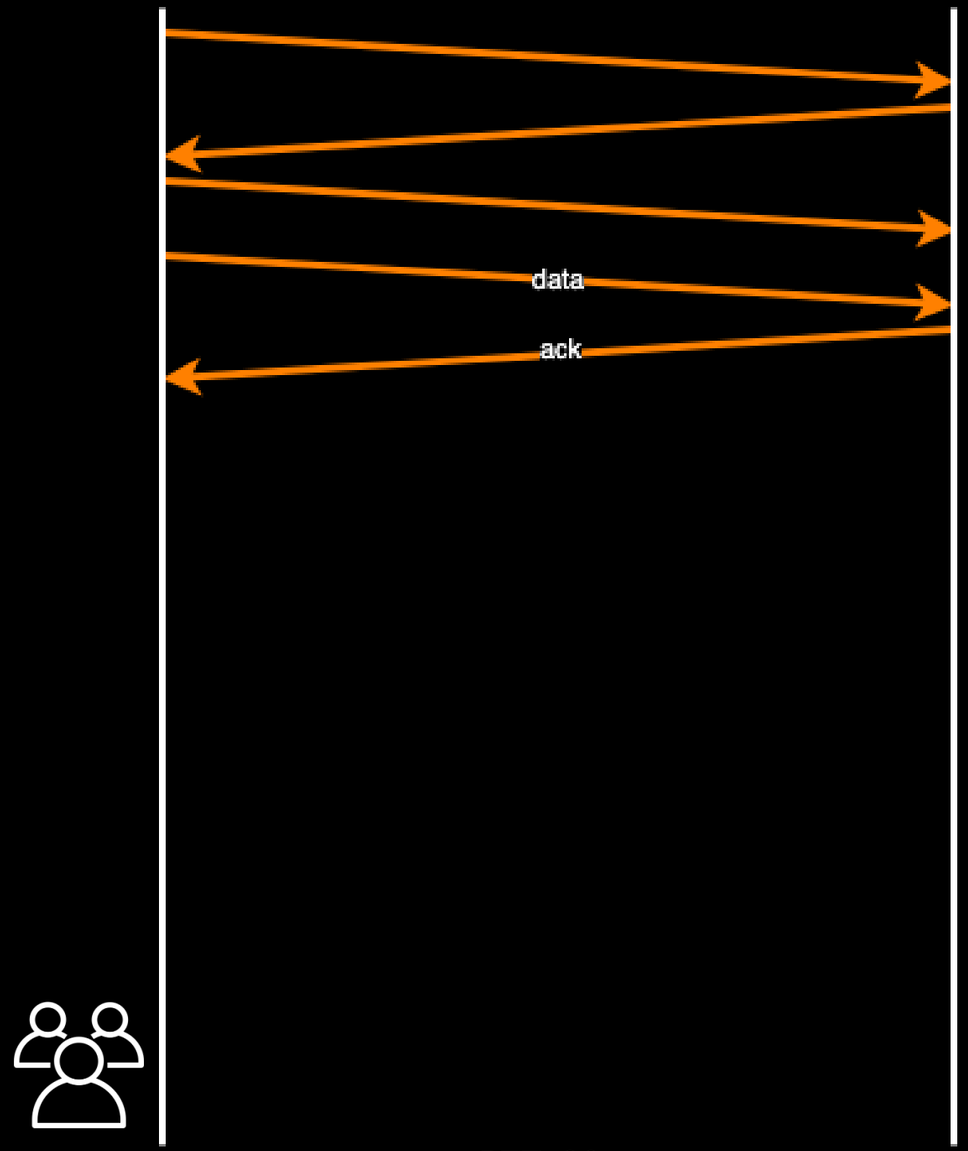


Data transfer

Request



Data transfer



Your code runs

Users

RTT: 80 ms

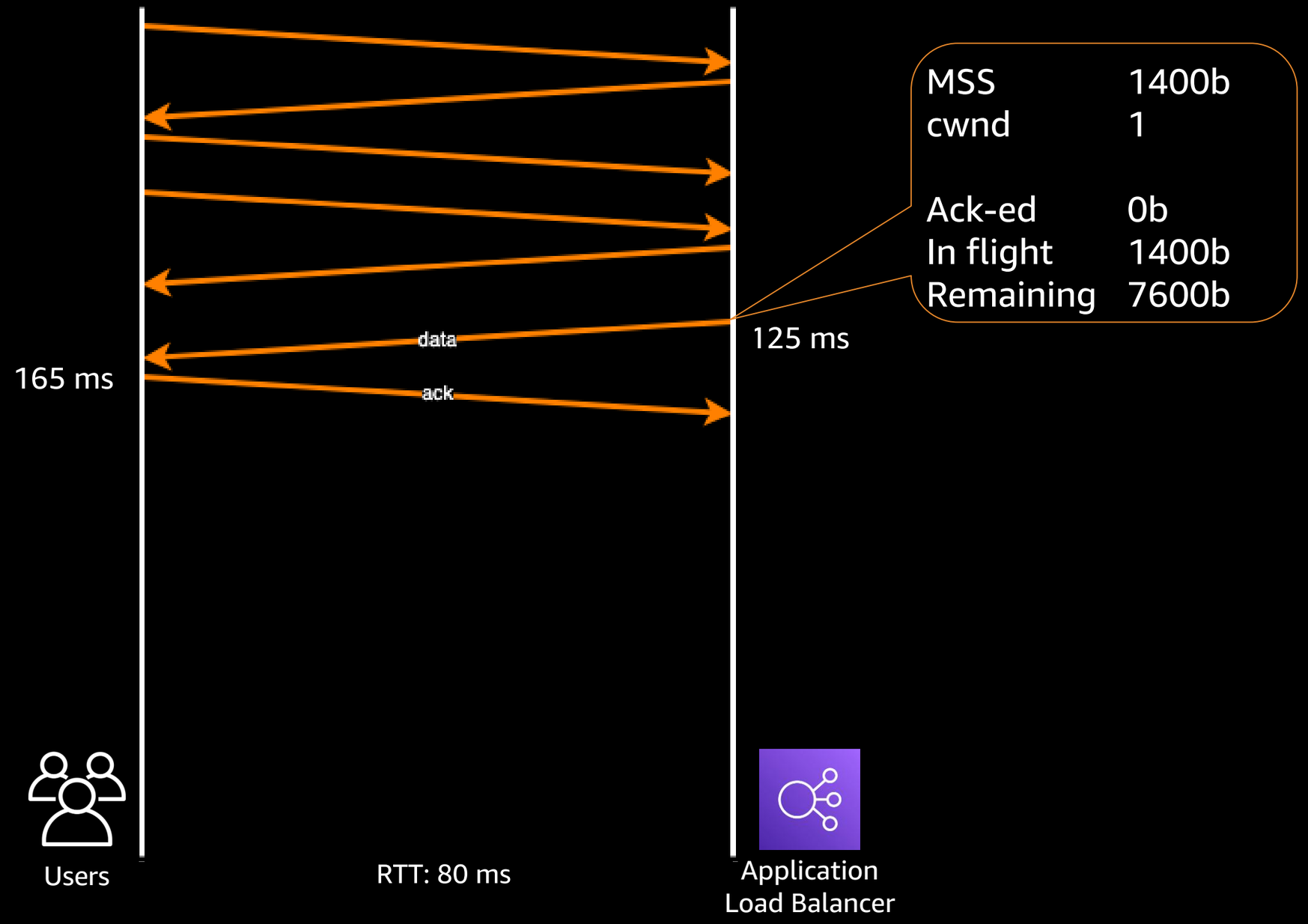
Application Load Balancer

+

Lambda

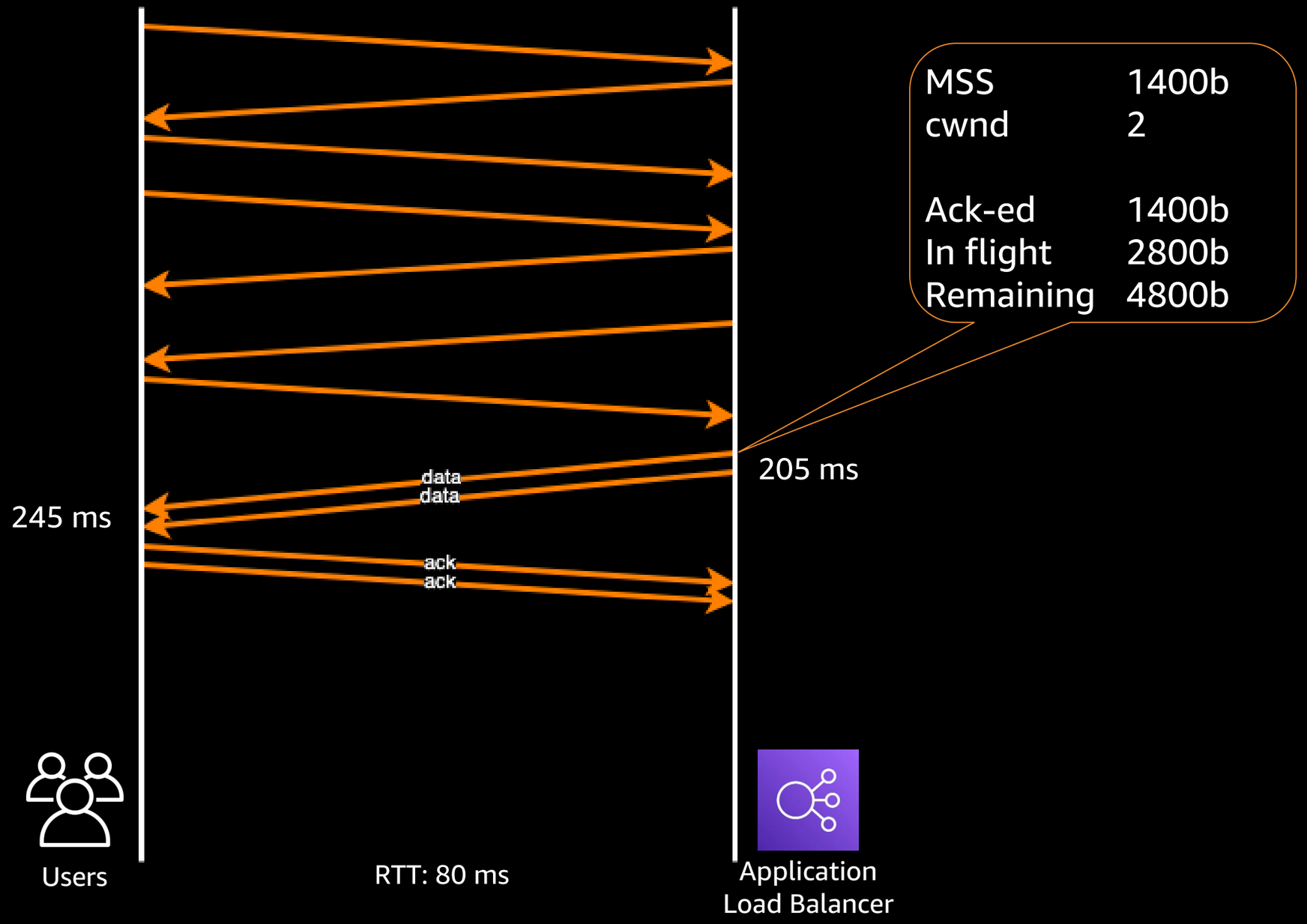
Data transfer

Response



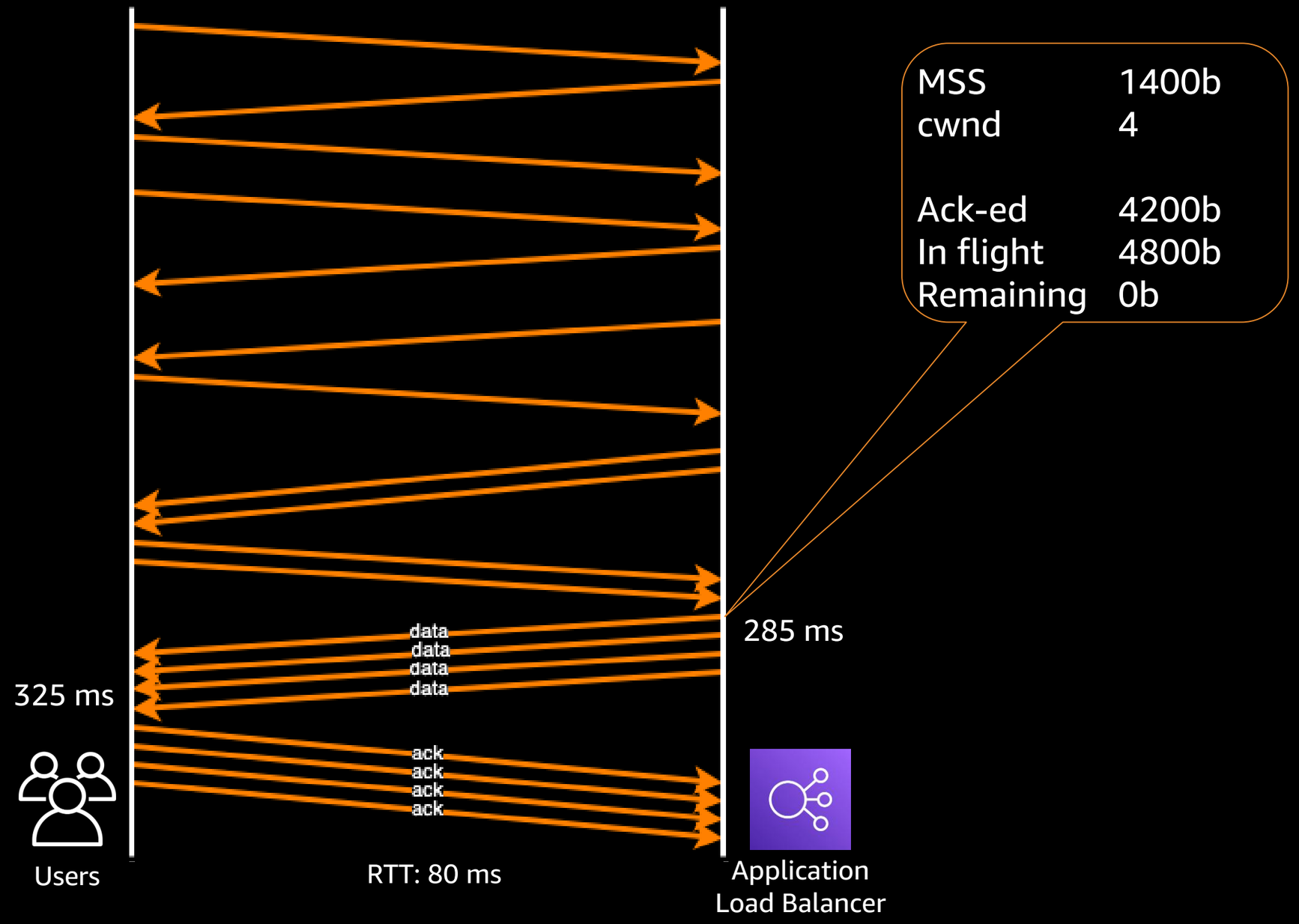
Data transfer

Response



Data transfer

Response



Total time spent

Your code: 5 ms

Network latency: 320 ms

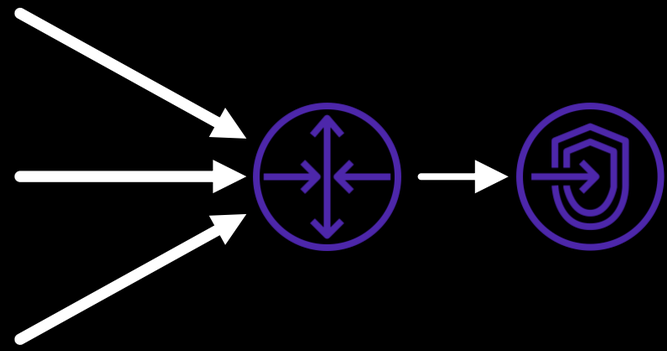
Total: 325 ms

TCP congestion control – The good and the bad

Goals

Use all available bandwidth

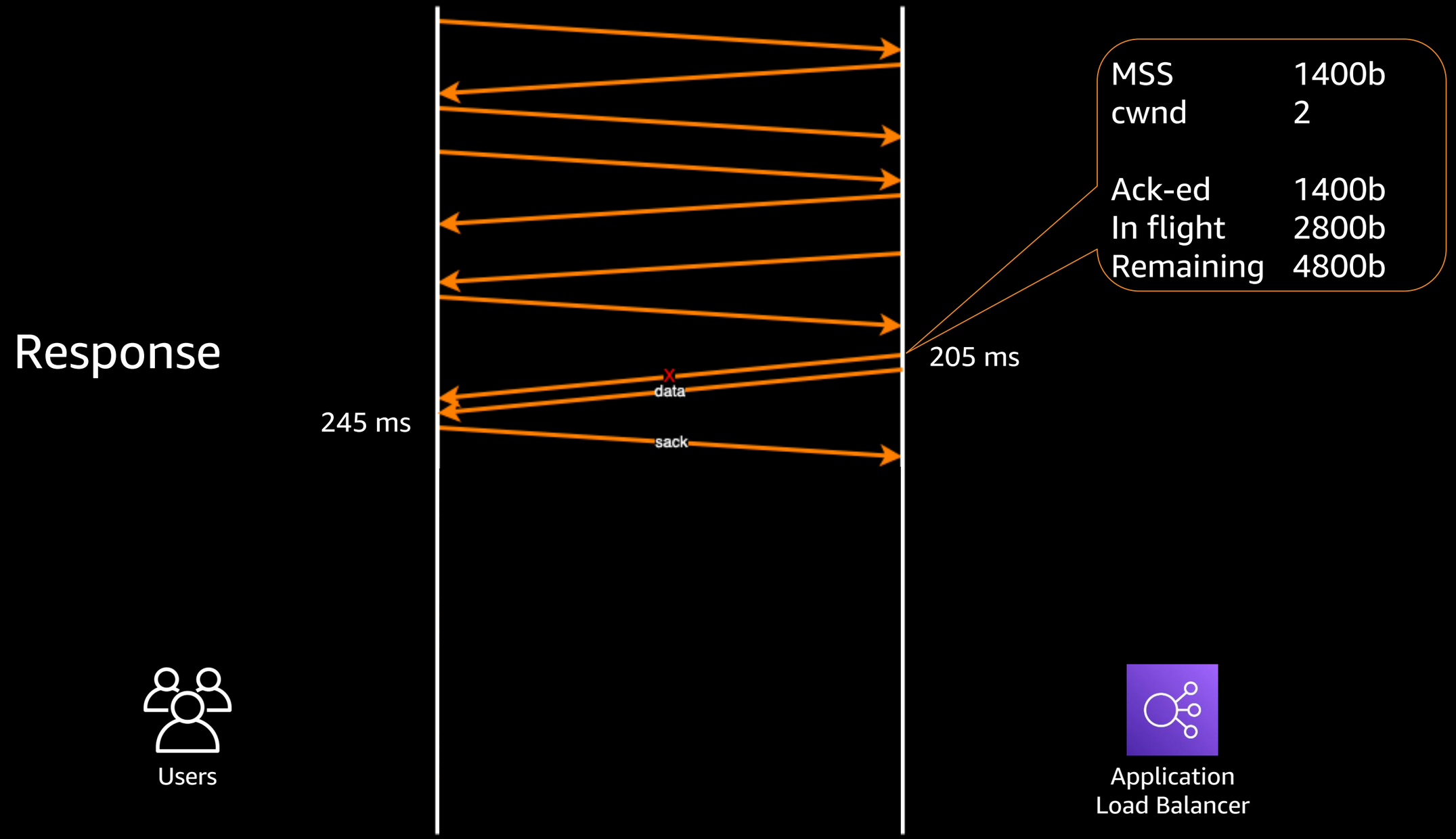
- Without saturating network
- While being fair to others



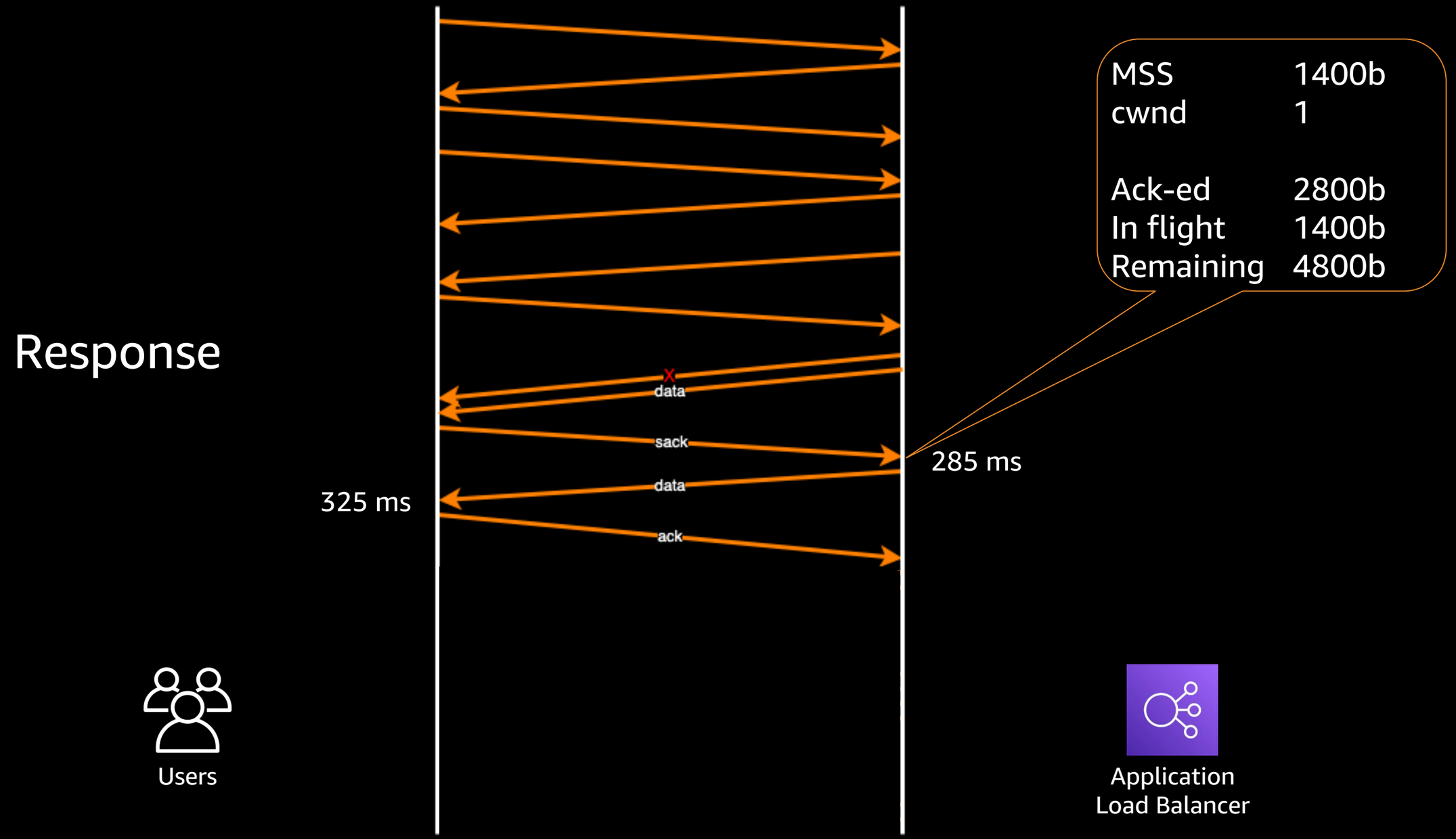
Downsides

- Number of hops increases chance of loss
- Any loss can have serious impact to performance

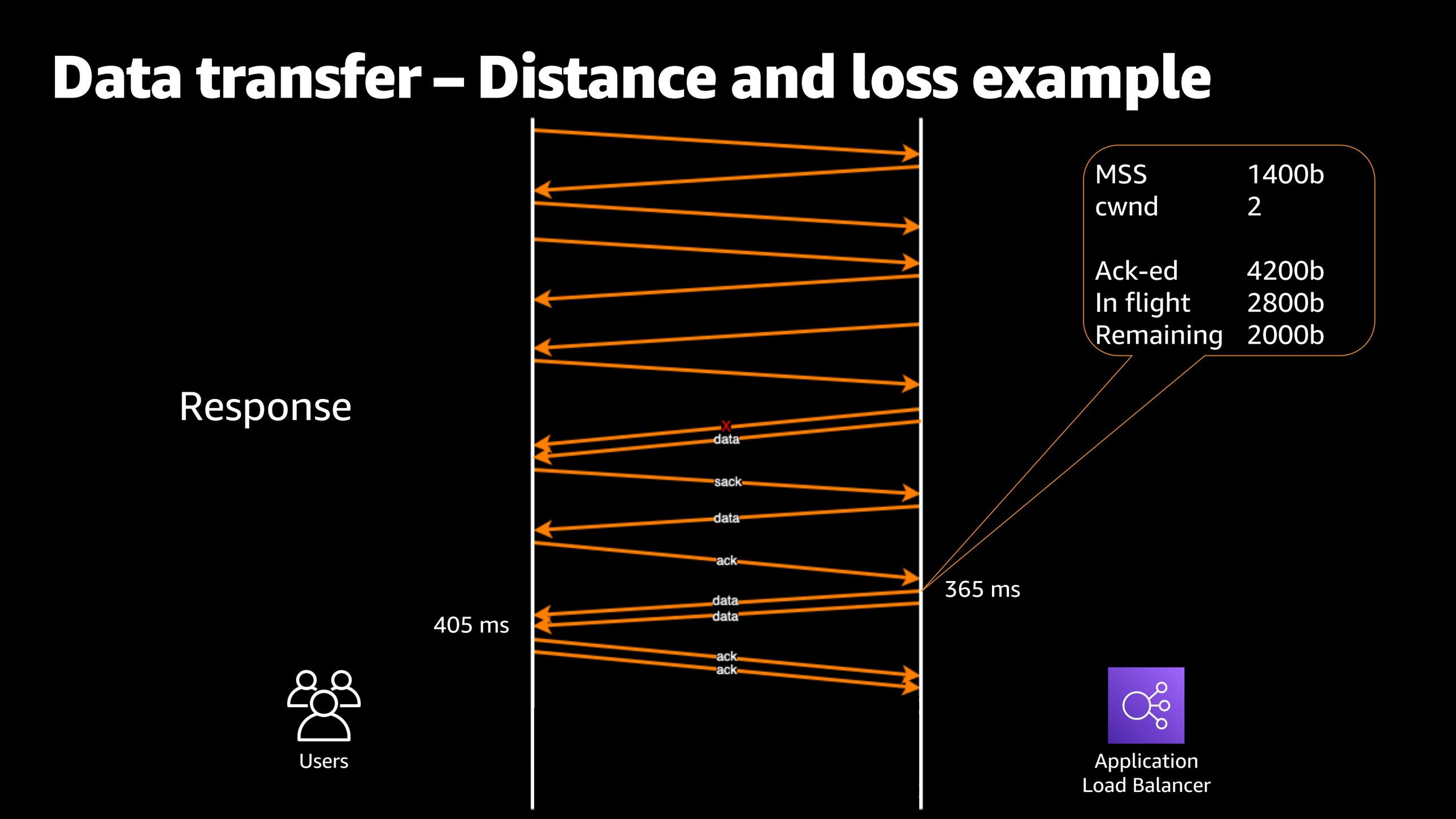
Data transfer – Distance and loss example



Data transfer – Distance and loss example

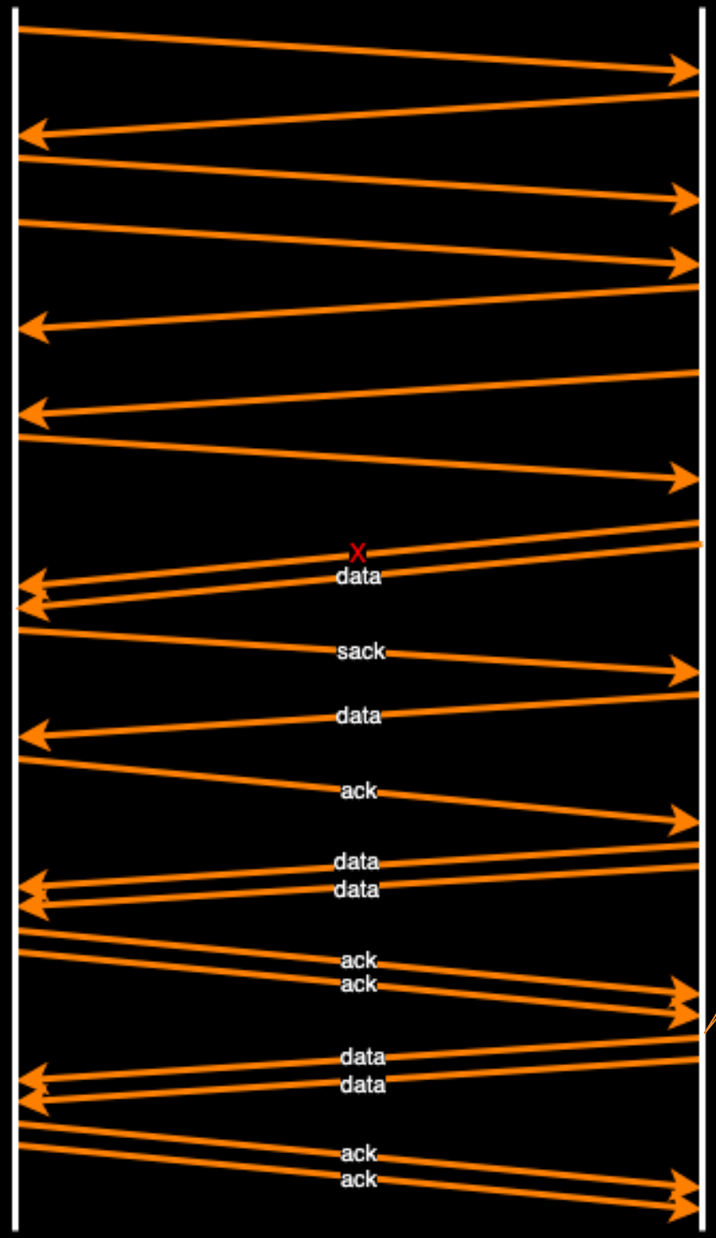


Data transfer – Distance and loss example



Data transfer – Distance and loss example

Response



485 ms

445 ms

MSS	1400b
cwnd	4
Ack-ed	7000b
In flight	2000b
Remaining	0b

Users

Application
Load Balancer

Total time spent comparison

Without packet loss

325 ms

With packet loss

485 ms

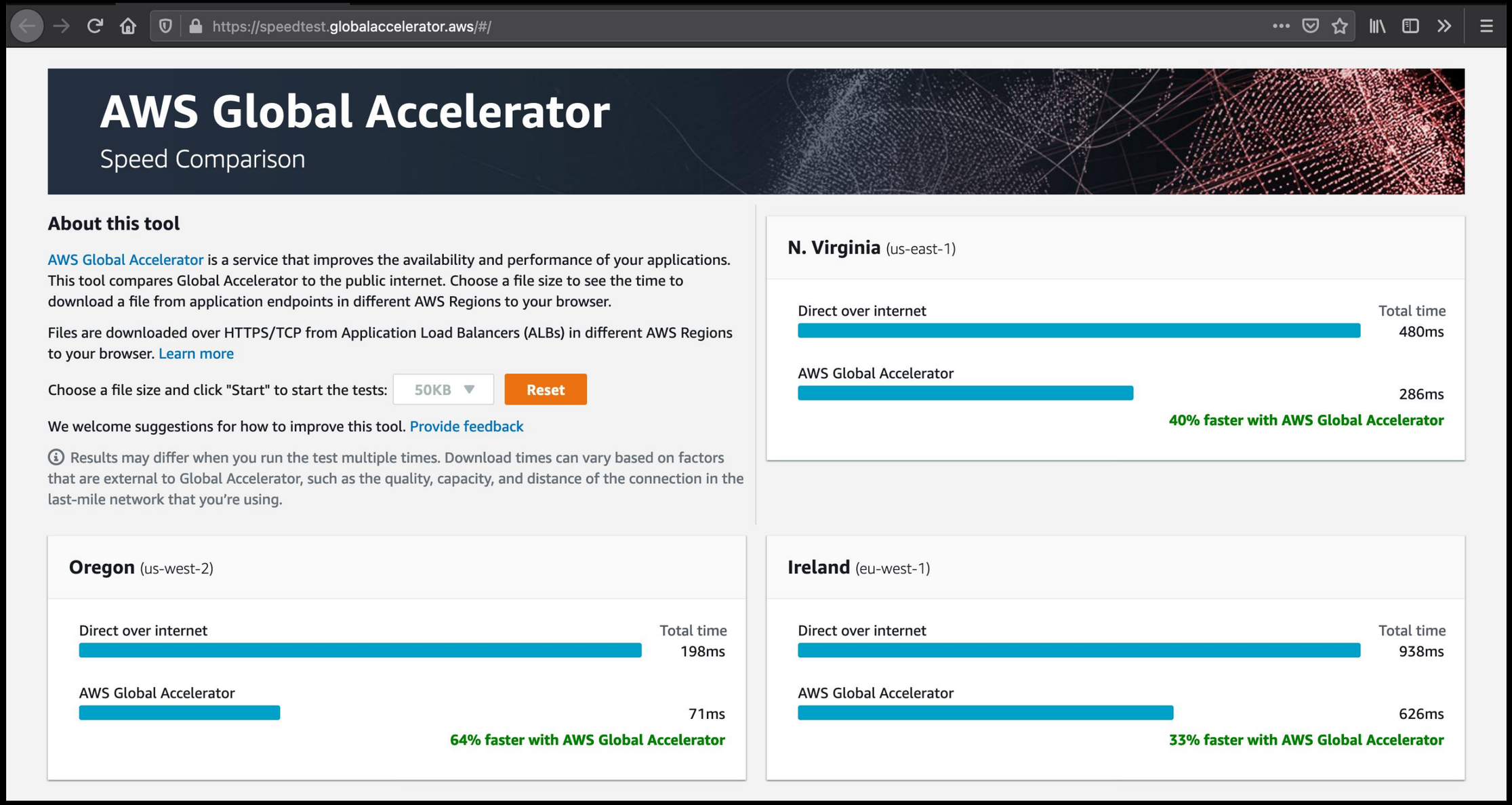
Pecky tries AWS Global Accelerator



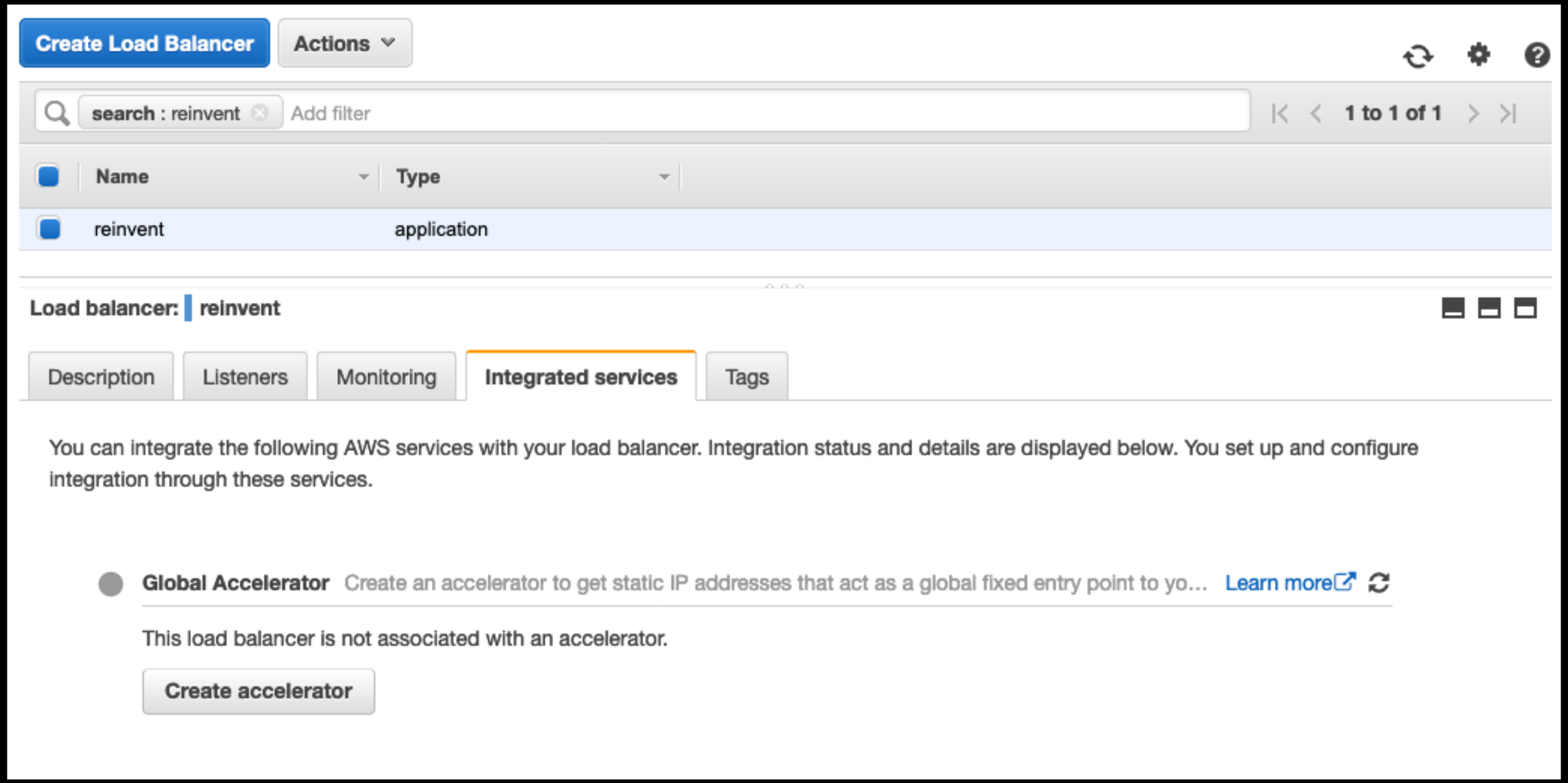


AWS Global Accelerator

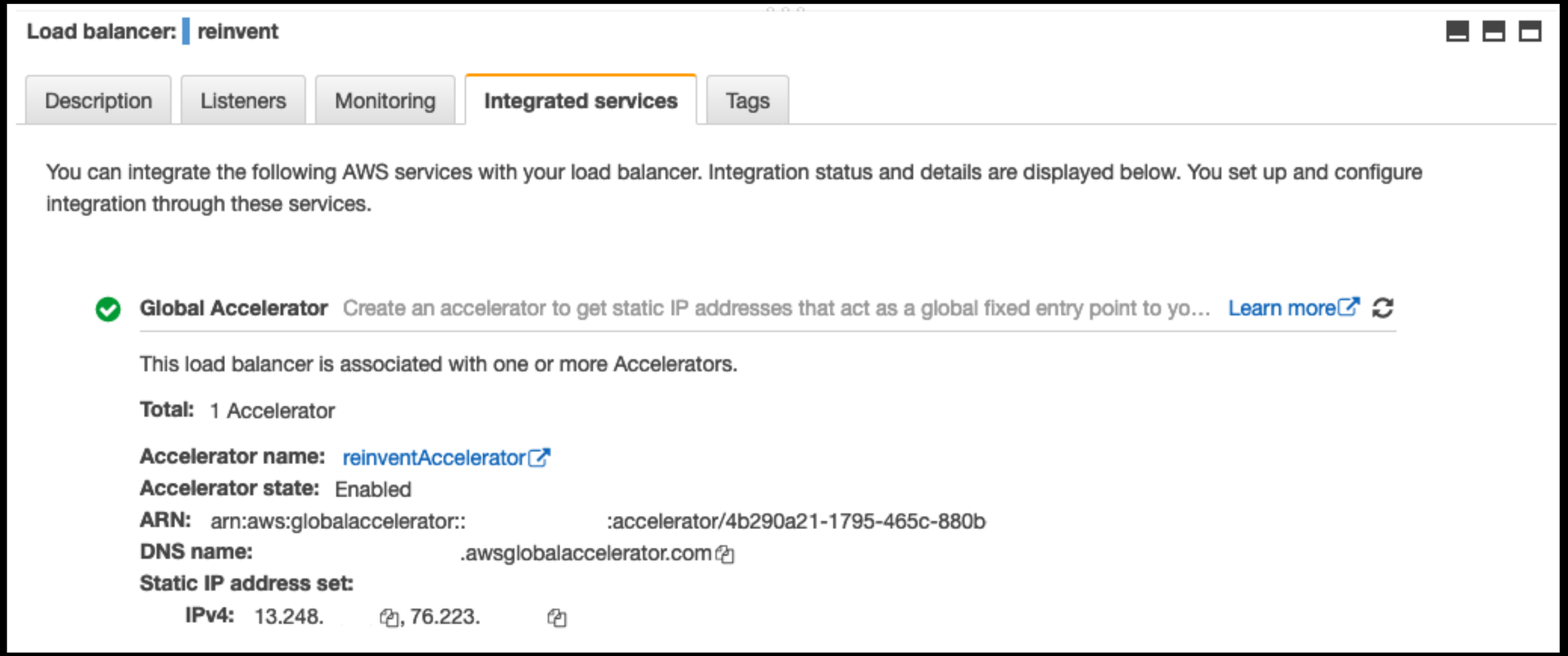
Testing AWS Global Accelerator



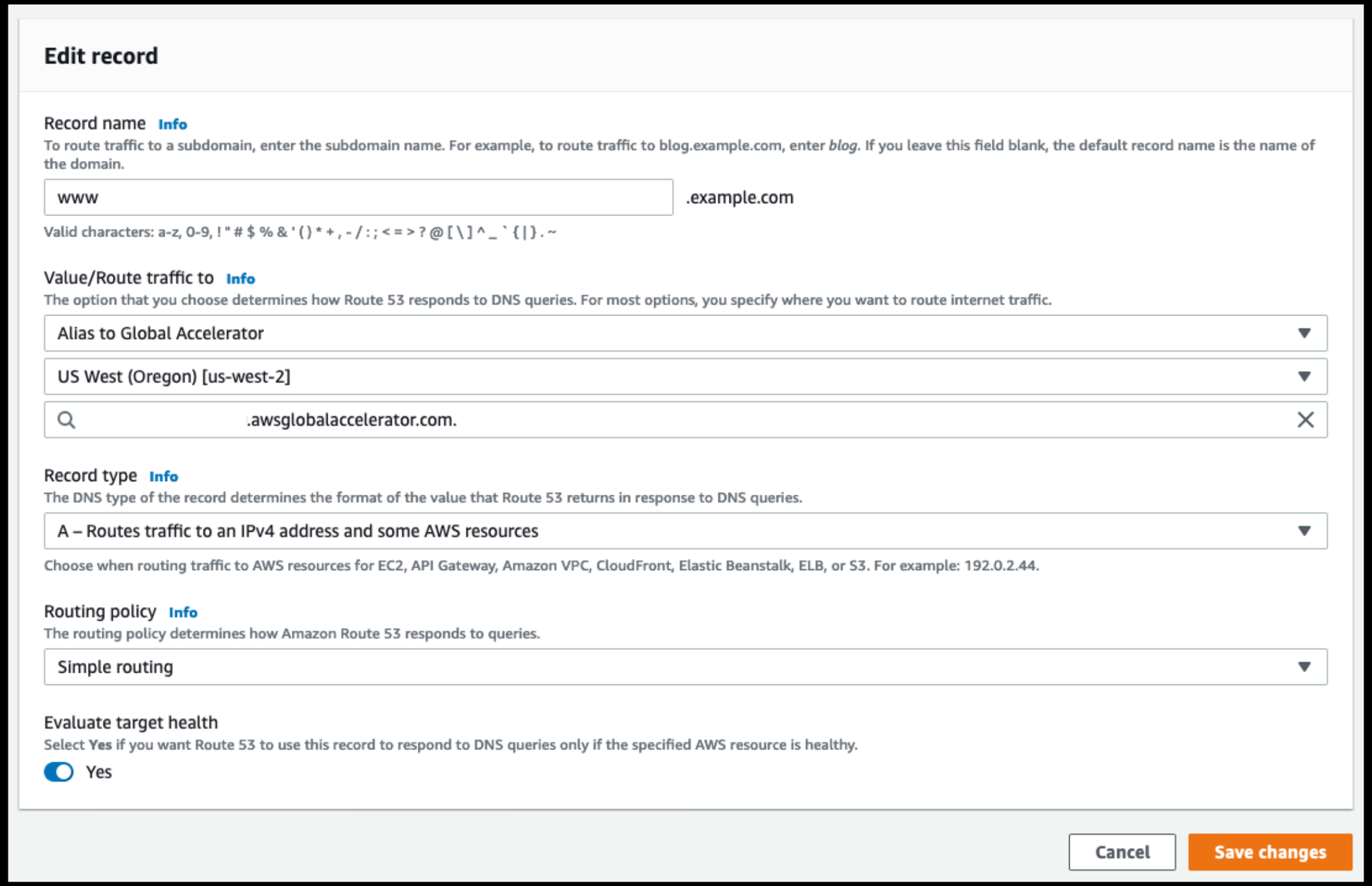
One-click acceleration



One-click acceleration

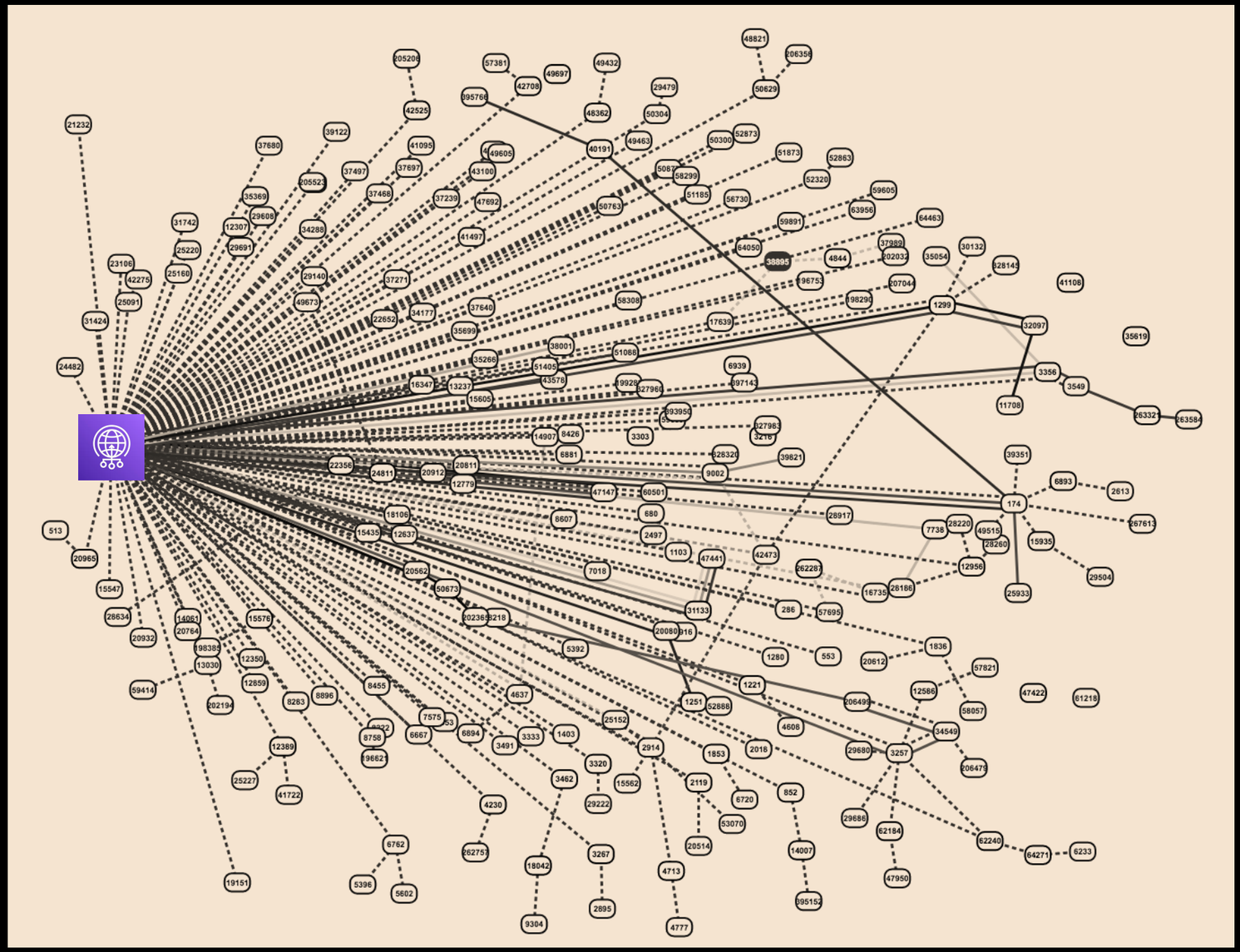


One-click acceleration

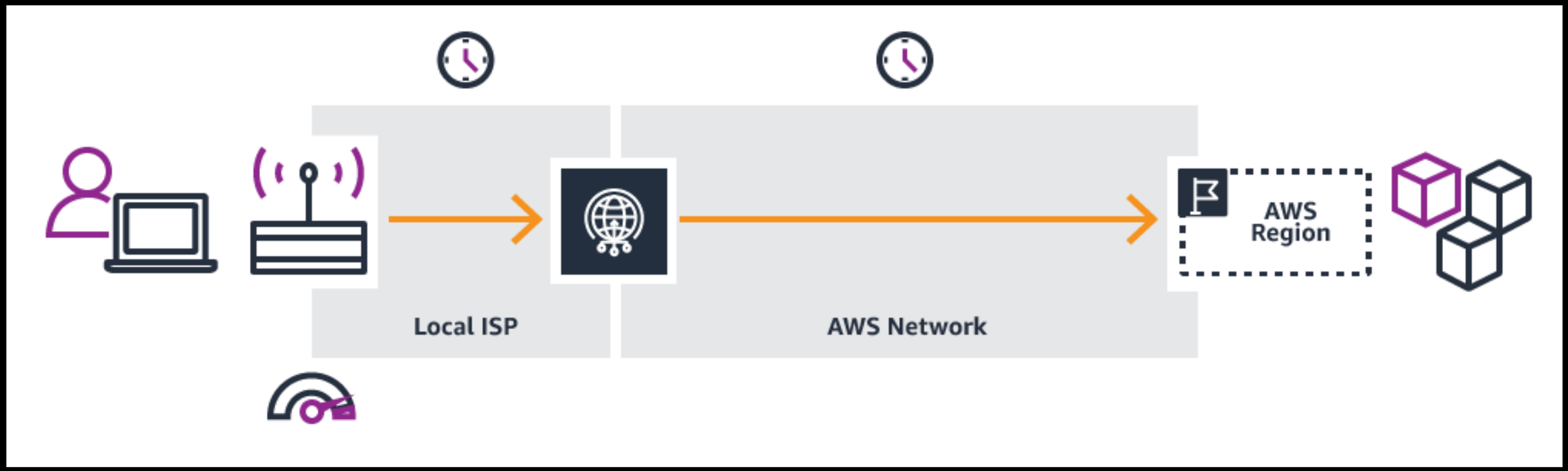


Internet view of an AWS Global Accelerator

Anycast from every AWS edge location



Shortening the distance improves performance



AWS Global Accelerator inner workings

Public internet

AWS backbone

AWS Global Accelerator inner workings

Public internet

- Variable throughput

AWS backbone

- Consistent high throughput

AWS Global Accelerator inner workings

Public internet

- Variable throughput
- Close to client, so low RTT

AWS backbone

- Consistent high throughput
- Most of the path, so most of the RTT

AWS Global Accelerator inner workings

Public internet

- Variable throughput
- Close to client, so low RTT
- Higher probability of congestion, packet loss & jitter

AWS backbone

- Consistent high throughput
- Most of the path, so most of the RTT
- Very low probability of congestion, packet loss & jitter

AWS Global Accelerator inner workings

Public internet

- Variable throughput
- Close to client, so low RTT
- Higher probability of congestion, packet loss & jitter
- Restricted MSS

AWS backbone

- Consistent high throughput
- Most of the path, so most of the RTT
- Very low probability of congestion, packet loss & jitter
- Jumbo frames, large MSS

AWS Global Accelerator inner workings

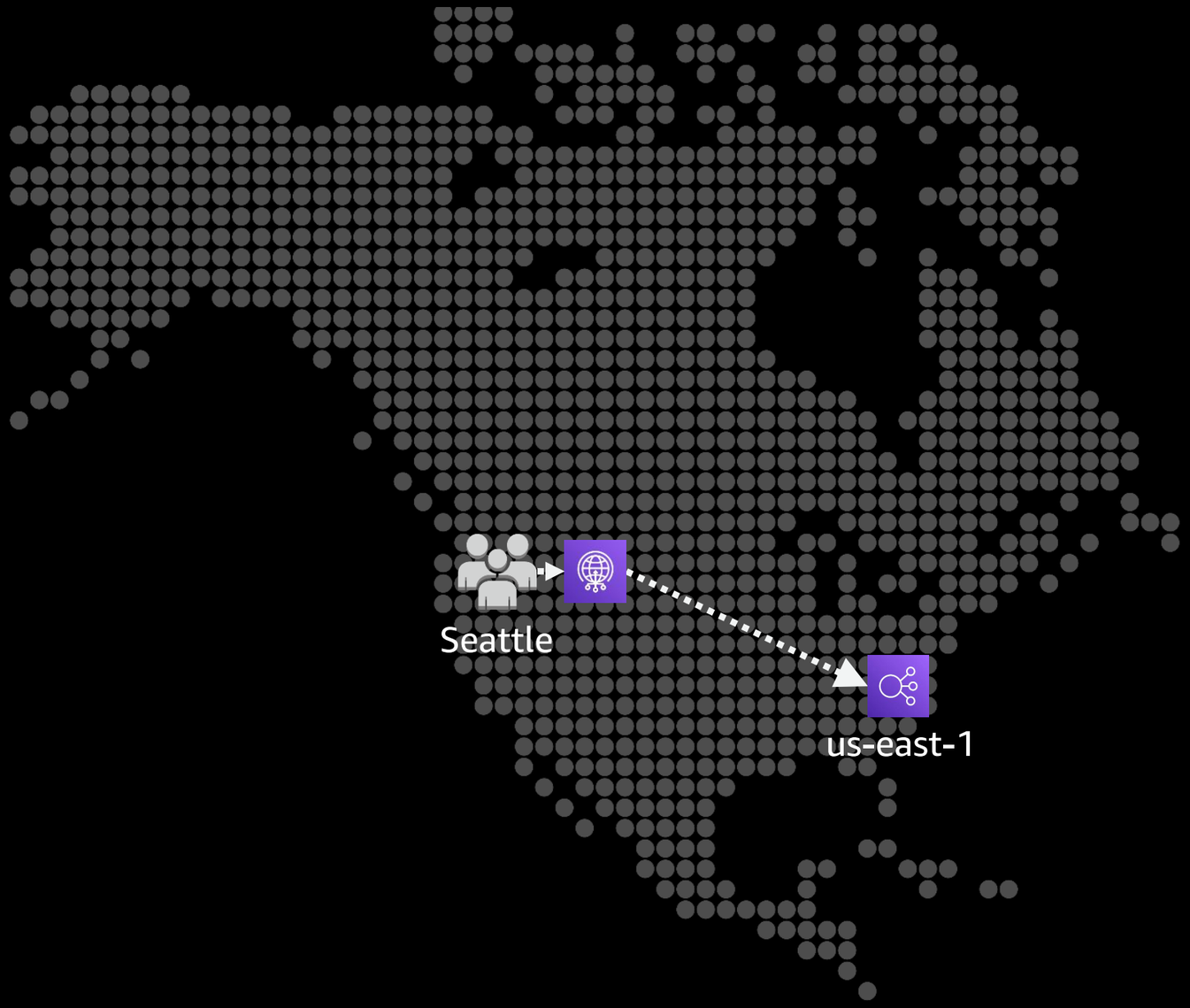
Public internet

Optimized for fast
TCP congestion control
ramp-up & recovery

AWS backbone

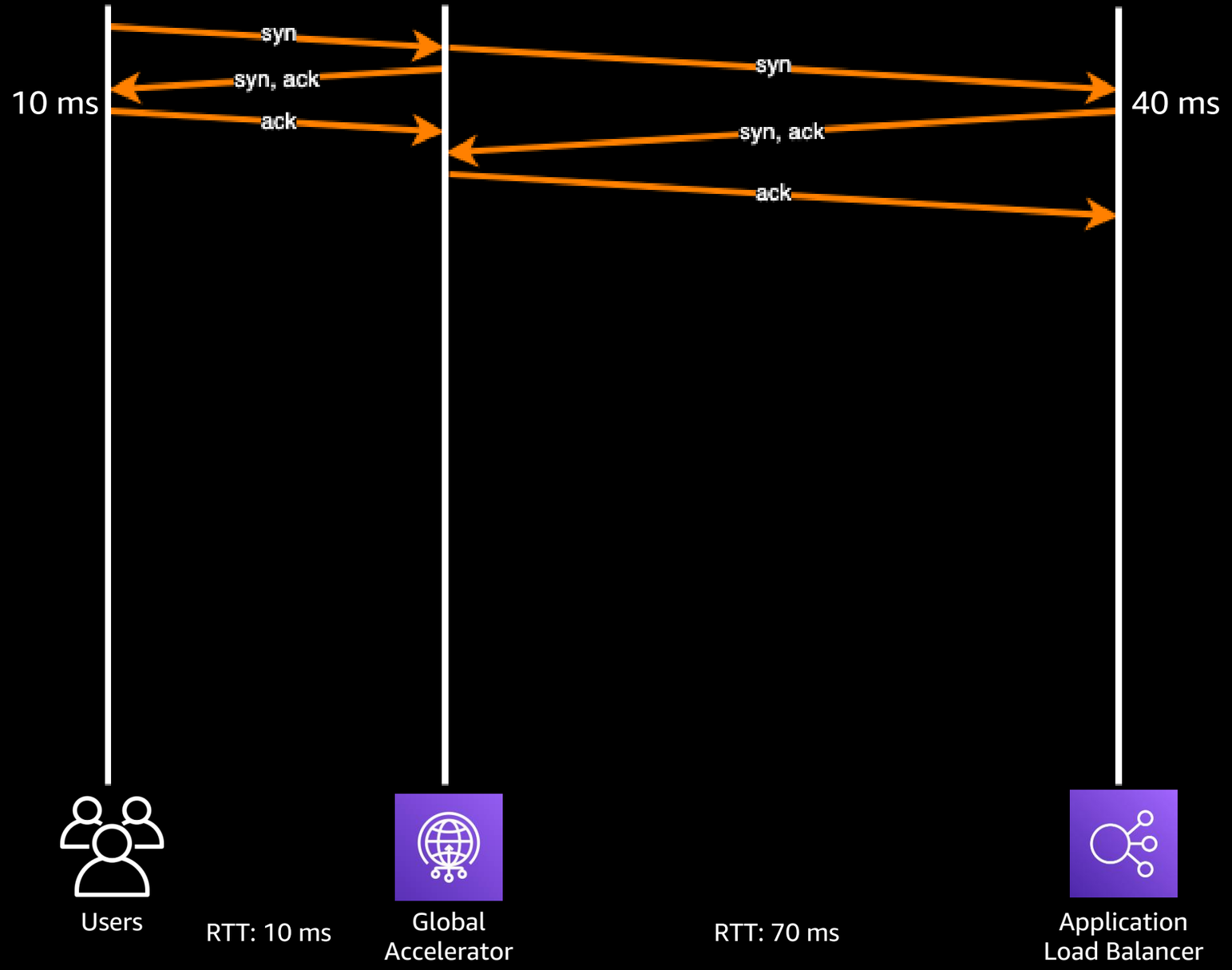
Optimized for maximum
network usage

Data transfer example

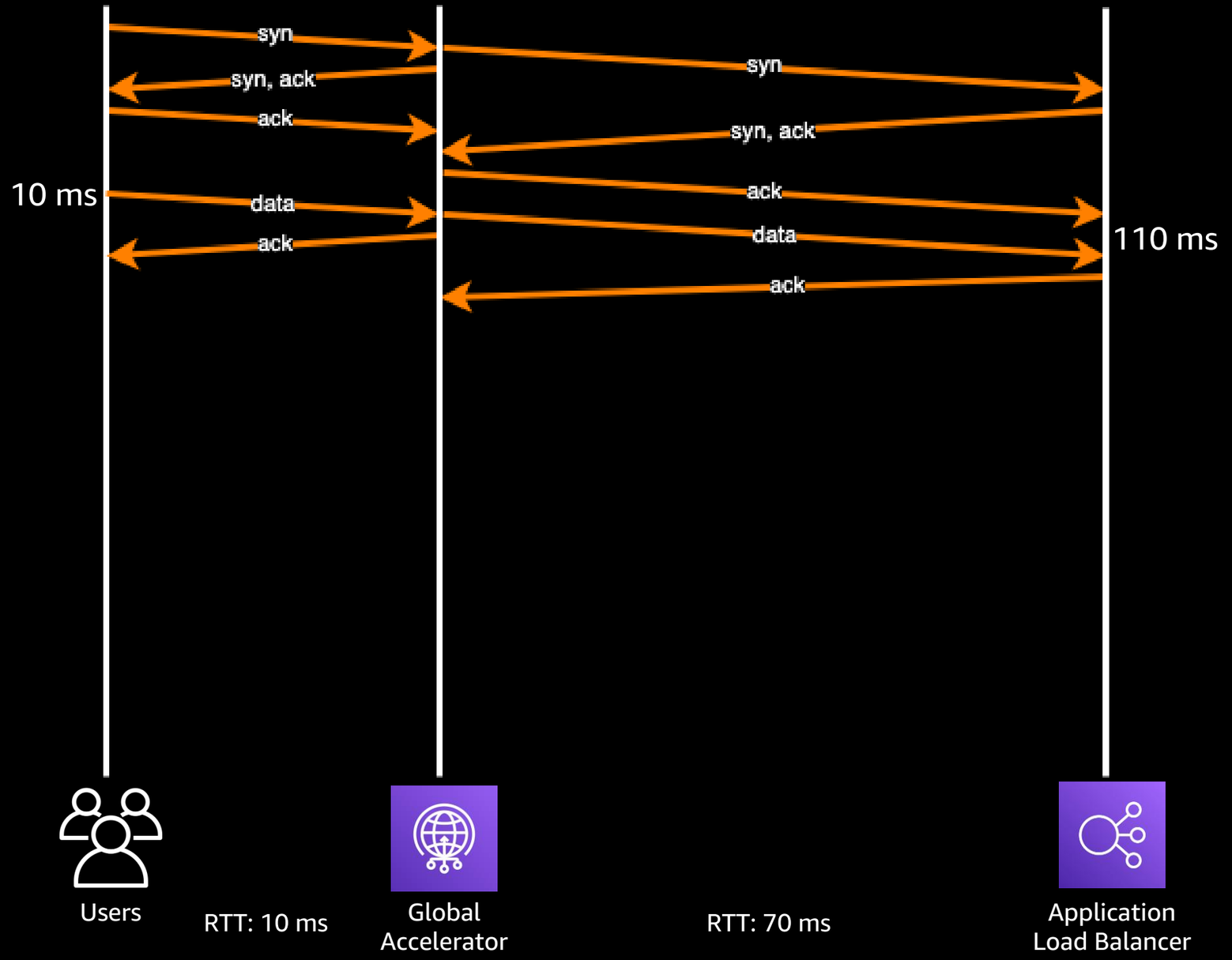


Data transfer via AWS Global Accelerator

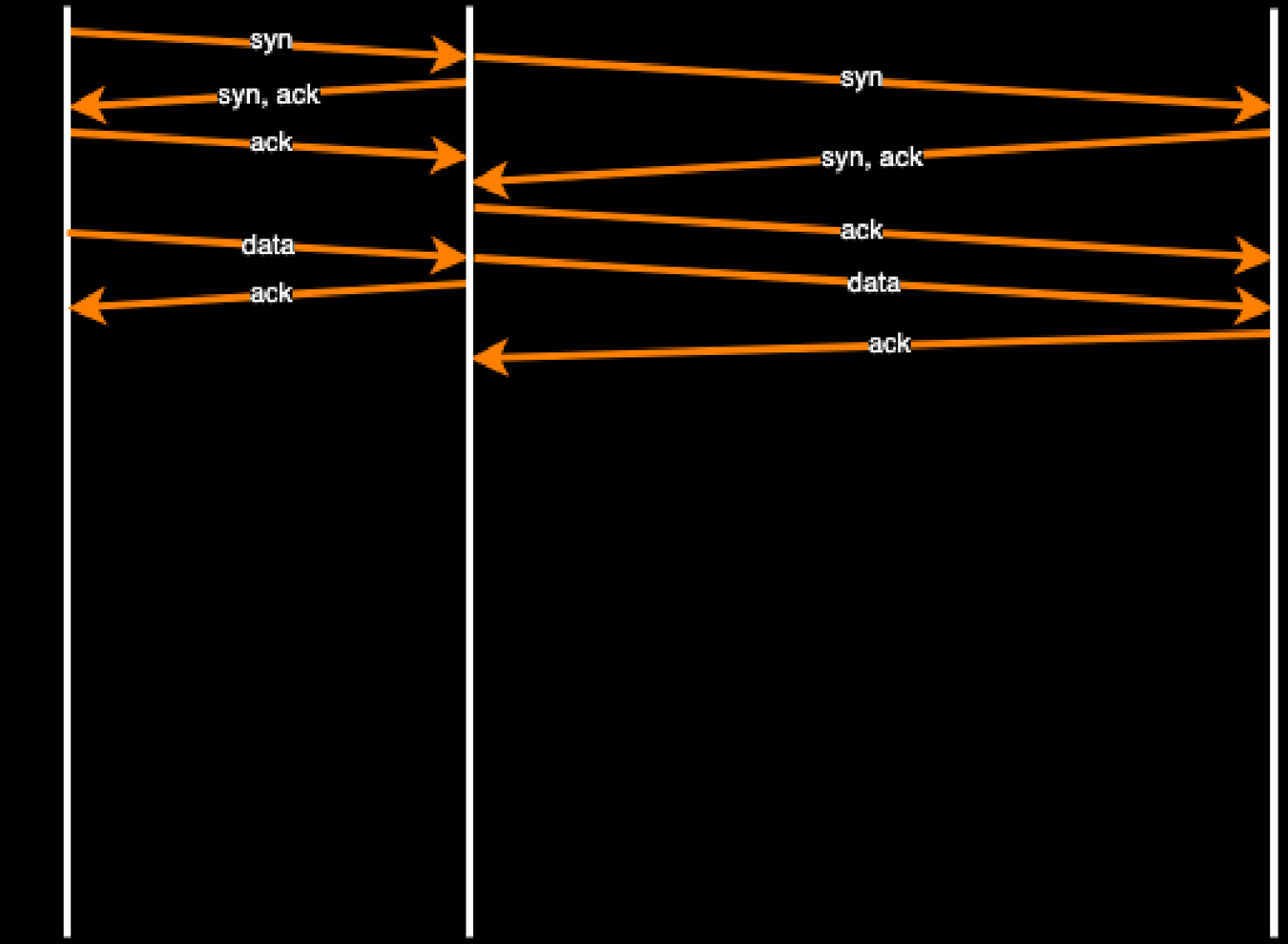
Handshake



Data transfer via AWS Global Accelerator



Data transfer via AWS Global Accelerator



Your code runs

Users

RTT: 10 ms

Global Accelerator

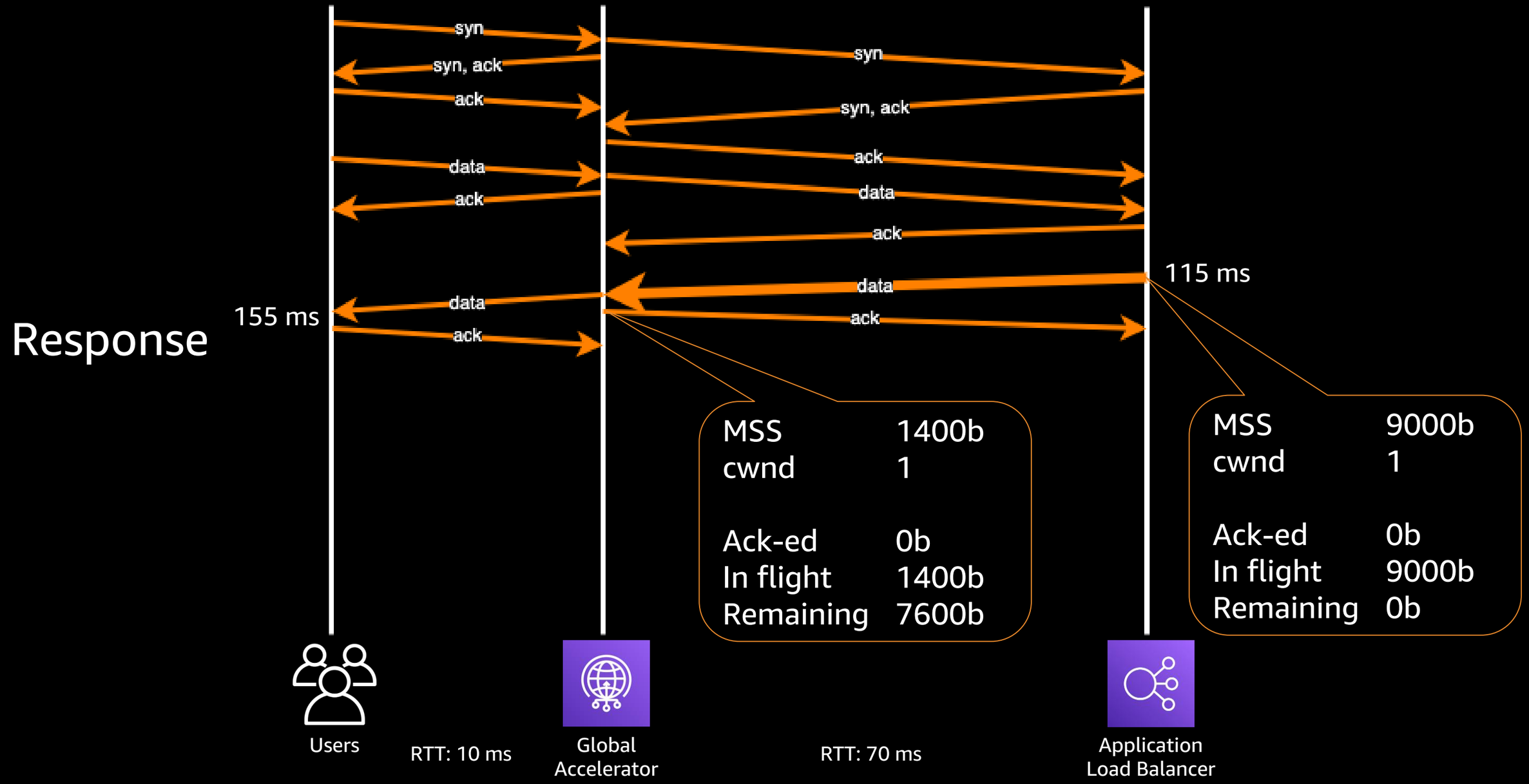
RTT: 70 ms

Application Load Balancer

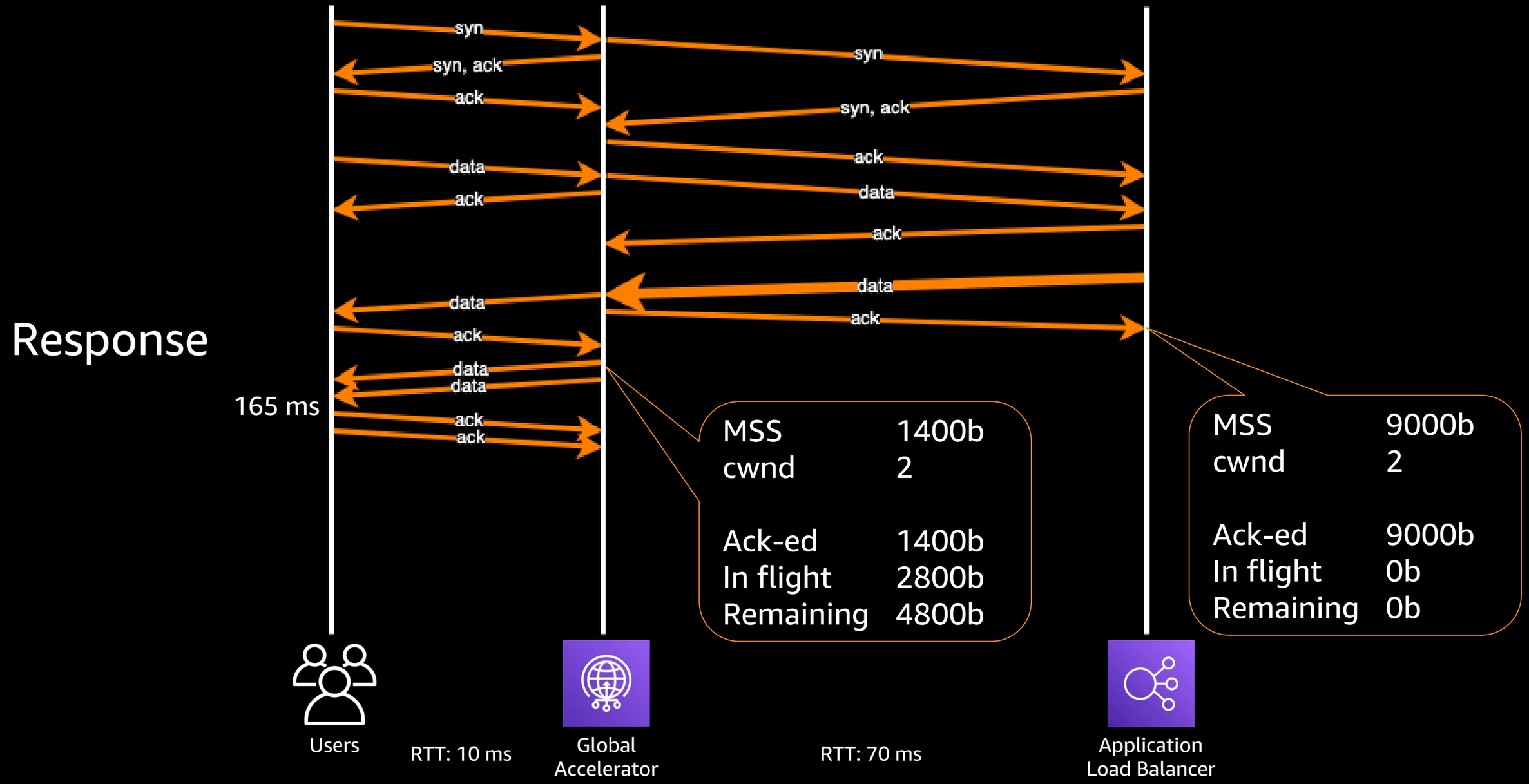
+

Lambda

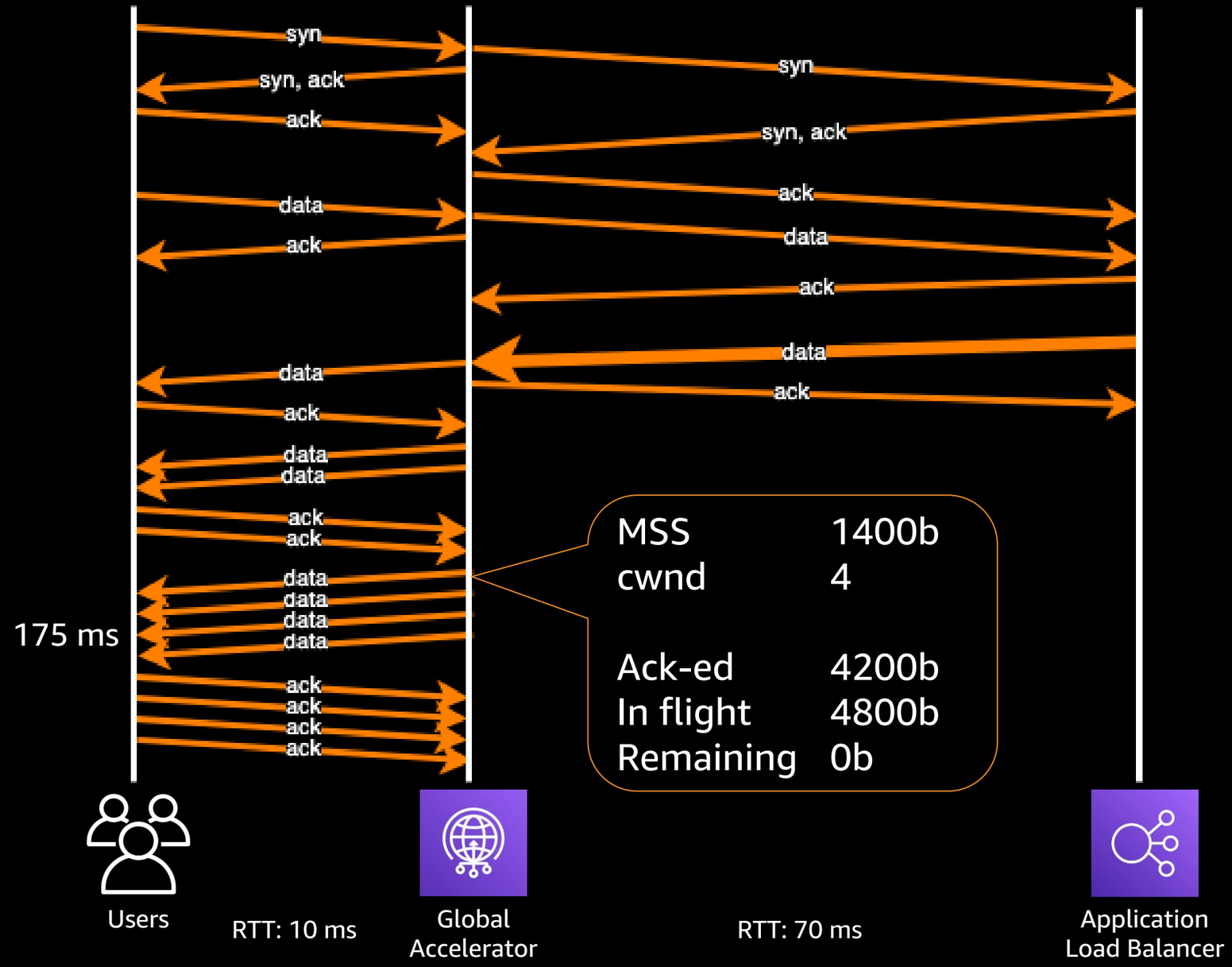
Data transfer via AWS Global Accelerator



Data transfer via AWS Global Accelerator



Data transfer via AWS Global Accelerator



Total time spent comparison

With Global Accelerator

175 ms

Without Global Accelerator

325 ms

Improvement

150 ms, or 46%

How our story ends





“Thanks for making your app faster. I wish everyone reacted to my tickets this fast!”
– Rufus

Tips and tricks



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tips and tricks

1. Enable client-side metrics

Tips and tricks

1. Enable client-side metrics
2. Reuse your SDK clients & HTTP connections

Tips and tricks

1. Enable client-side metrics
2. Reuse your SDK clients & HTTP connections
3. Reduce your request and cookie size

Tips and tricks

1. Enable client-side metrics
2. Reuse your SDK clients & HTTP connections
3. Reduce your request and cookie size
4. Increase DNS TTL when using Global Accelerator

Recap



Understand how networking impacts application performance



Learn how AWS Global Accelerator improves performance and availability



Ways that you can mitigate network challenges yourself

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.