



UCL

FROST: A WEB PAGE

COMPLETION TIME ANALYSIS TOOL

Lynne Salameh, Mark Handley, Brad Karp

A Typical Web Page Contains Several Resources

```
<html>
  <head>
    <script>
      function downloadImage()
      {
        //Create a new Image
        var anImage = new Image();
        anImage.setAttribute("src", "rainbow.jpg");

        //Append it to the div tag
        var element = document.getElementById("firstDiv");
        element.appendChild(anImage);
      }
      function fireTimer(){
        setTimeout(downloadImage, 10);
      }

      function loadFrame() {
        fireTimer();
      }
    </script>
  </head>
  <body onload=loadFrame()>
    <link rel="stylesheet" type="text/css" href="theme.css"></link>
    <div id="firstDiv">
      <p2> Some Paragraph </p2>
    </div>
    <script src="image_create_sea_append.js"></script>
    <div id="secondDiv">
      <p> Some more text </p>
    </div>
  </body>
</html>
```

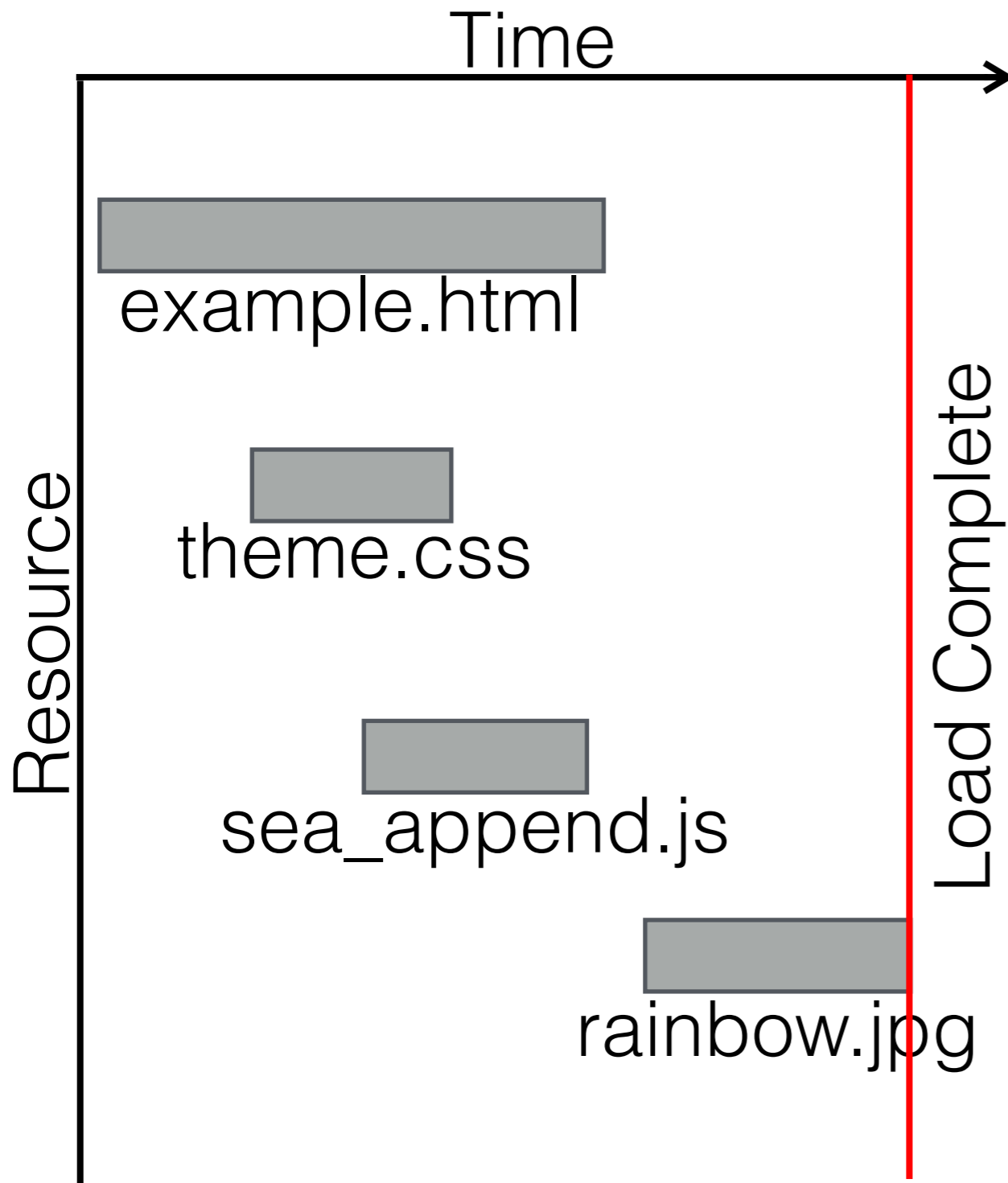
A Typical Web Page Contains Several Resources

```
<html>
  <head>
    <script>
      function downloadImage()
      {
        //Create a new Image
        var anImage = new Image();
        anImage.setAttribute("src", "rainbow.jpg");

        //Append it to the div tag
        var element = document.getElementById("firstDiv");
        element.appendChild(anImage);
      }
      function fireTimer(){
        setTimeout(downloadImage, 10);
      }

      function loadFrame() {
        fireTimer();
      }
    </script>
  </head>
  <body onload=loadFrame()>
    <link rel="stylesheet" type="text/css" href="theme.css"></link>
    <div id="firstDiv">
      <p2> Some Paragraph </p2>
    </div>
    <script src="image_create_sea_append.js"></script>
    <div id="secondDiv">
      <p> Some more text </p>
    </div>
  </body>
</html>
```

Resources Downloaded As Web Page is Parsed



```
<html>
  <head>
    <script>
      function downloadImage()
      {
        //Create a new Image
        var anImage = new Image();
        anImage.setAttribute("src", "rainbow.jpg");

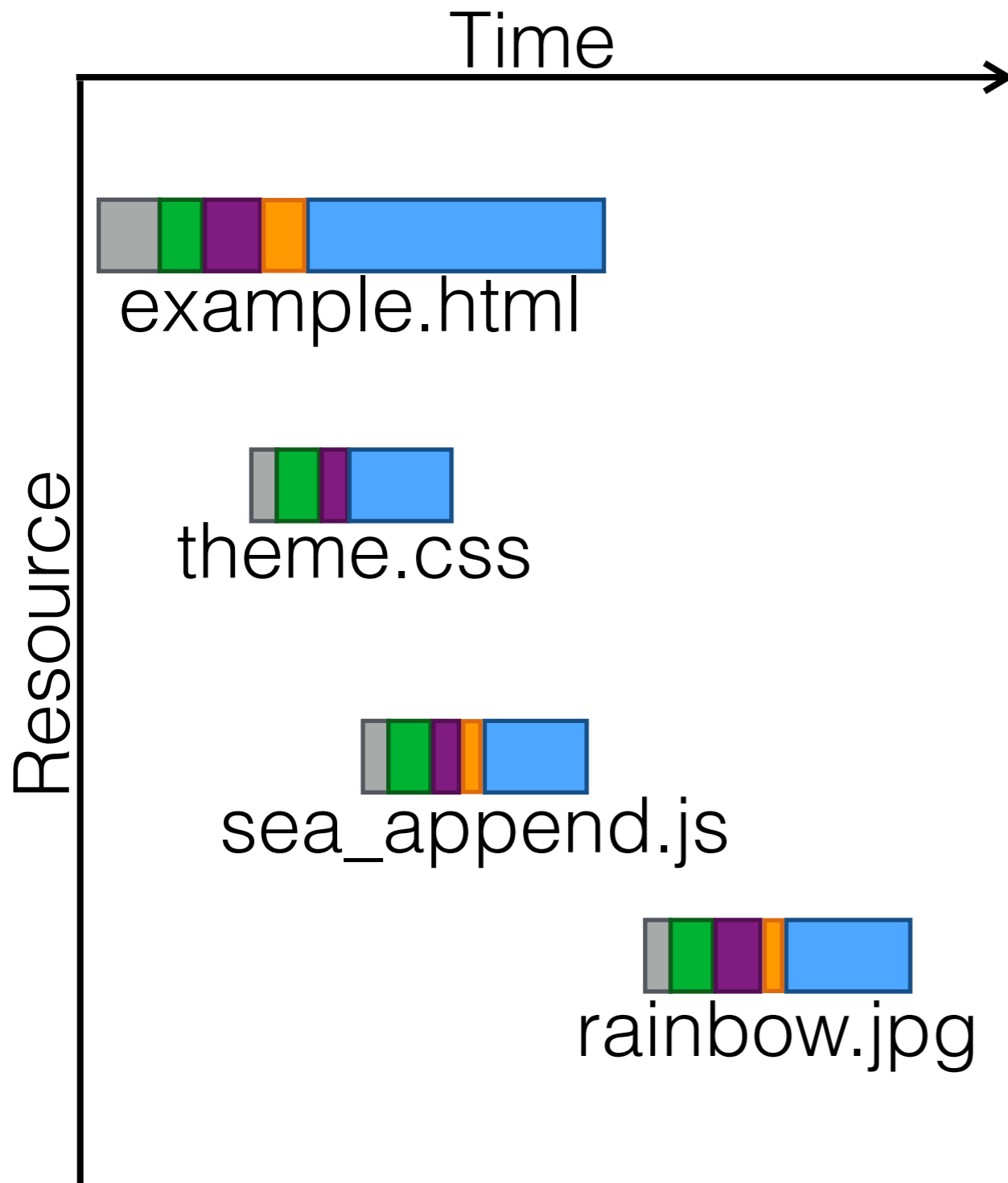
        //Append it to the div tag
        var element = document.getElementById("firstDiv");
        element.appendChild(anImage);
      }
      function fireTimer(){
        setTimeout(downloadImage, 10);
      }

      function loadFrame() {
        fireTimer();
      }
    </script>
  </head>
  <body onload=loadFrame()>
    <link rel="stylesheet" type="text/css" href="theme.css"></link>
    <div id="firstDiv">
      <p2> Some Paragraph </p2>
    </div>
    <script src="image_create_sea_append.js"></script>
    <div id="secondDiv">
      <p> Some more text </p>
    </div>
  </body>
</html>
```

Parsing



Resource Load Duration Has Several Components

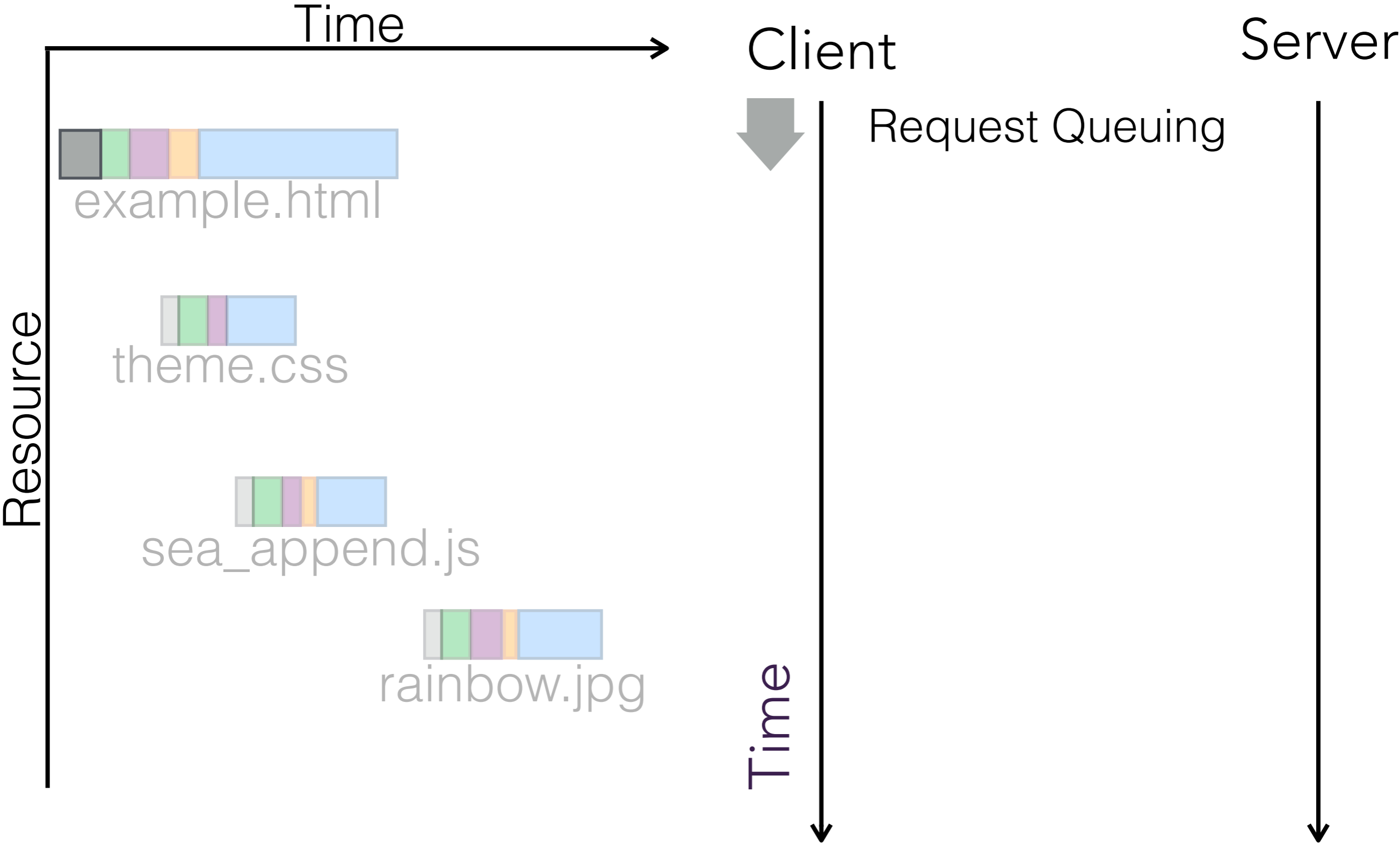


```
<html>
<head>
  <script>
    function downloadImage()
    {
      //Create a new Image
      var anImage = new Image();
      anImage.setAttribute("src", "rainbow.jpg");

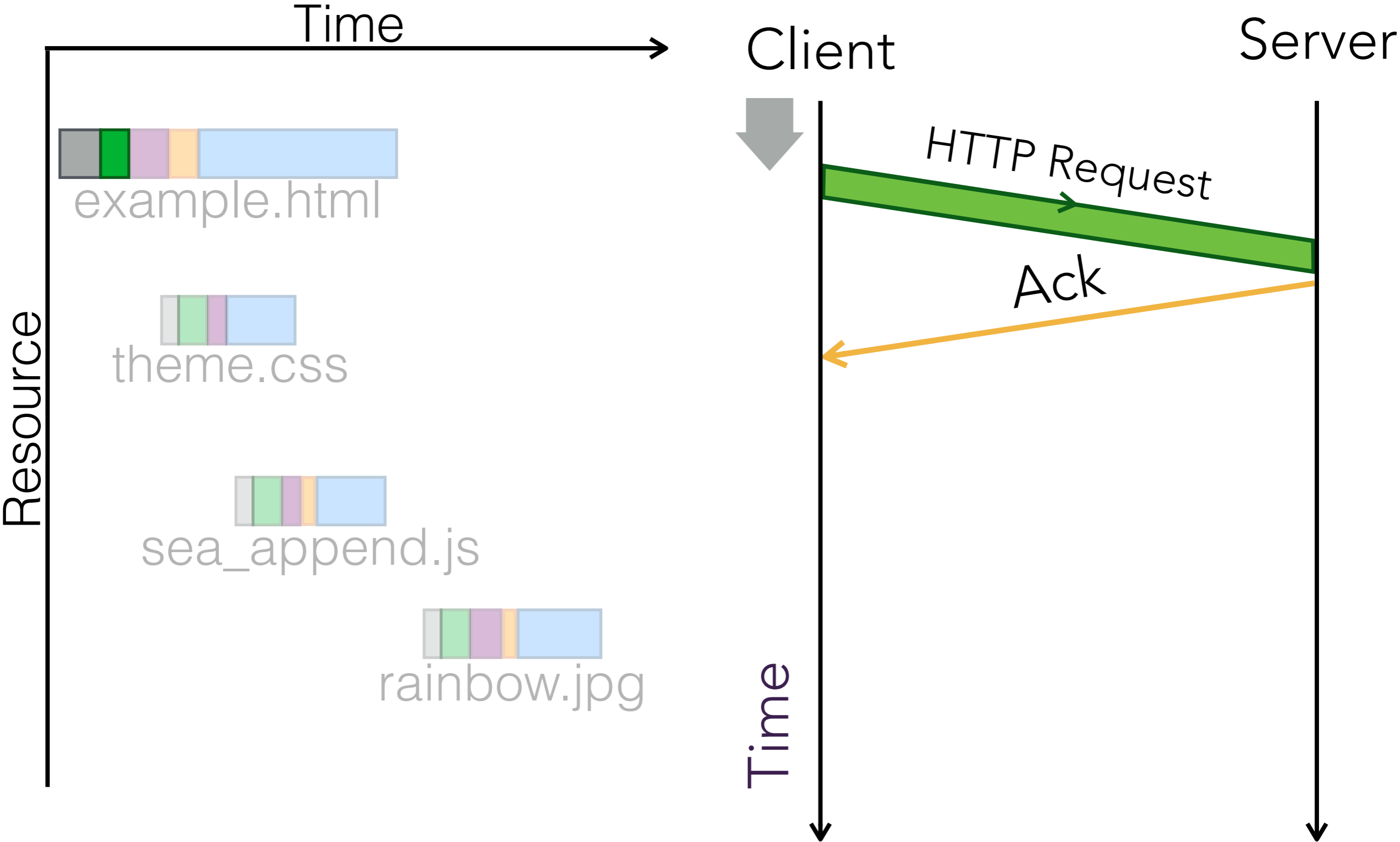
      //Append it to the div tag
      var element = document.getElementById("firstDiv");
      element.appendChild(anImage);
    }
    function fireTimer(){
      setTimeout(downloadImage, 10);
    }

    function loadFrame() {
      fireTimer();
    }
  </script>
</head>
<body onload=loadFrame()>
  <link rel="stylesheet" type="text/css" href="theme.css"></link>
  <div id="firstDiv">
    <p2> Some Paragraph </p2>
  </div>
  <script src="image_create_sea_append.js"></script>
  <div id="secondDiv">
    <p> Some more text </p>
  </div>
</body>
</html>
```

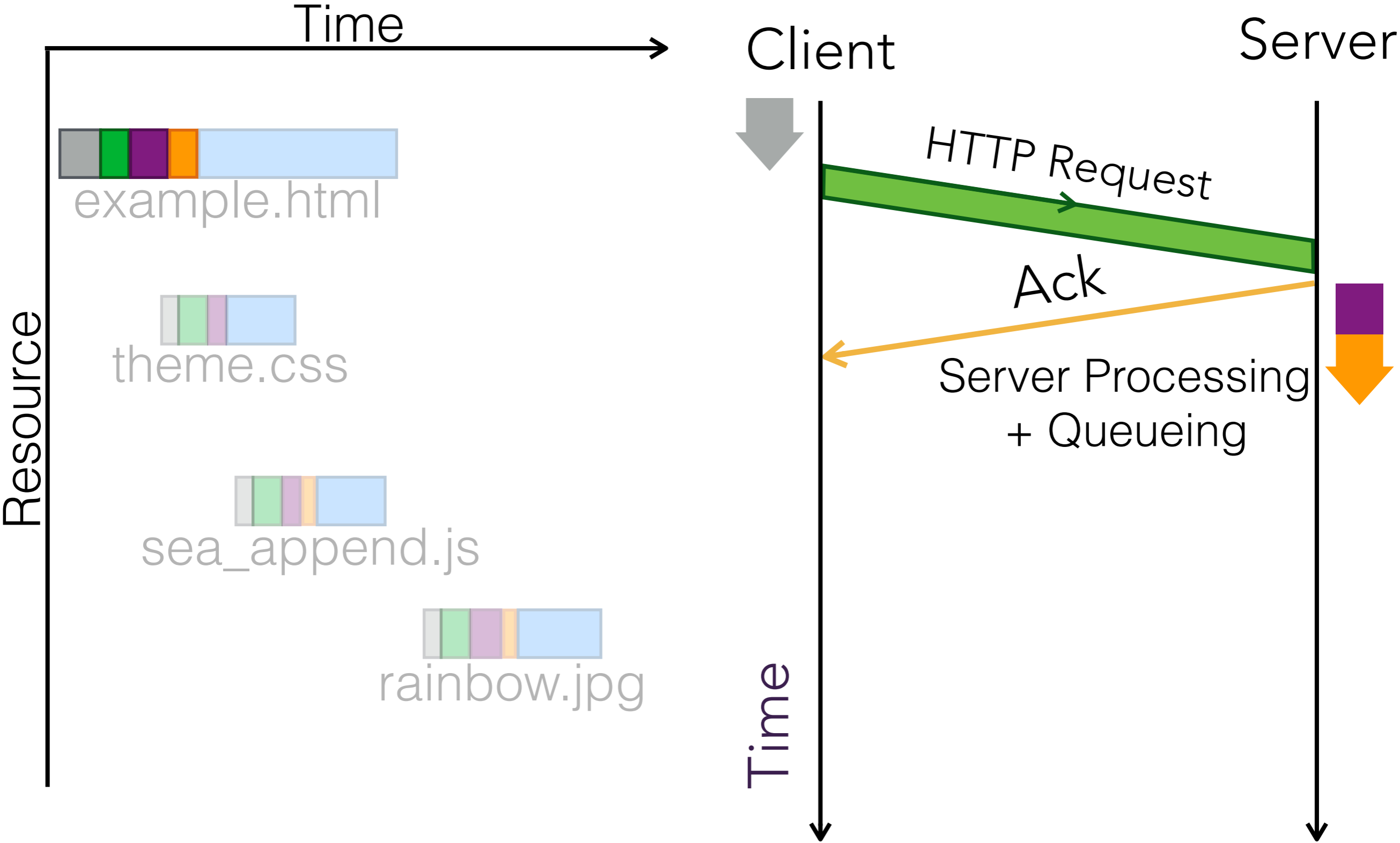
Resource Load Duration Has Several Components



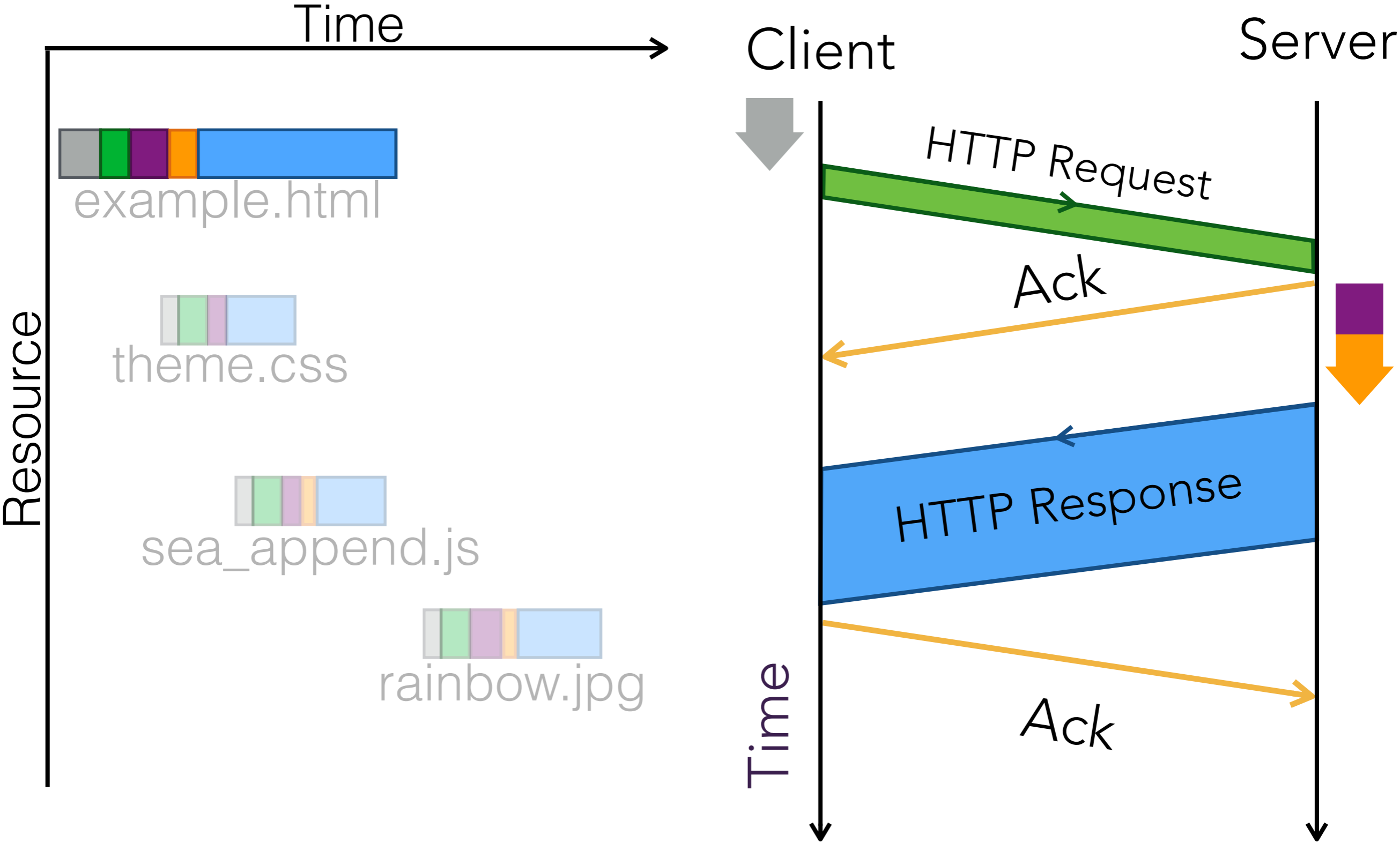
Resource Load Duration Has Several Components



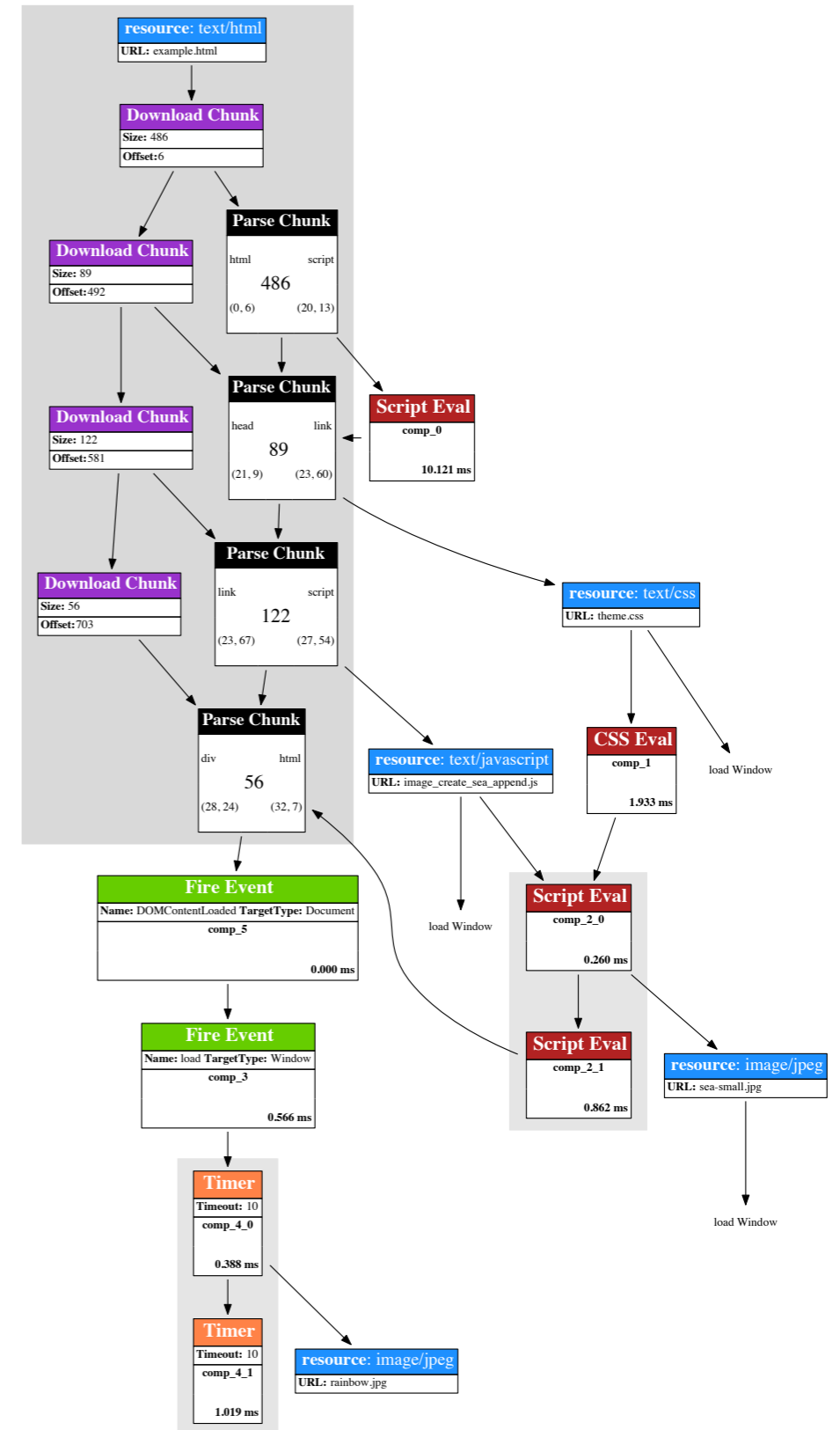
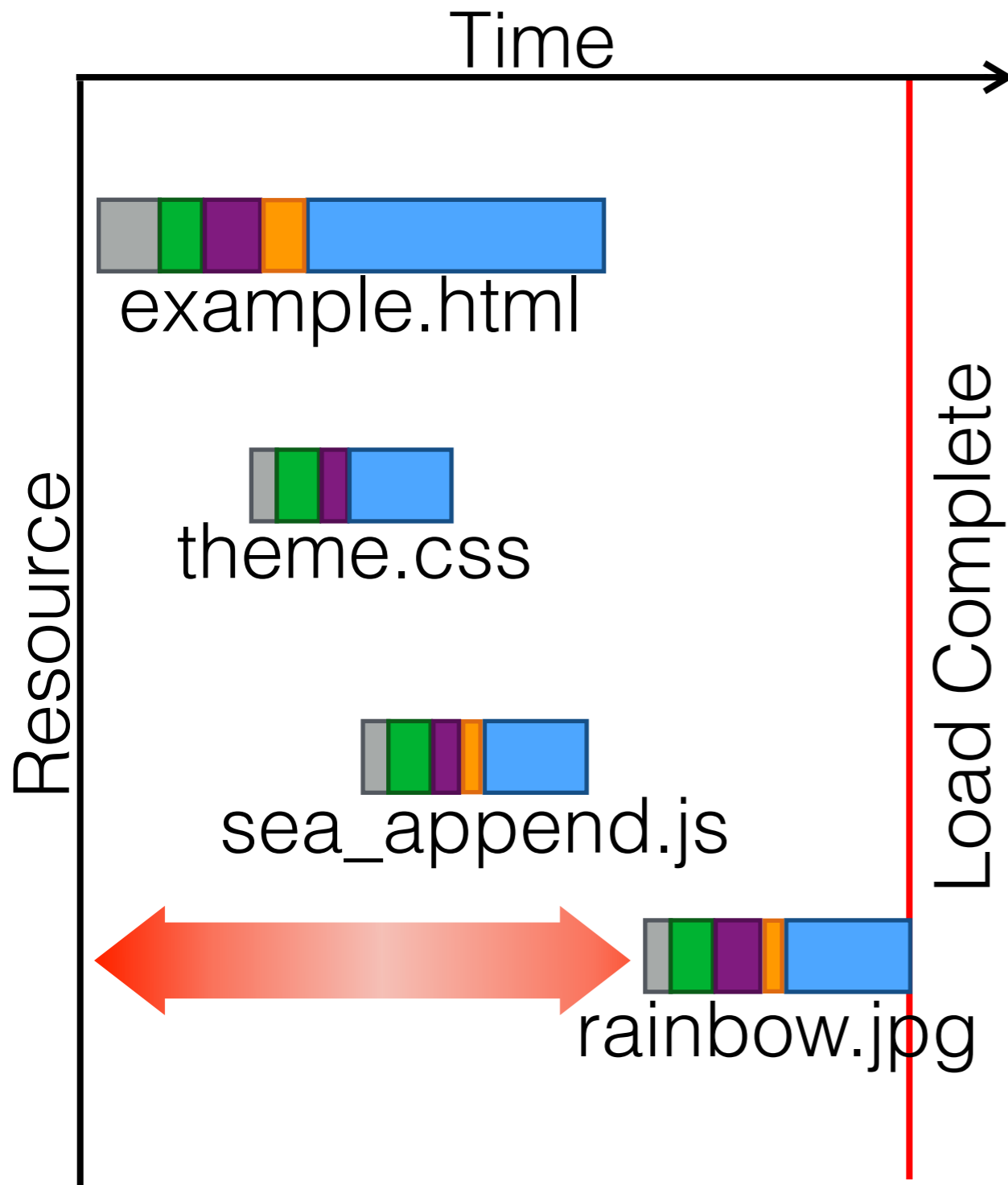
Resource Load Duration Has Several Components



Resource Load Duration Has Several Components



Web Page Resource Dependencies Influence Load Times



Goal: Use Multipath to Speed Up Page Loads

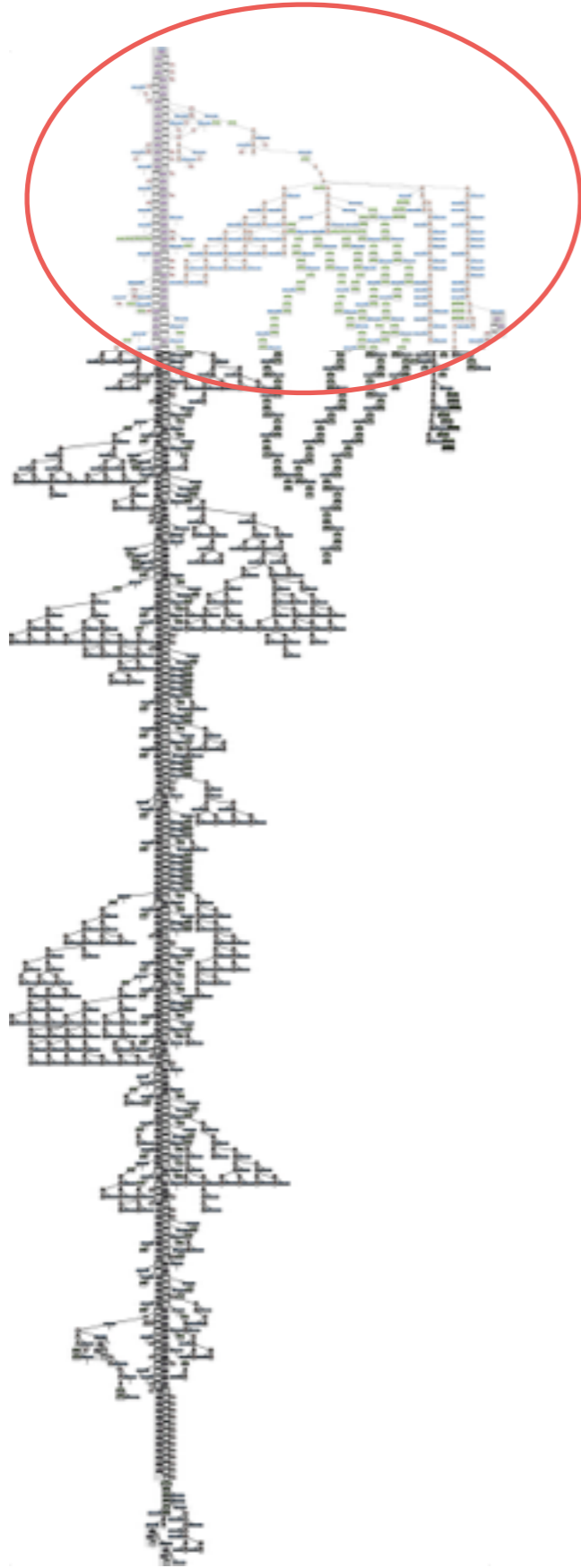
- But first: better understand behaviour of current web protocols and server designs under different network conditions.
- Build the tools that will allow us to simulate and evaluate our designs.

FROST: a simulation framework which models web page loads and allows us to compare different protocols and strategies.

FROST Uses Real Traces for Simulation

- Collected traces from ~100 Alexa 500 websites:
 - Derived dependency graphs.
 - Measured server processing time.
- Emulate a website load using ns-3 packet level simulation with dependency graphs as input.
- Model web page download over bottleneck link.

Modern Web Pages Have Complex Dependency Graphs



www.amazon.co.uk

Simulate With Large Parameter Space

Bandwidth 500 Kbps, 1Mbps, 10 Mbps, 50 Mbps,
100 Mbps, 500 Mbps, 1 Gbps

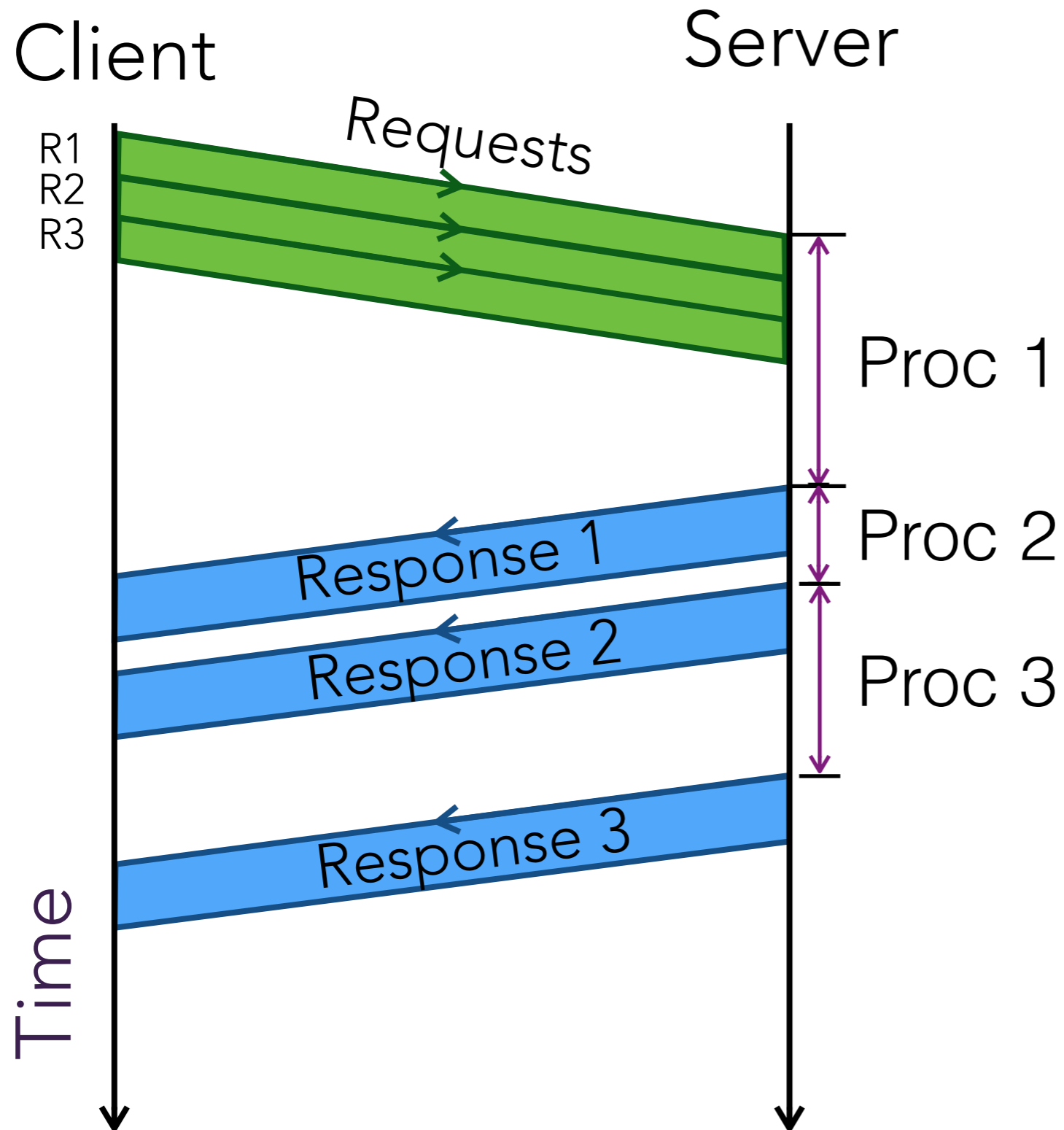
RTT 4 ms, 40 ms, 100 ms, 200 ms

Server Architecture Parallel, Sequential, Stream-Based

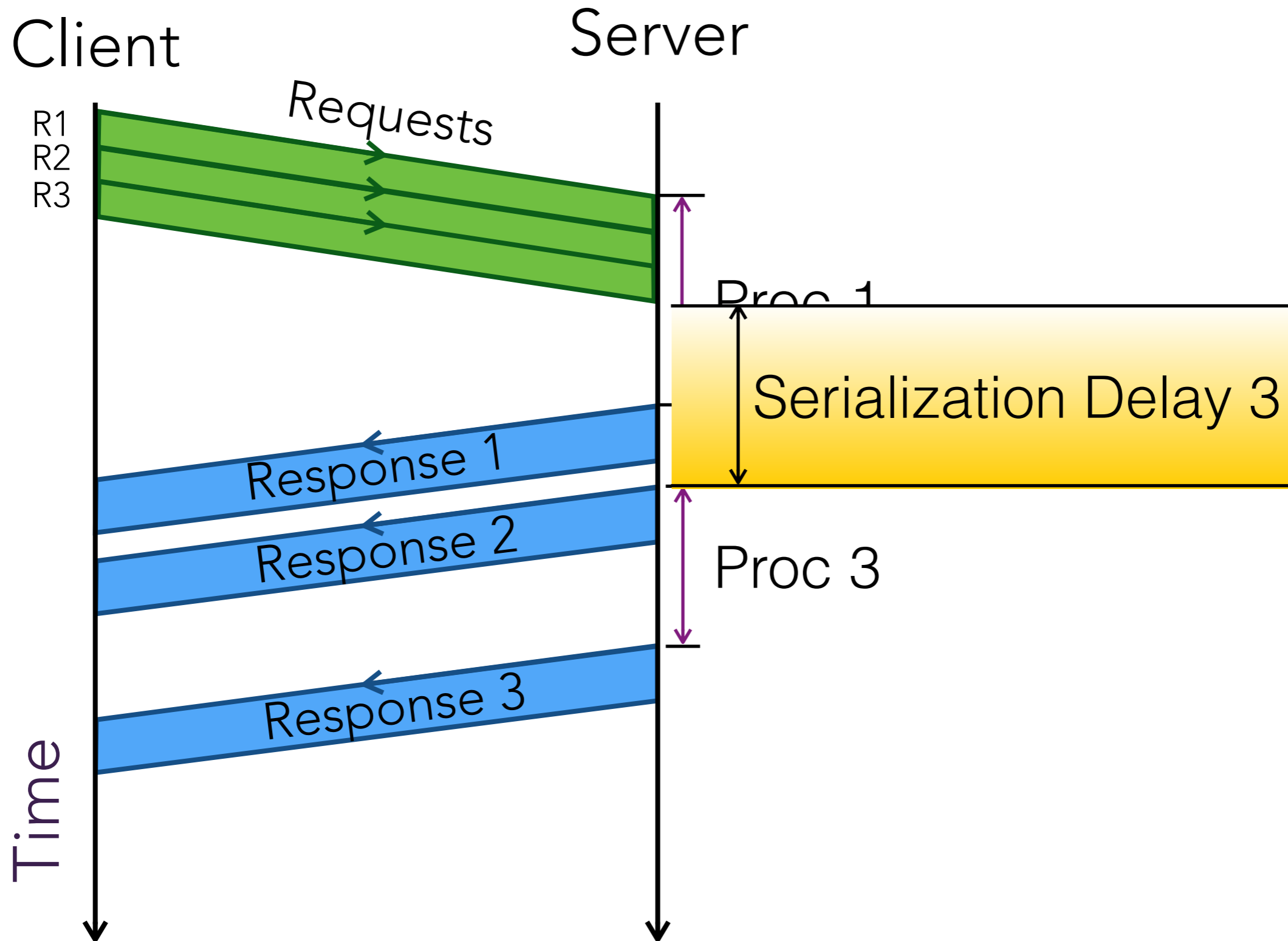
HTTP Protocol HTTP/1.0,
HTTP/1.1,
HTTP/1.1 Pipelined,
HTTP/2

Connections 1, 2, 4, 6, 10

Server Architecture Matters: A Serial Server

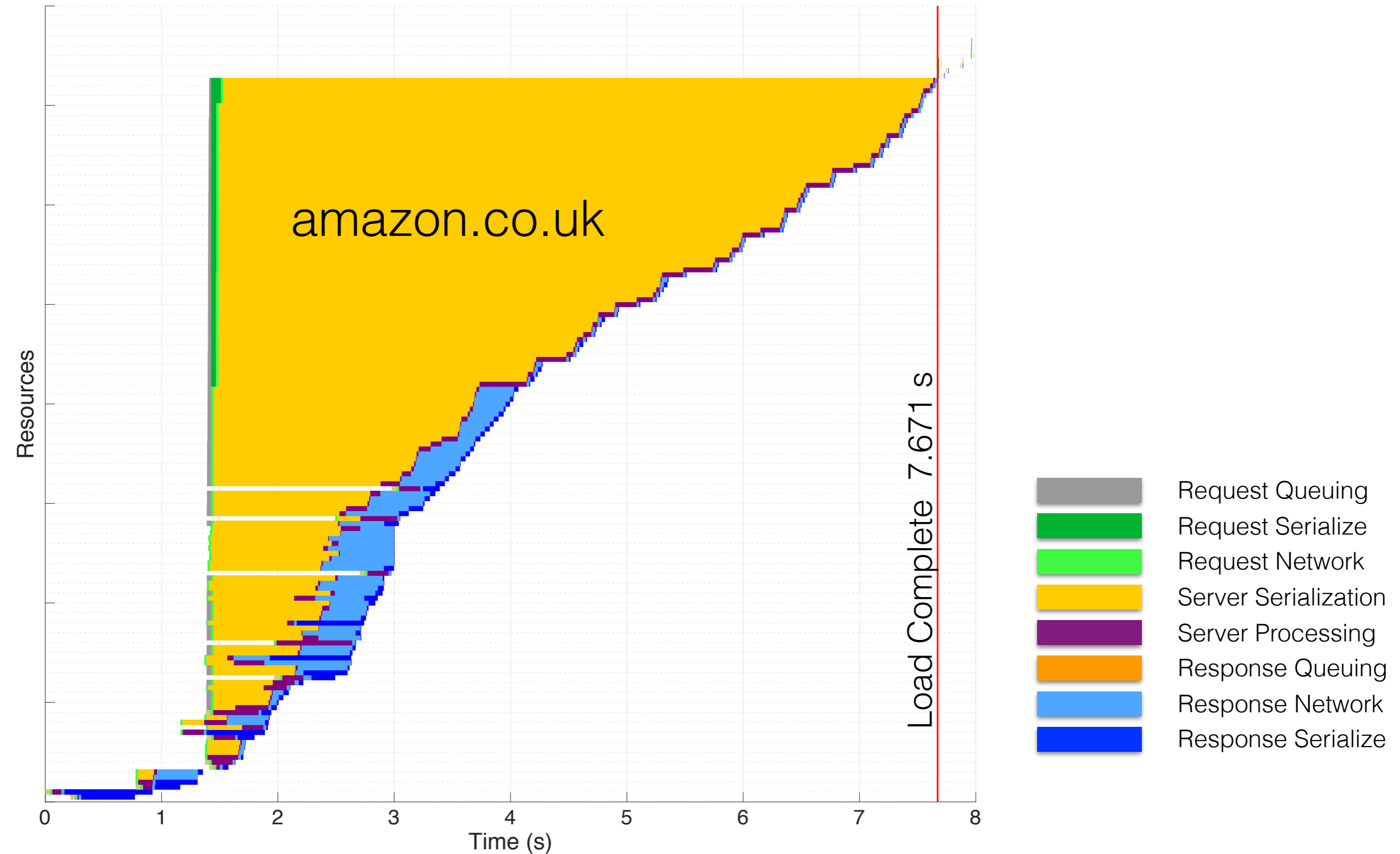


Server Architecture Matters: A Serial Server

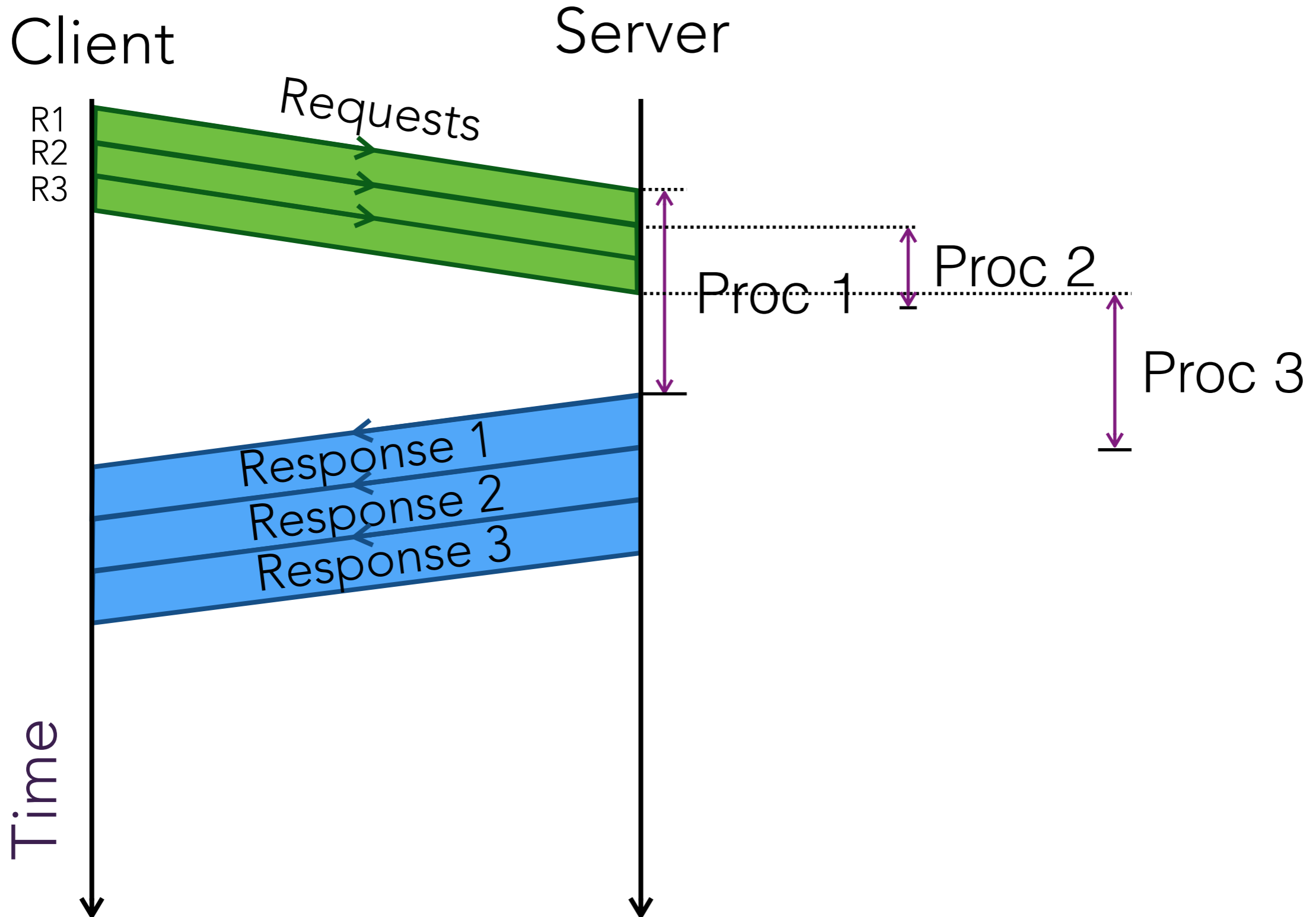


Server Architecture Matters: A Serial Server

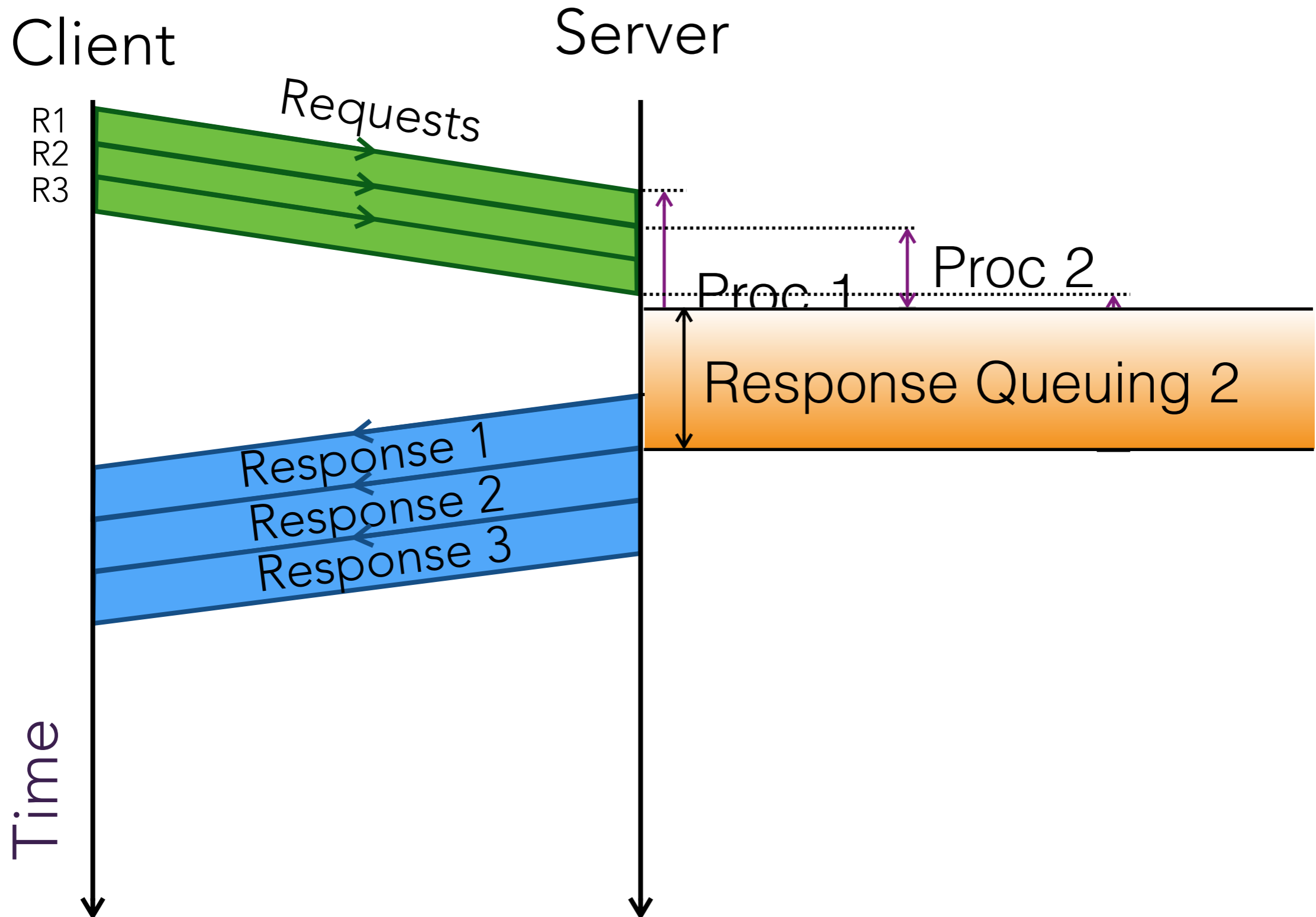
Bandwidth = 10 Mbps, RTT = 40 ms



Server Architecture Matters: A Parallel Server

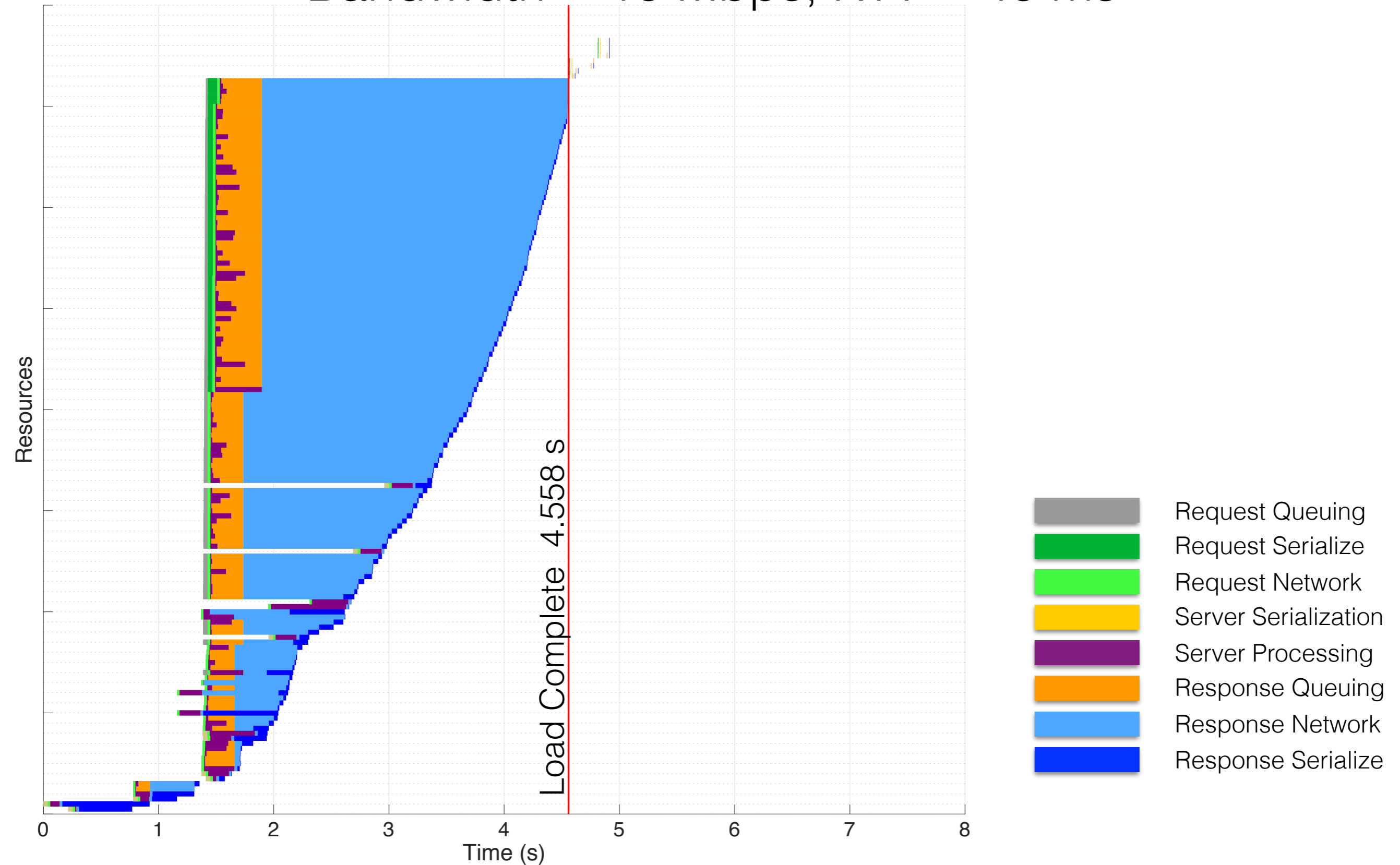


Server Architecture Matters: A Parallel Server

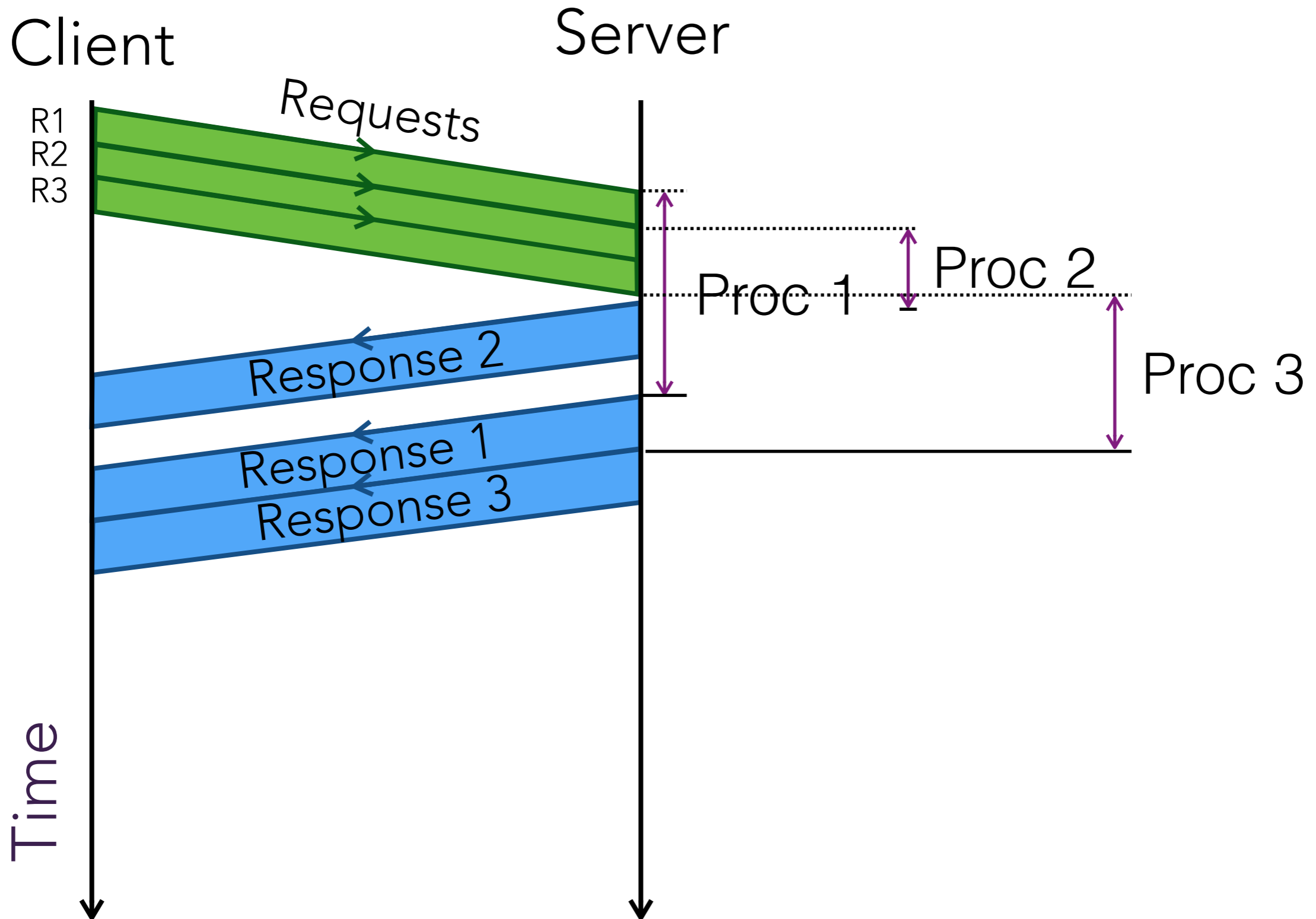


Server Architecture Matters: A Parallel Server

Bandwidth = 10 Mbps, RTT = 40 ms

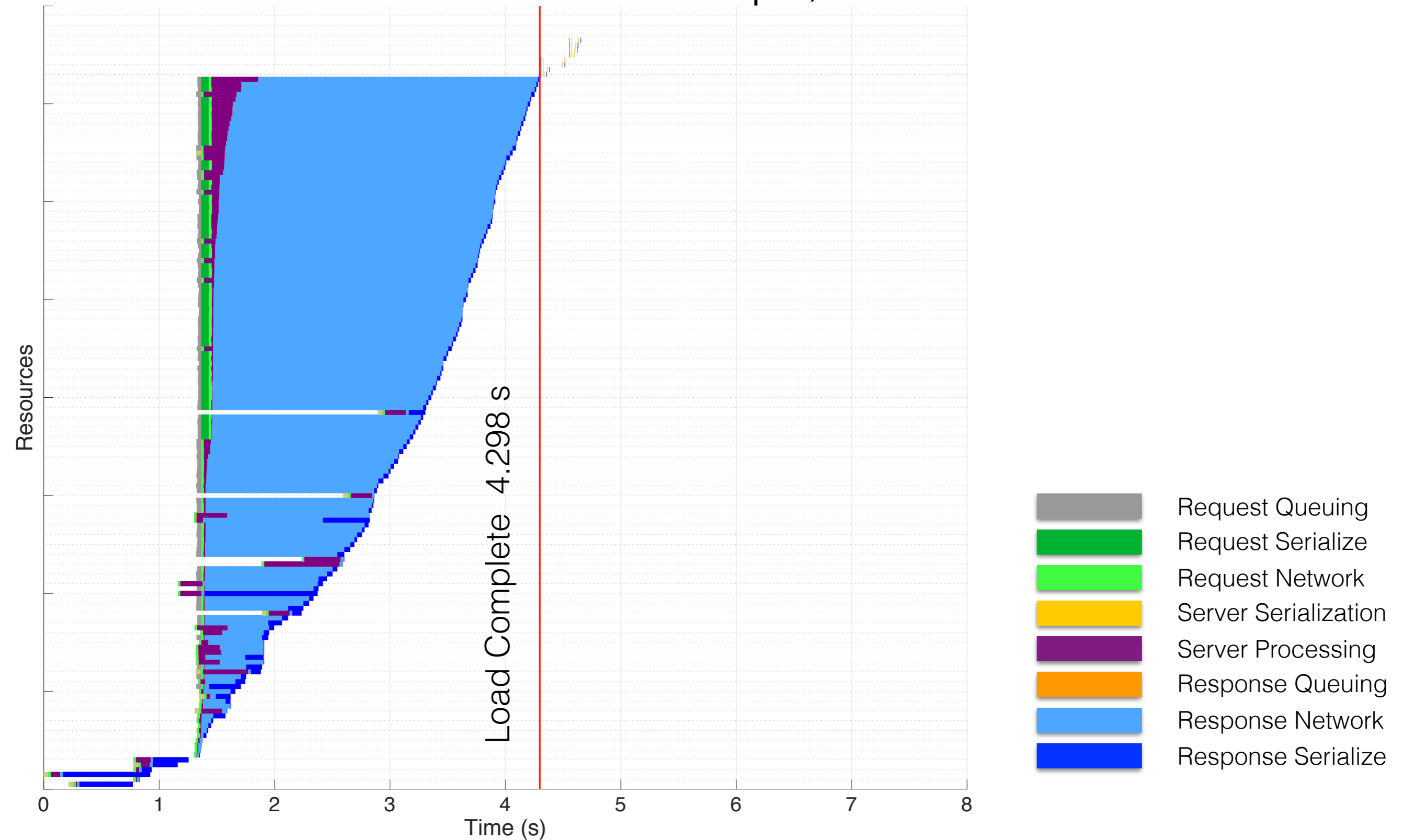


Server Architecture Matters: A Stream-Based Server



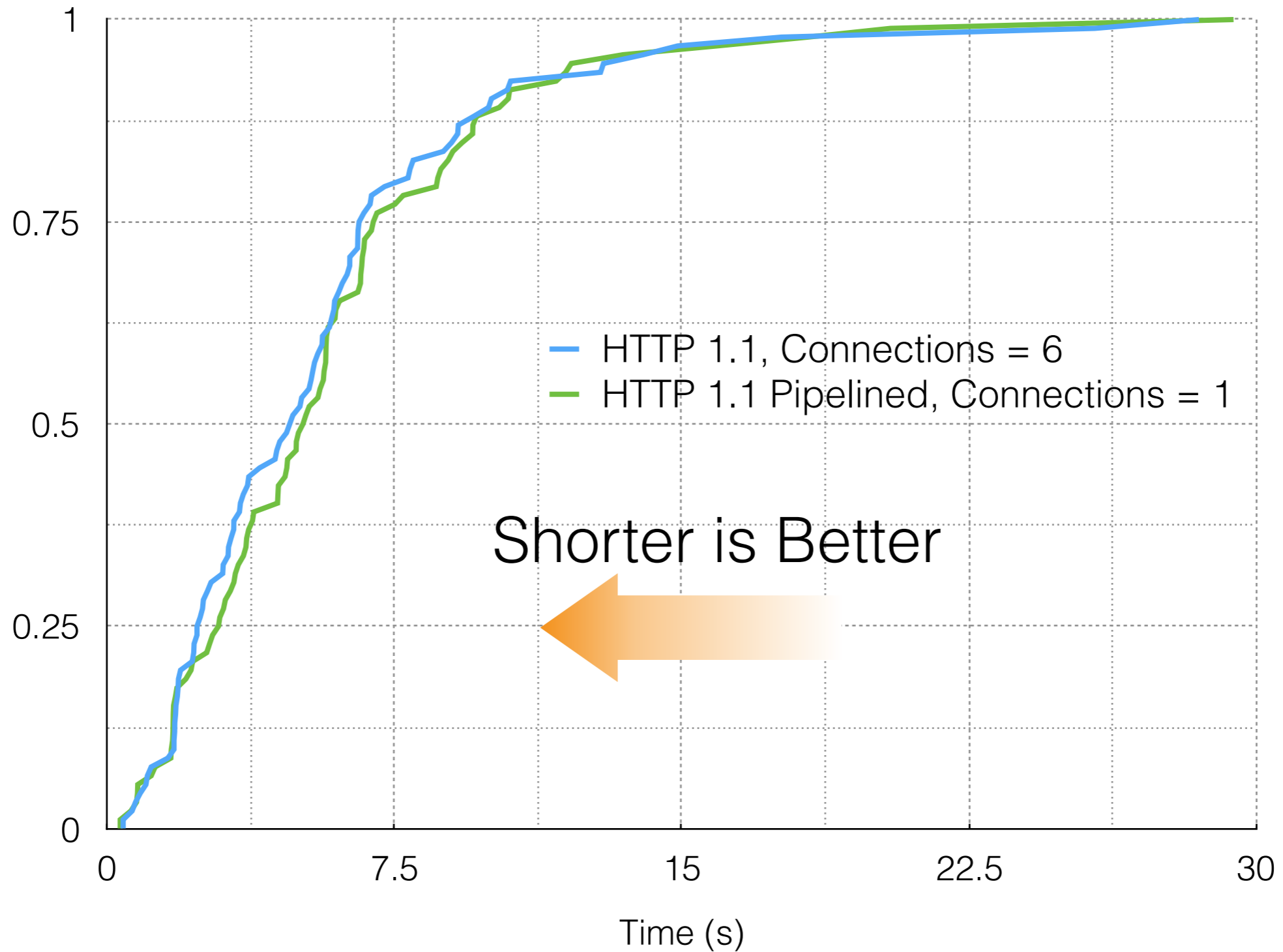
Server Architecture Matters: A Stream-Based Server

Bandwidth = 10 Mbps, RTT = 40 ms

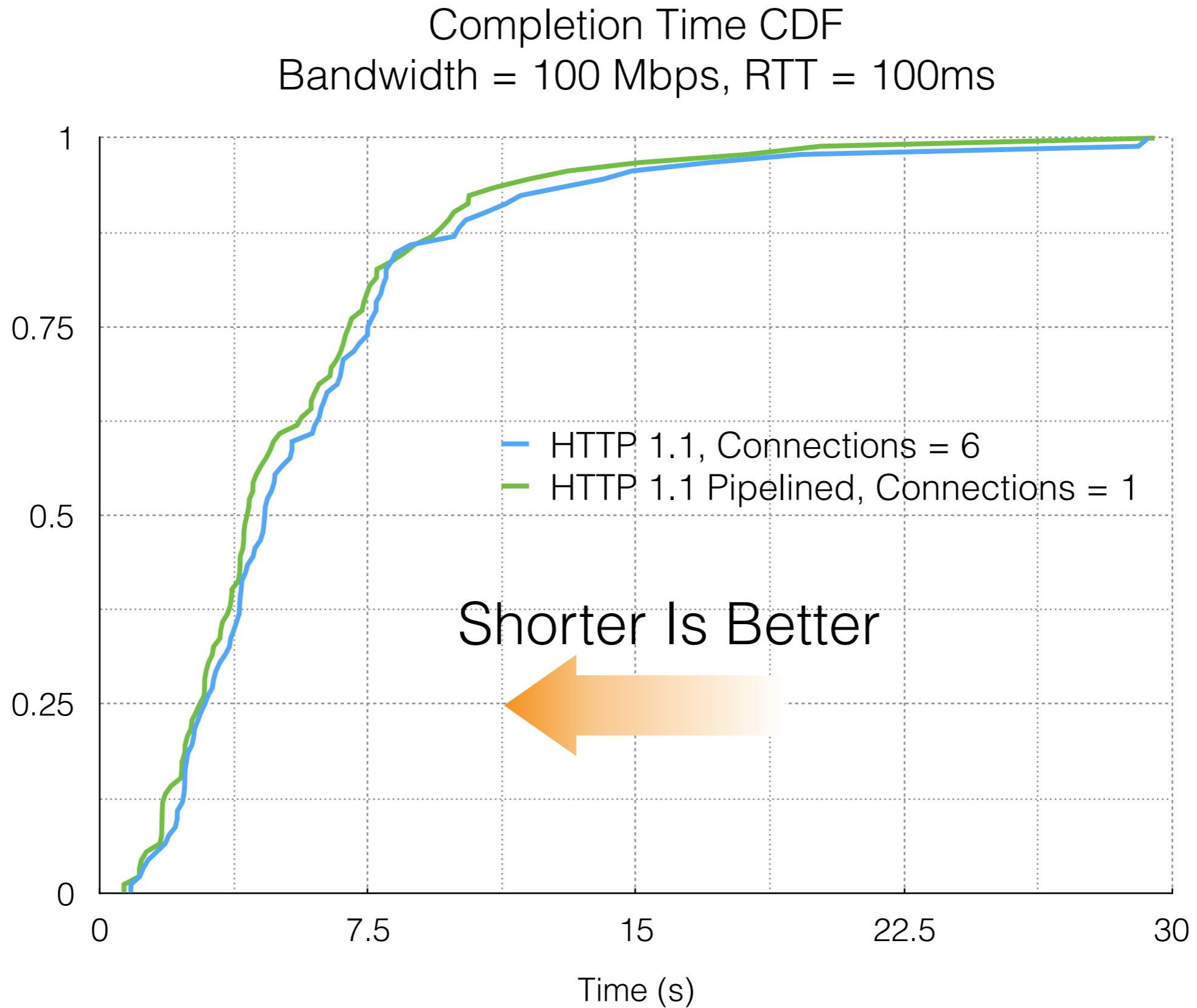


Which Protocol Works Best Depends on Speed of Network

Completion Time CDF
Bandwidth = 10 Mbps, RTT = 40ms

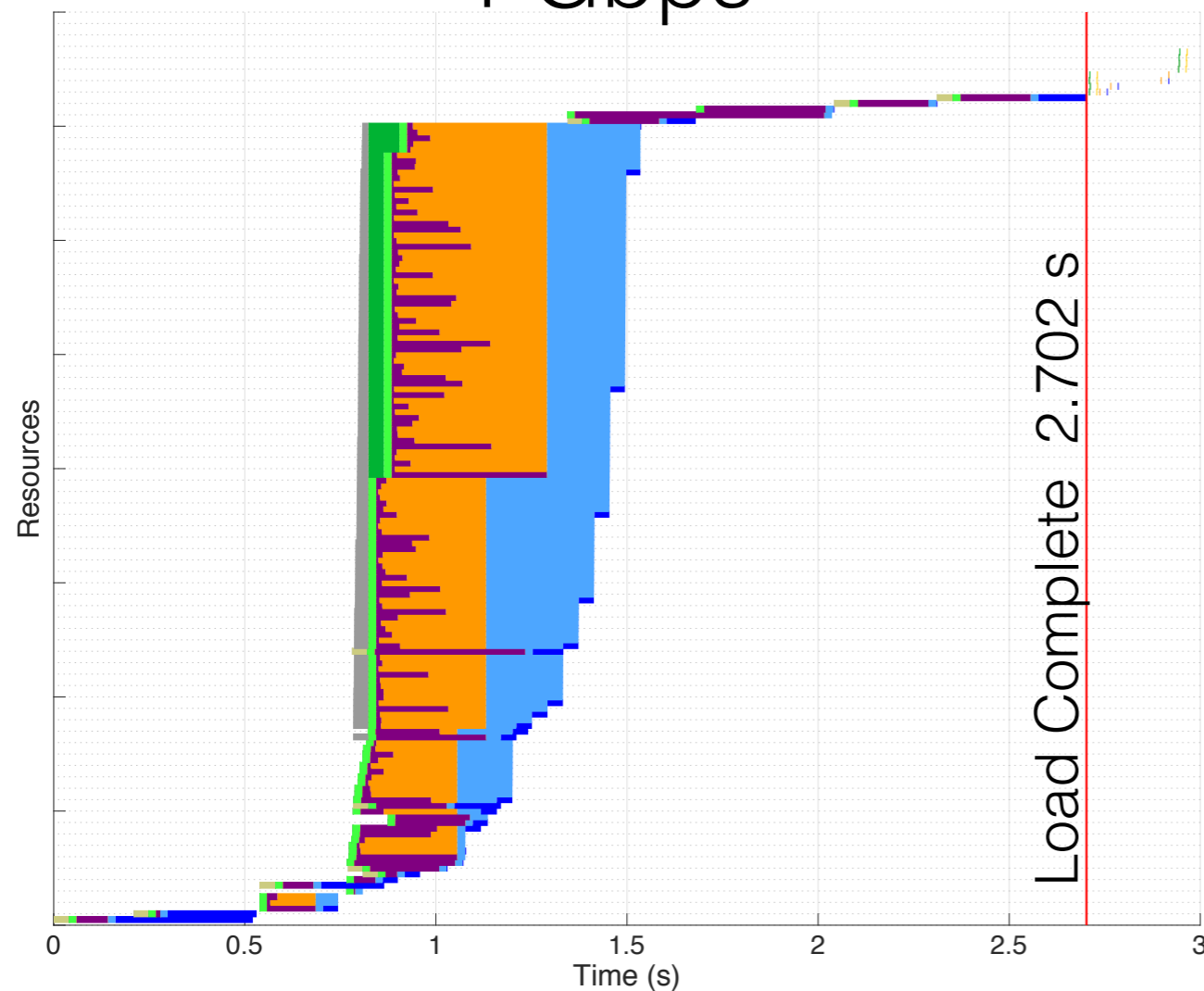


Which Protocol Works Best Depends on Speed of Network

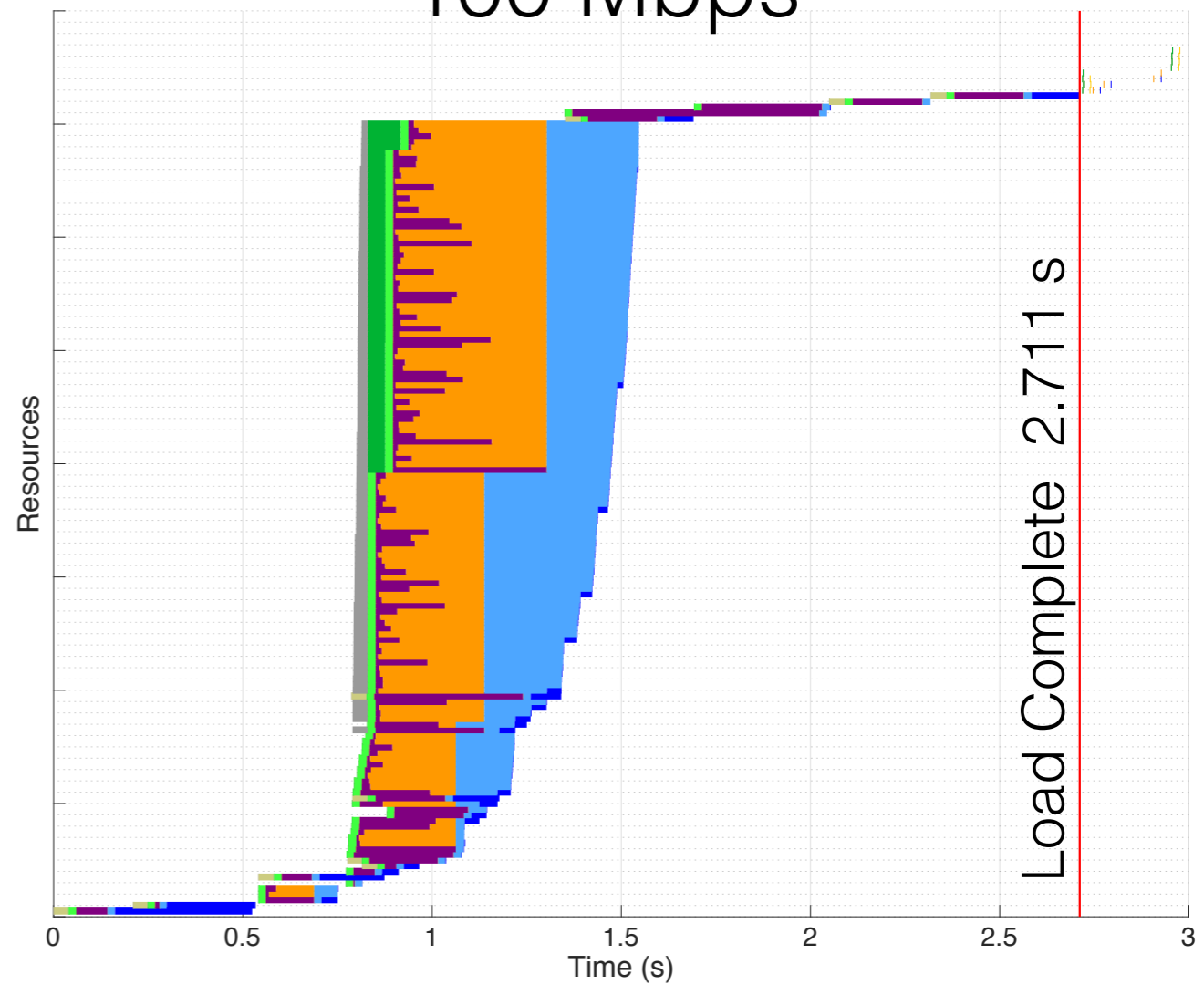


How Fast is Fast Enough?

1 Gbps

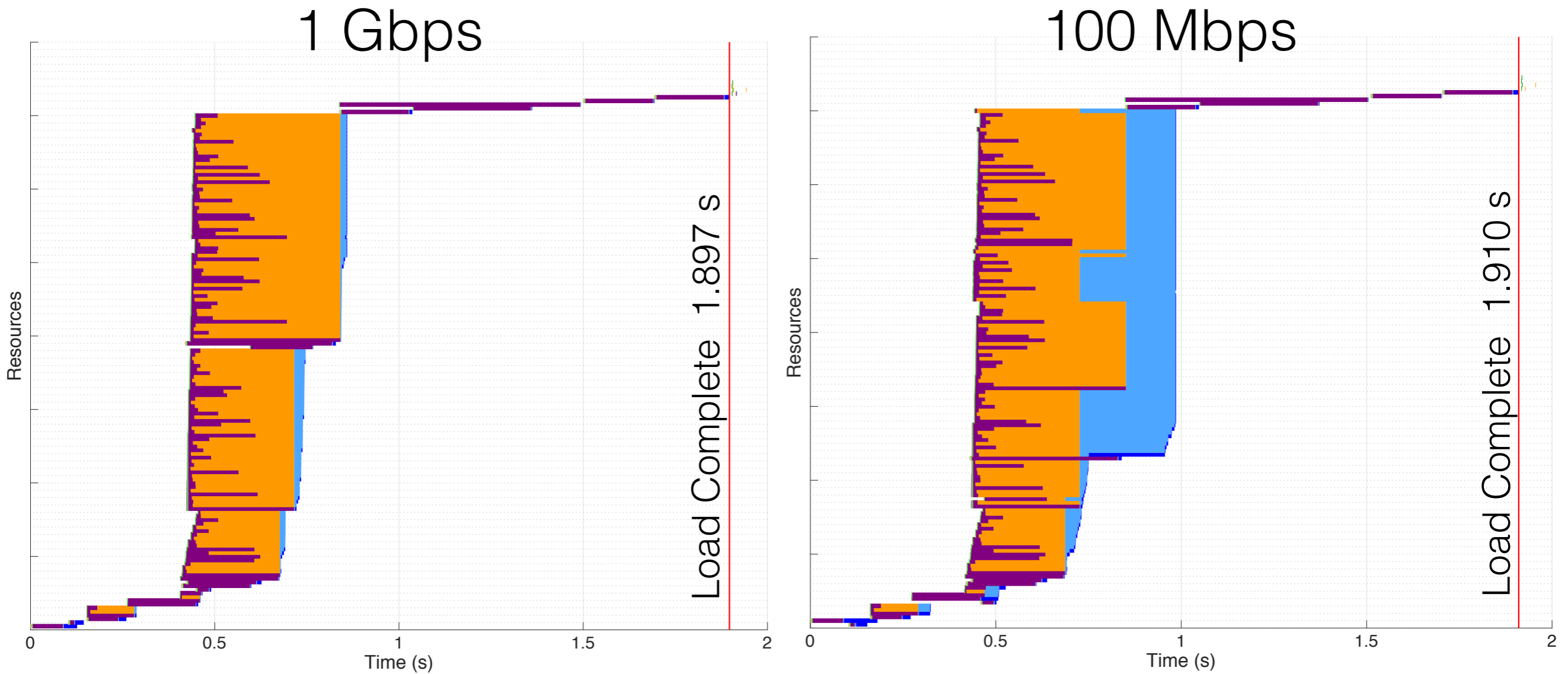


100 Mbps



Parallel Server amazon.co.uk Timeline,
RTT = 40 ms

How Fast is Fast Enough?



Parallel Server amazon.co.uk Timeline,
RTT = 4 ms

So, What Happens Next ... ?

- Add Multipath TCP to FROST.
- Critical path analysis to determine most important resources.
- Even simpler: prioritise objects based on MIME type.