A Client-Side Argument for Changing TCP Slow Start

Mike Belshe - mbelshe@chromium.org - Jan 11, 2010

Google™

Slow Start is a key part of TCP's congestion control algorithms...

http://www.faqs.org/rfcs/rfc2581.html

Google™

And, "*lack of attention to the dynamics of packet forwarding can result in severe service degradation or "Internet meltdown*"'.

- Sally Floyd, http://www.rfc-editor.org/rfc/rfc2914.txt

Google™

But Web Browsers today are inadvertently subverting slow start....

HTTP Background

HTTP originally was simple...

1. Open a connection
2. Send a request
3. Receive a response
4. When the server hangs up, you're done!

Couldn't be easier to implement...

But not very efficient on the network...

Google™

But Inefficiency Didn't Matter Much

- Web Pages were tiny
- Just a little text and a picture of a dog or two
- No CSS, JS, Video, etc.

As Web Pages grew, simple optimizations were built:

- Keep-Alives recycle TCP connections
- Better caching support
- etc

But....

Google™

Web Pages Continued to Grow

Technologies evolved.

- CSS, JavaScript, and image usage ballooned
- The average site in the Top-100 now makes ~40 requests per page (uncached)

Browsers were still limited to 2 connections per site (per spec).

Workarounds were easy...

Google™

Circumventing Slow Start

- Sending only two concurrent requests was a bottleneck
- Sites seeking speed needed more connections
  - "Subdomain sharding" was invented, since existing browsers would open 2 connections to each subdomain
- Eventually browsers kicked up connection limits
  - 2 connections per domain became 6

And voila! Slow Start is subverted.

```
   6 Connections Per Domain
*  6 Subdomains
*  3 initial cwnd per connection
------
  108 initial cwnd  (not exactly what RFC2581 suggests!)
```

Google™

Who is Wrong?

Is the TCP specification wrong?

- Is Slow Start too aggressive?
- Maybe the internet won't break if you turn it off...

Or maybe this is why we measure 1-1.5% packet loss on web pages today...

I don't know, but these are the facts.

Google™

Trying to Build a Better Protocol

SPDY attempts to solve some of HTTP's more fundamental problems:

- serialized requests
- not enough compression

And it is more efficient:

- Overall 40% reduction in packets
- 15% reduction in bytes transmitted

And *most* of the time it is faster.

Google™

Slow Start Gets In The Way

SPDY attempts to be efficient by using the network more intelligently.

Fewer TCP Connections.

But

- SPDY uses a single connection per domain.
- HTTP uses 6 connections per domain.

This gives HTTP an initial cwnd 6 times larger than an "efficient" protocol!

Google™

Example: The Slow Start Bottleneck in Action

In This Example

Network:

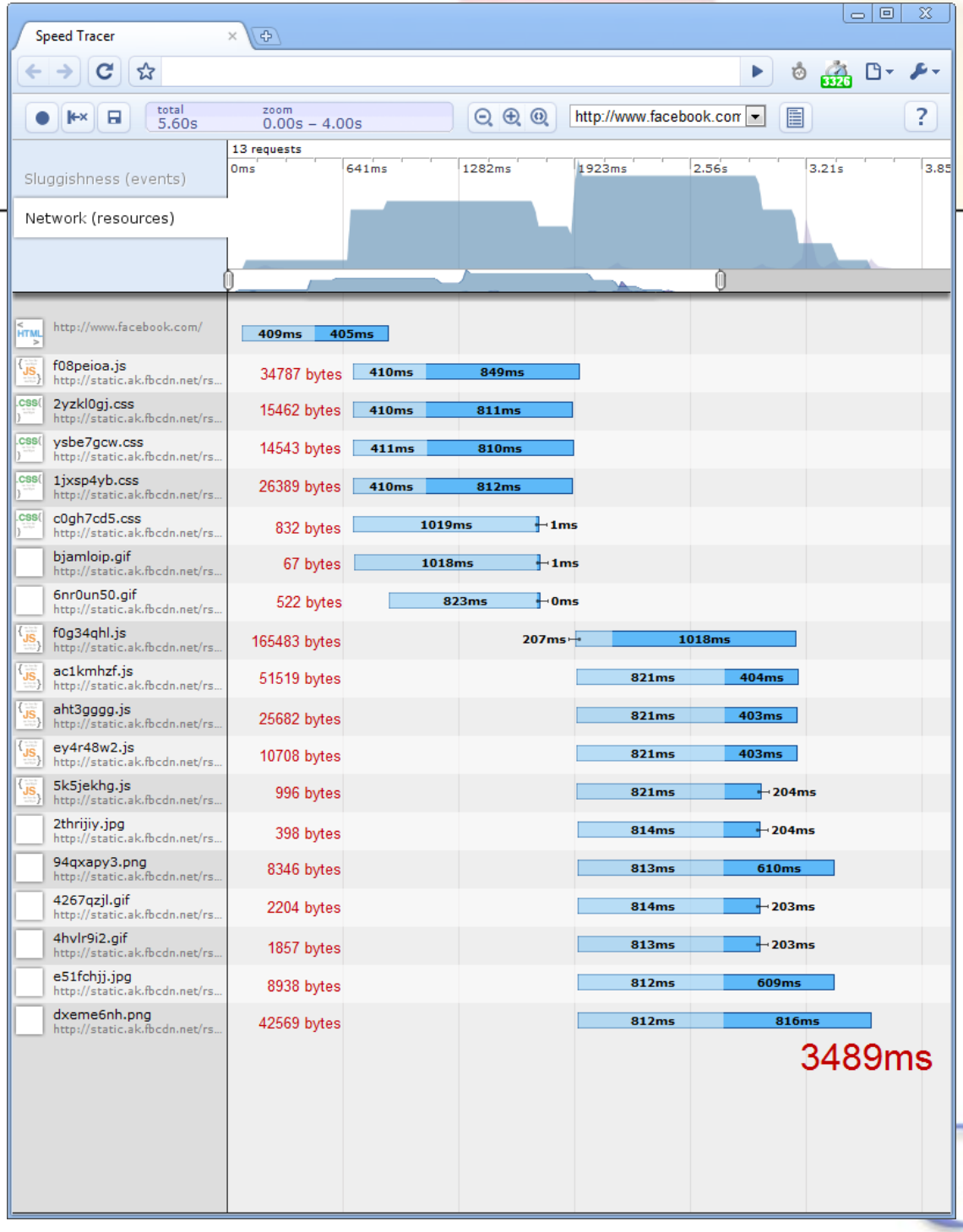
- 100Mbps
- 200ms RTT
- 0% loss

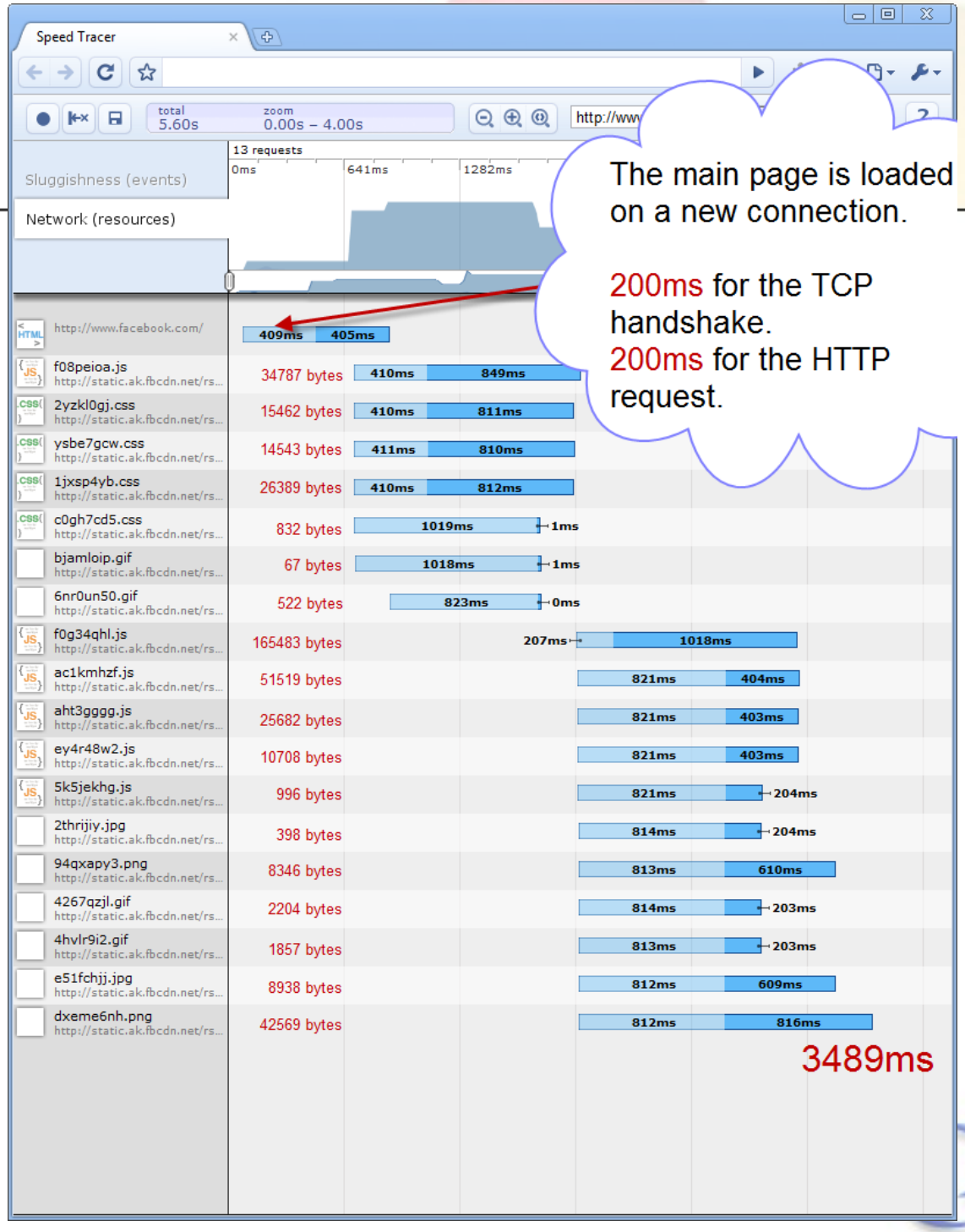
Browser

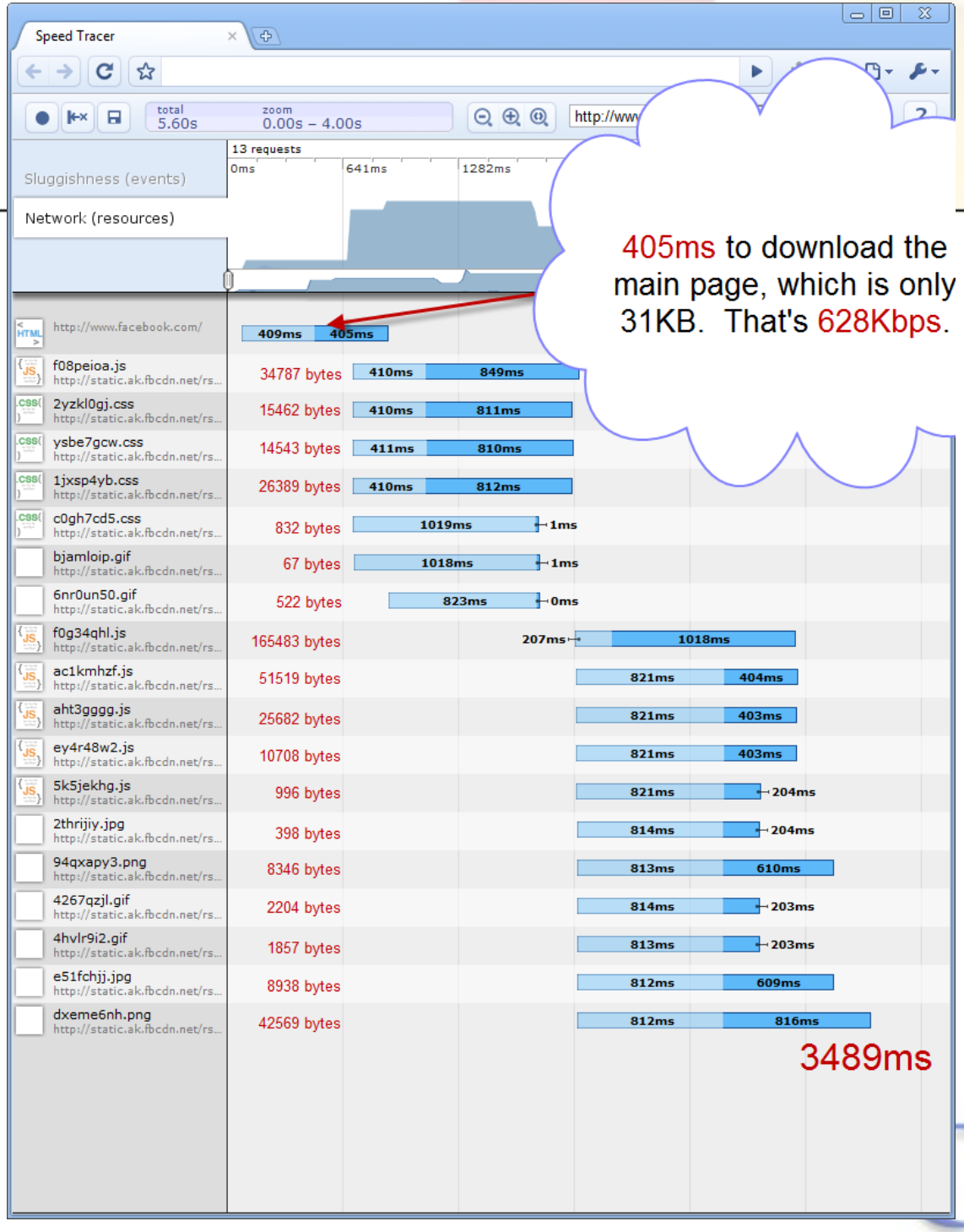
- Chrome
- Using SPDY, to load the page over a single connection
- No resources are cached

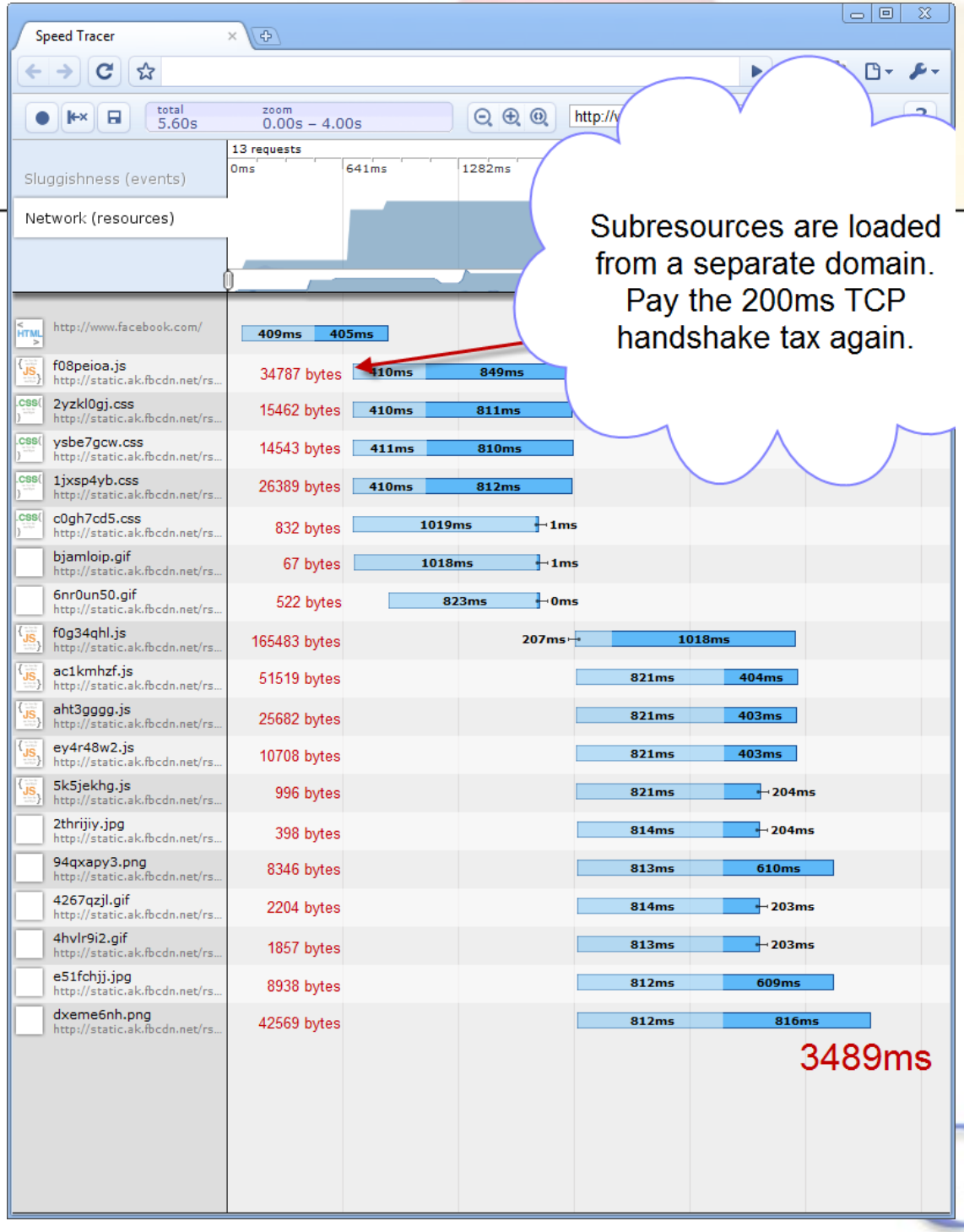
WebPage

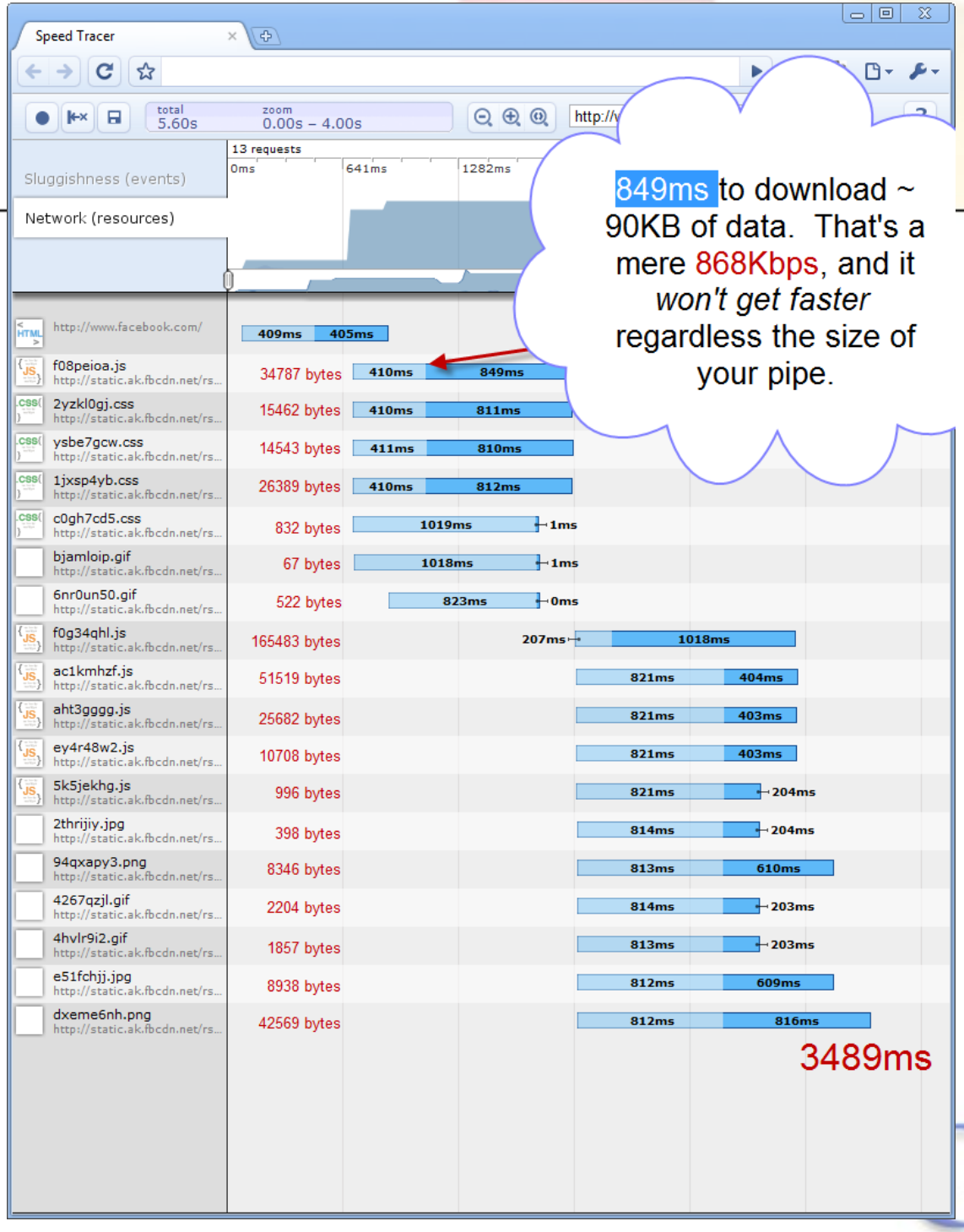
- http://www.facebook.com/
- But this can be witnessed on many many webpages.
- There is nothing wrong with the facebook page content.

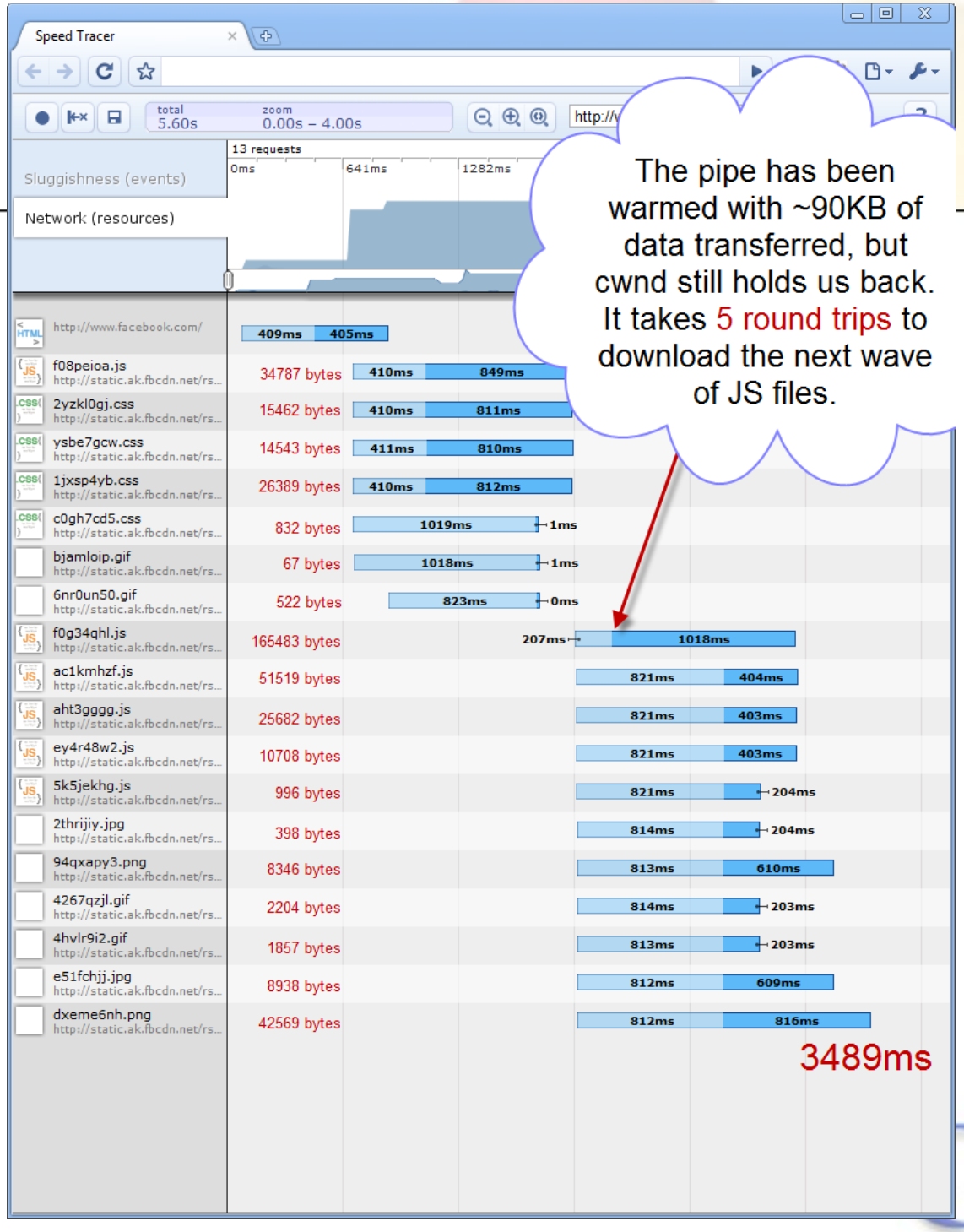

Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
http://www.facebook.com
13 requests
Sluggishness (events)
Network (resources)
0ms
641ms
1282ms
1923ms
2.56s
3.21s
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms


Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
13 requests
0ms
641ms
1282ms
Sluggishness (events)
Network (resources)
The main page is loaded on a new connection.
200ms for the TCP handshake.
200ms for the HTTP request.
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms


Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
13 requests
0ms
641ms
1282ms
Sluggishness (events)
Network (resources)
405ms to download the main page, which is only 31KB. That's 628Kbps.
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms

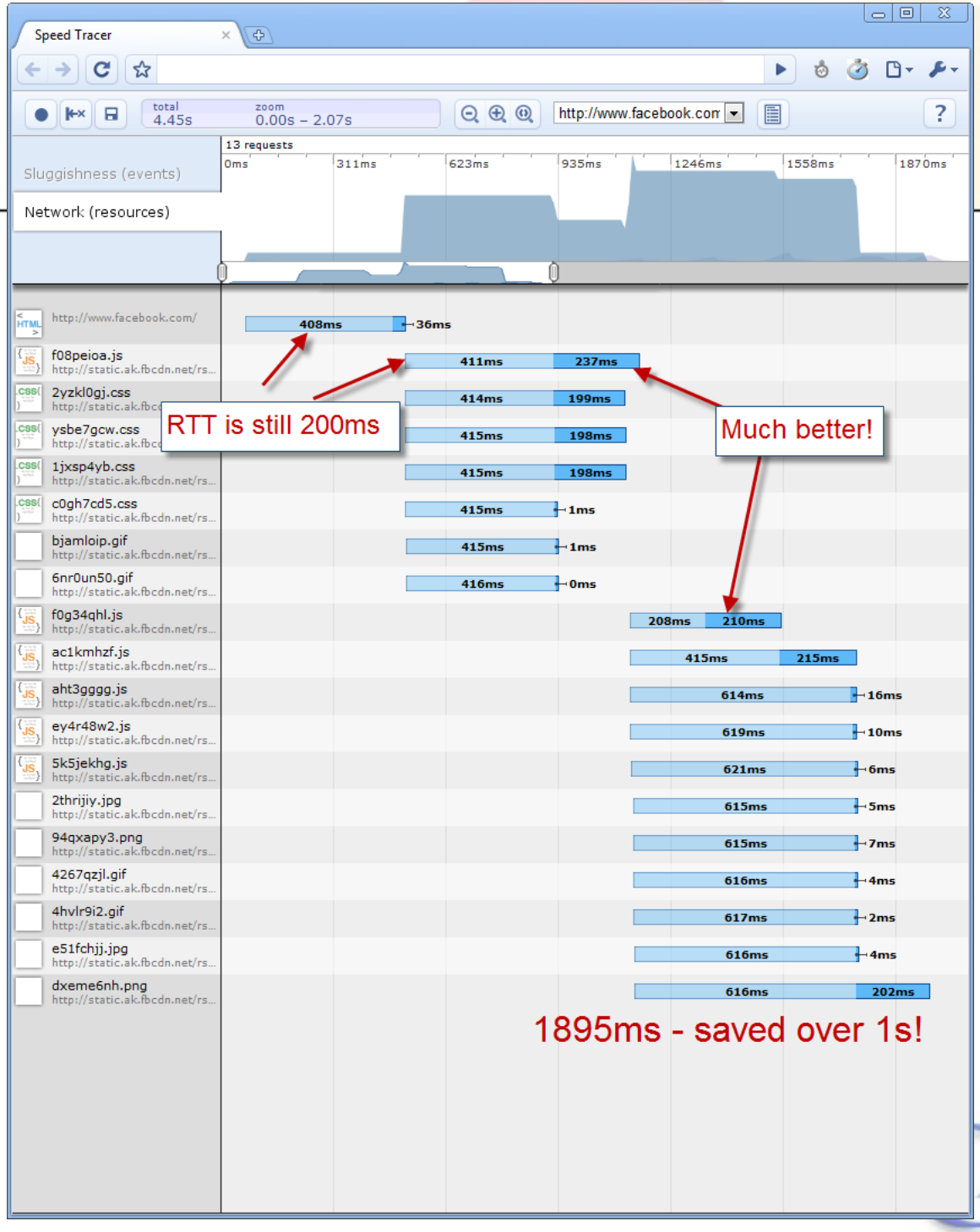

Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
13 requests
0ms
641ms
1282ms
Sluggishness (events)
Network (resources)
Subresources are loaded from a separate domain. Pay the 200ms TCP handshake tax again.
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms


Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
13 requests
0ms
641ms
1282ms
Sluggishness (events)
Network (resources)
849ms to download ~ 90KB of data. That's a mere 868Kbps, and it won't get faster regardless the size of your pipe.
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms

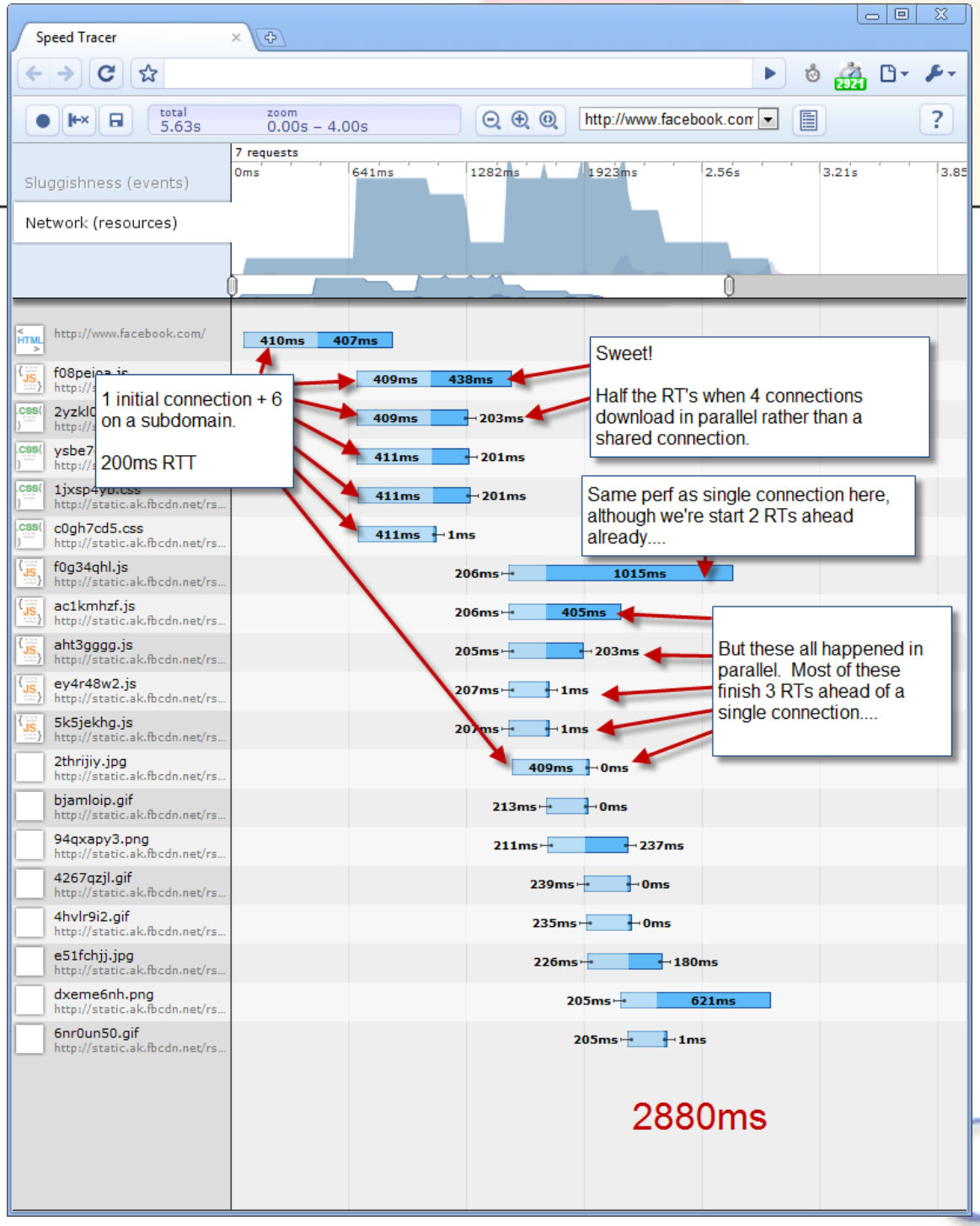

Speed Tracer
total 5.60s
zoom 0.00s – 4.00s
13 requests
0ms
641ms
1282ms
Sluggishness (events)
Network (resources)
The pipe has been warmed with ~90KB of data transferred, but cwnd still holds us back. It takes 5 round trips to download the next wave of JS files.
http://www.facebook.com/ 409ms 405ms
f08peioa.js 34787 bytes 410ms 849ms
2yzkl0gj.css 15462 bytes 410ms 811ms
ysbe7gcw.css 14543 bytes 411ms 810ms
1jxsp4yb.css 26389 bytes 410ms 812ms
c0gh7cd5.css 832 bytes 1019ms 1ms
bjamloip.gif 67 bytes 1018ms 1ms
6nr0un50.gif 522 bytes 823ms 0ms
f0g34qhl.js 165483 bytes 207ms 1018ms
ac1kmhzf.js 51519 bytes 821ms 404ms
aht3gggg.js 25682 bytes 821ms 403ms
ey4r48w2.js 10708 bytes 821ms 403ms
5k5jekhg.js 996 bytes 821ms 204ms
2thrijiy.jpg 398 bytes 814ms 204ms
94qxapy3.png 8346 bytes 813ms 610ms
4267qzjl.gif 2204 bytes 814ms 203ms
4hvlr9i2.gif 1857 bytes 813ms 203ms
e51fchjj.jpg 8938 bytes 812ms 609ms
dxeme6nh.png 42569 bytes 812ms 816ms
3489ms

What happens if we increase the initial cwnd to..... 18?

Google™


Speed Tracer
total 4.45s
zoom 0.00s – 2.07s
http://www.facebook.com
13 requests
Sluggishness (events)
Network (resources)
0ms
311ms
623ms
935ms
1246ms
1558ms
1870ms
http://www.facebook.com/
f08peioa.js
2yzkl0gj.css
ysbe7gcw.css
1jxsp4yb.css
c0gh7cd5.css
bjamloip.gif
6nr0un50.gif
f0g34qhl.js
ac1kmhzf.js
aht3gggg.js
ey4r48w2.js
5k5jekhg.js
2thrijiy.jpg
94qxapy3.png
4267qzjl.gif
4hvlr9i2.gif
e51fchjj.jpg
dxeme6nh.png
408ms
36ms
411ms
237ms
414ms
199ms
415ms
198ms
415ms
198ms
415ms
1ms
415ms
1ms
416ms
0ms
208ms
210ms
415ms
215ms
614ms
16ms
619ms
10ms
621ms
6ms
615ms
5ms
615ms
7ms
616ms
4ms
617ms
2ms
616ms
4ms
616ms
202ms
RTT is still 200ms
Much better!
1895ms - saved over 1s!

Compare to Traditional HTTP


Speed Tracer
total 5.63s
zoom 0.00s – 4.00s
http://www.facebook.com
7 requests
Sluggishness (events)
Network (resources)
0ms
641ms
1282ms
1923ms
2.56s
3.21s
http://www.facebook.com/
f08peiqa.js
2yzkl0.css
ysbe7.css
1jxsp4yb.css
c0gh7cd5.css
f0g34qhl.js
ac1kmhzf.js
aht3gggg.js
ey4r48w2.js
5k5jekhg.js
2thrijiy.jpg
bjamloip.gif
94qxapy3.png
4267qzjl.gif
4hvlr9i2.gif
e51fchjj.jpg
dxeme6nh.png
6nr0un50.gif
http://static.ak.fbcdn.net/rs...
410ms
407ms
409ms
438ms
409ms
203ms
411ms
201ms
411ms
201ms
411ms
1ms
206ms
1015ms
206ms
405ms
205ms
203ms
207ms
1ms
207ms
1ms
409ms
0ms
213ms
0ms
211ms
237ms
239ms
0ms
235ms
0ms
226ms
180ms
205ms
621ms
205ms
1ms
1 initial connection + 6 on a subdomain.
200ms RTT
Sweet!
Half the RT's when 4 connections download in parallel rather than a shared connection.
Same perf as single connection here, although we're start 2 RTs ahead already....
But these all happened in parallel. Most of these finish 3 RTs ahead of a single connection....
2880ms

Increasing Initial CWND?

It's not a matter of whether we should increase initial cwnd. *For HTTP, it is already increased.*

Data:
Sites are already using init-cwnd = 3.
HTTP uses 6 connections per domain, enabling init-cwnd = 18.
Sites use 3-8 subdomains, increasing init-cwnd to 54-144.

Google™

But Is It Realistic?

initcwnd is generally a global setting hidden inside TCP.

If we increase it globally, HTTP's init-cwnd is always at least 6X larger than it should be.

Even with SPDY, until sites stop using subdomain sharding, effective cwnd is higher than what TCP is attempting to enforce.

Will the internet at large increase their cwnd? Needs a kernel patch at the minimum - possibly a sockets API change.

Google™

Conclusions

- Any TCP-based protocol attempting to be maximally efficient on the network will use fewer TCP connections than HTTP.
- But using fewer connections is a handicap due to TCP's Slow Start on high latency links.
- Slow Start, while specified at ~2-4 pkts, has, in practice, already been increased far beyond that by use of multiple connections and subdomains.
- If TCP cannot increase initcwnd, the next generation HTTP protocol will likely not be TCP based.

Google™