How to improve PLT

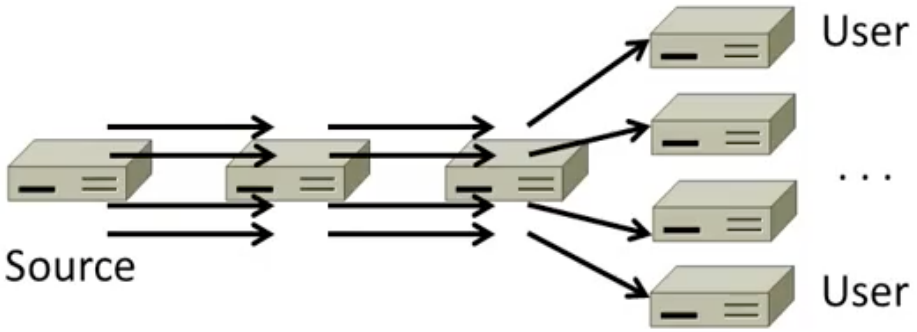
- Reduce content size for transfer
  - Smaller images, compression
- Change HTTP to make better use of available bandwidth
  - Persistent connections and pipelining
- Change HTTP to avoid repeated transfers of the same content
  - Caching and web-proxies
- Move content closer to the client
  - CDNs

Content Distribution Networks

- With the popularity of Web, traffic volumes grew tremendously
  - Increased load on popular web servers
  - Need for increased network bandwidth
  - Increase in PLT and poor user experience
- Browser and Proxy Caches help
  - Single client or clients in one organisation
- Place popular content near clients
  - Replicas
  - How is that possible?

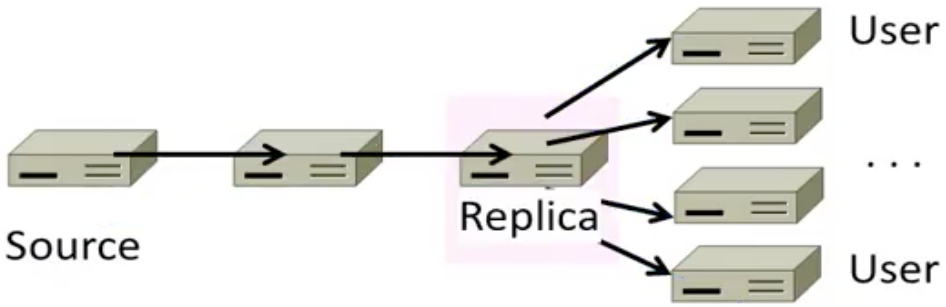
CDNs

- Sending content from the origin server to 4 users takes 4 x 3 = 12 "network Hops"



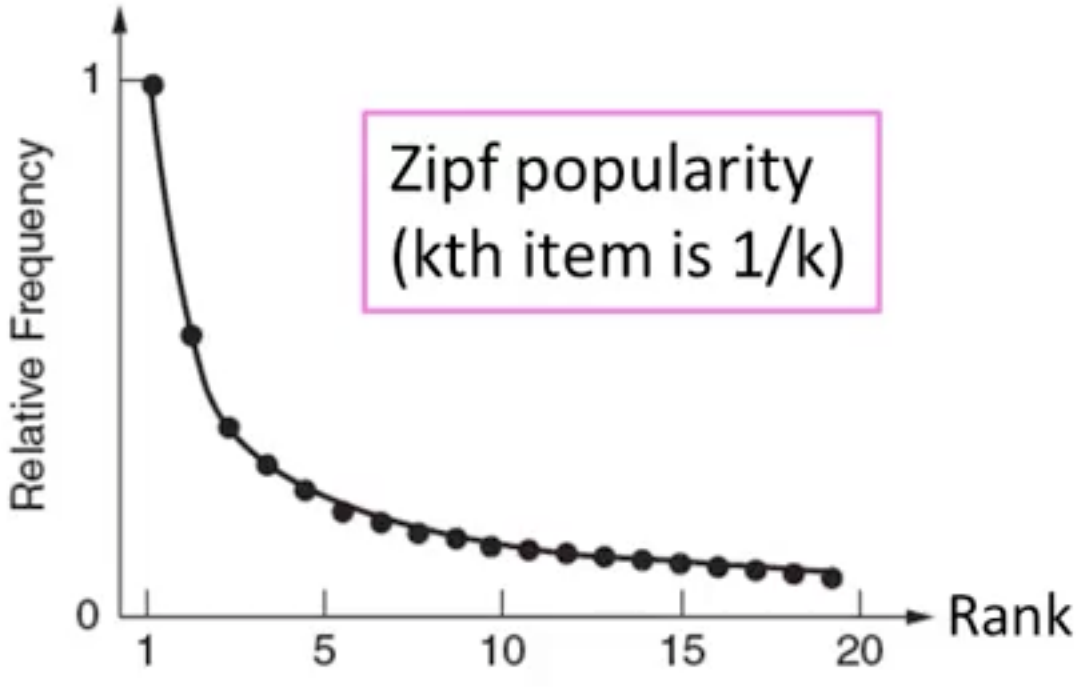
- Sending content via replicas takes only

4+2=6 "network hops"



Popularity of Content

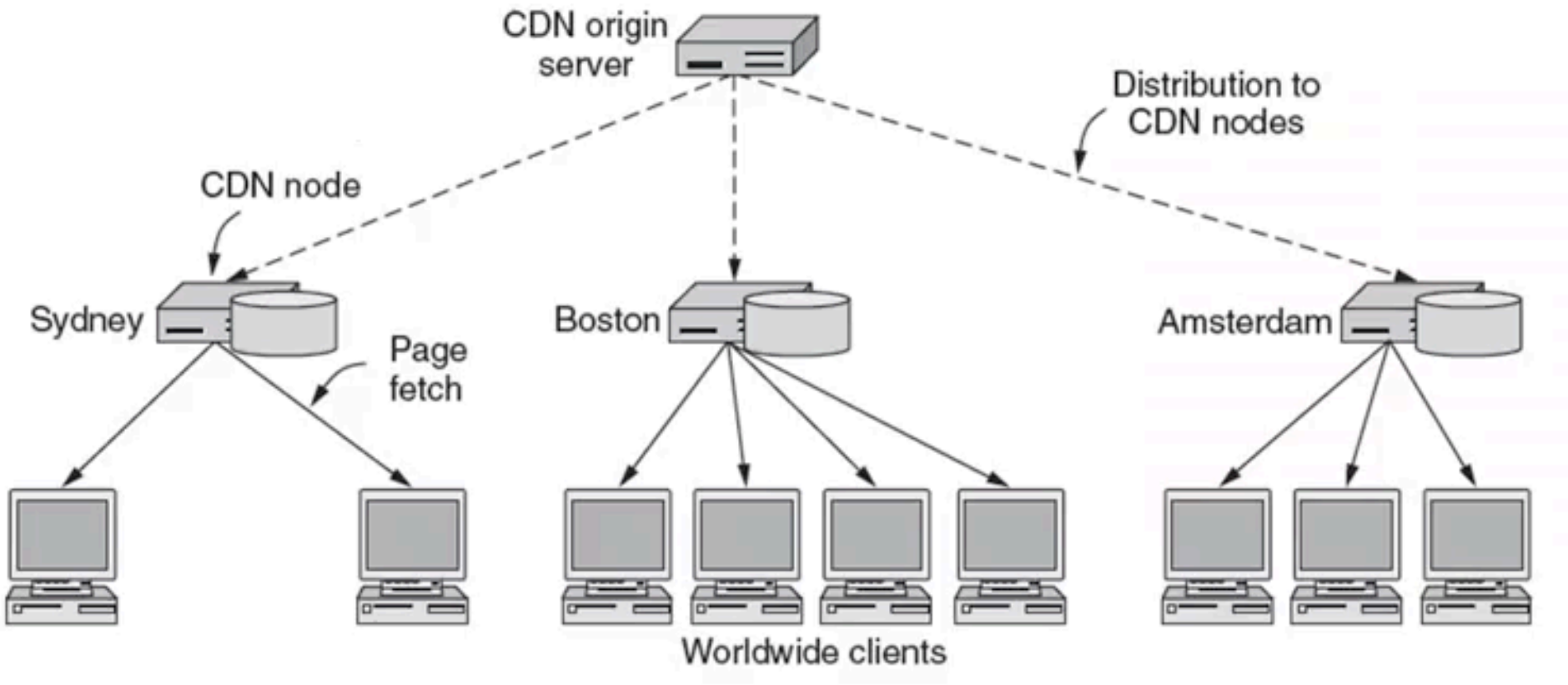
- Zipf's Law: few popular items and many unpopular ones
  - Relative frequency of words
  - Very common phenomenon in real world



Nuts and Bolts View

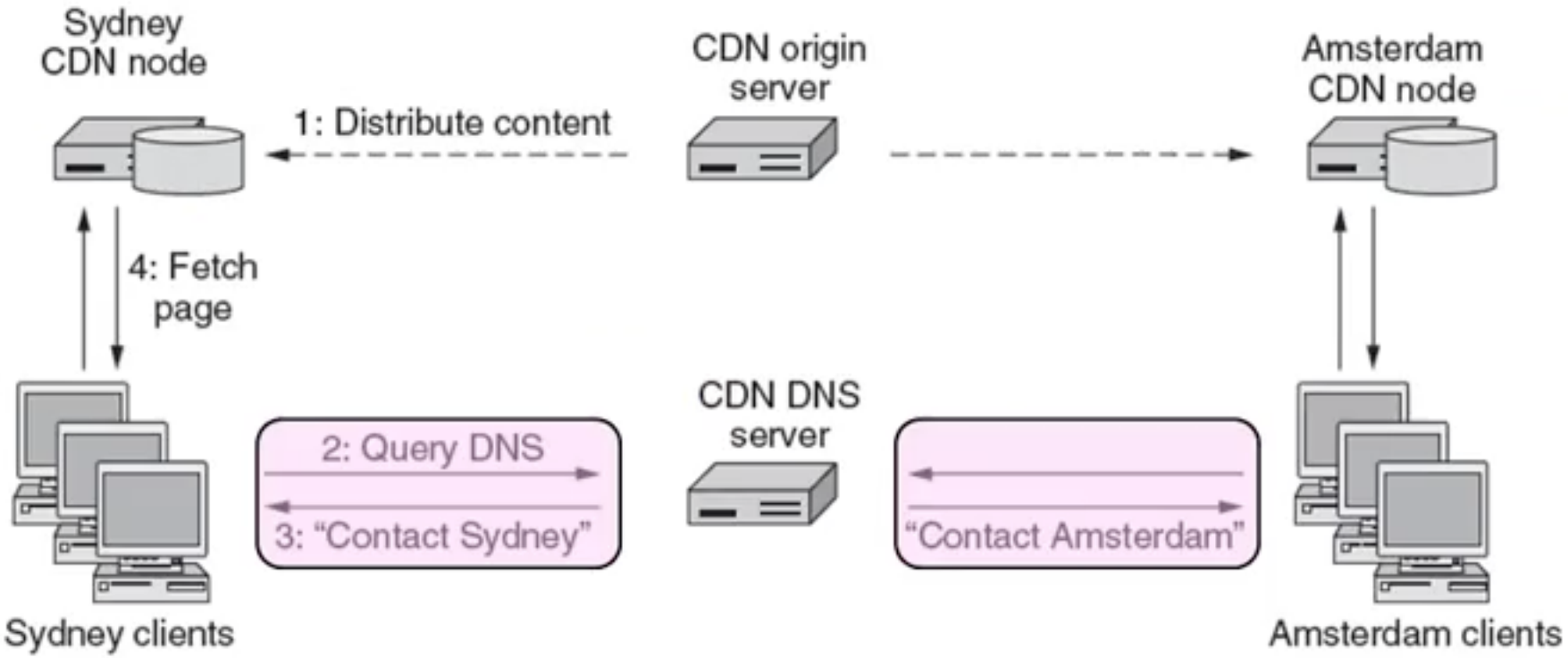
- Where to place replicas?
- How will a client find a "nearby" replica?
- How to keep replicas transparent to the client?
- CDNs rely on clever use of DNS

Nuts and Bolts View



Nuts and Bolts View

- DNS resolution of the site gives different answers to different clients
- Guides it towards the nearest CDN Node
- Depends on the client IP (or local NS), Reverse DNS ?



Business model for CDNs

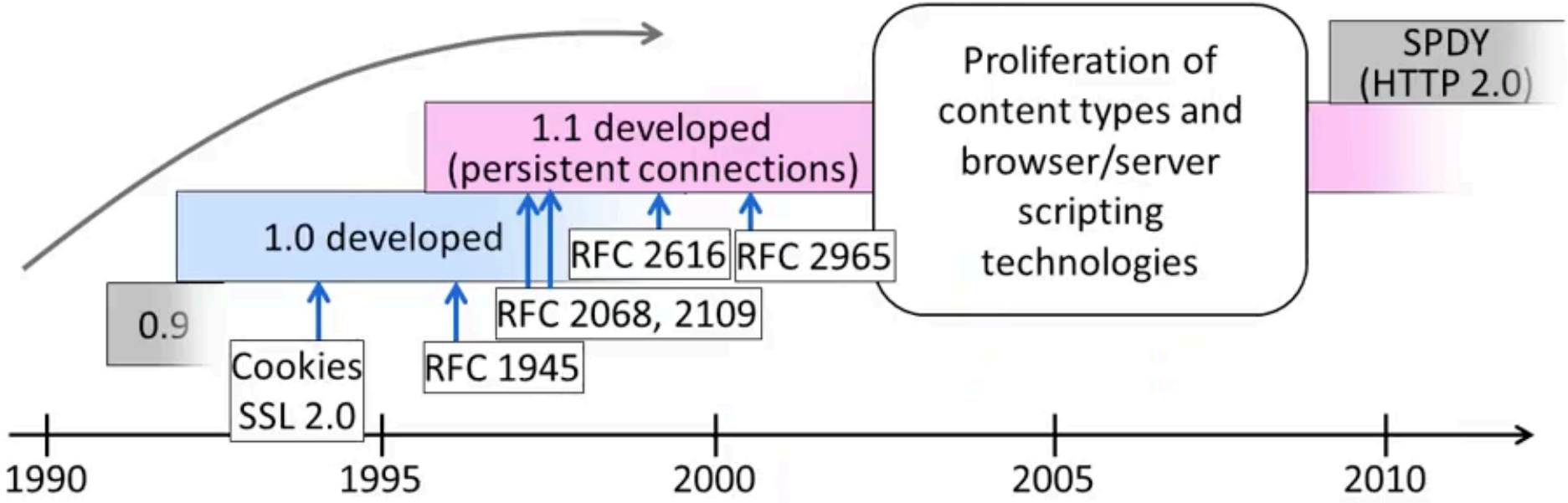
- CDN helpful for large e-commerce sites, video streaming and social networks
- A CDN operator gets paid by content providers (media companies, e-commerce vendors)
- In turn CDN pays ISPs, carriers for hosting its servers in their data centres
- ISPs gets paid and also saves on their SLAs by reducing traffic across the core
- Win-win situation

CDNs

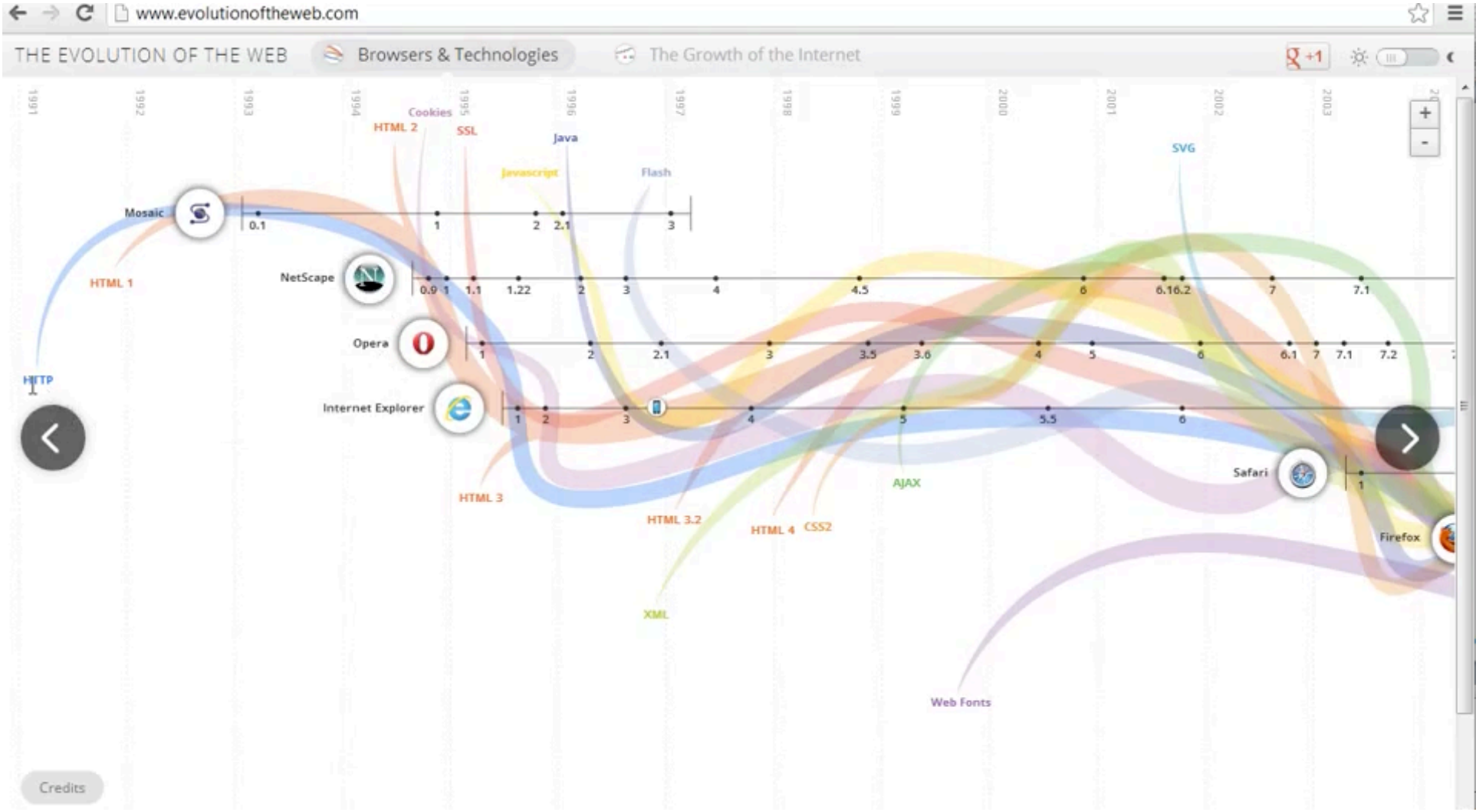
- Better performance and availability of content
- Offload content from content provider origin infrastructure
- Reduces impact of DoS attacks due to large distributed server infrastructure

Future of HTTP

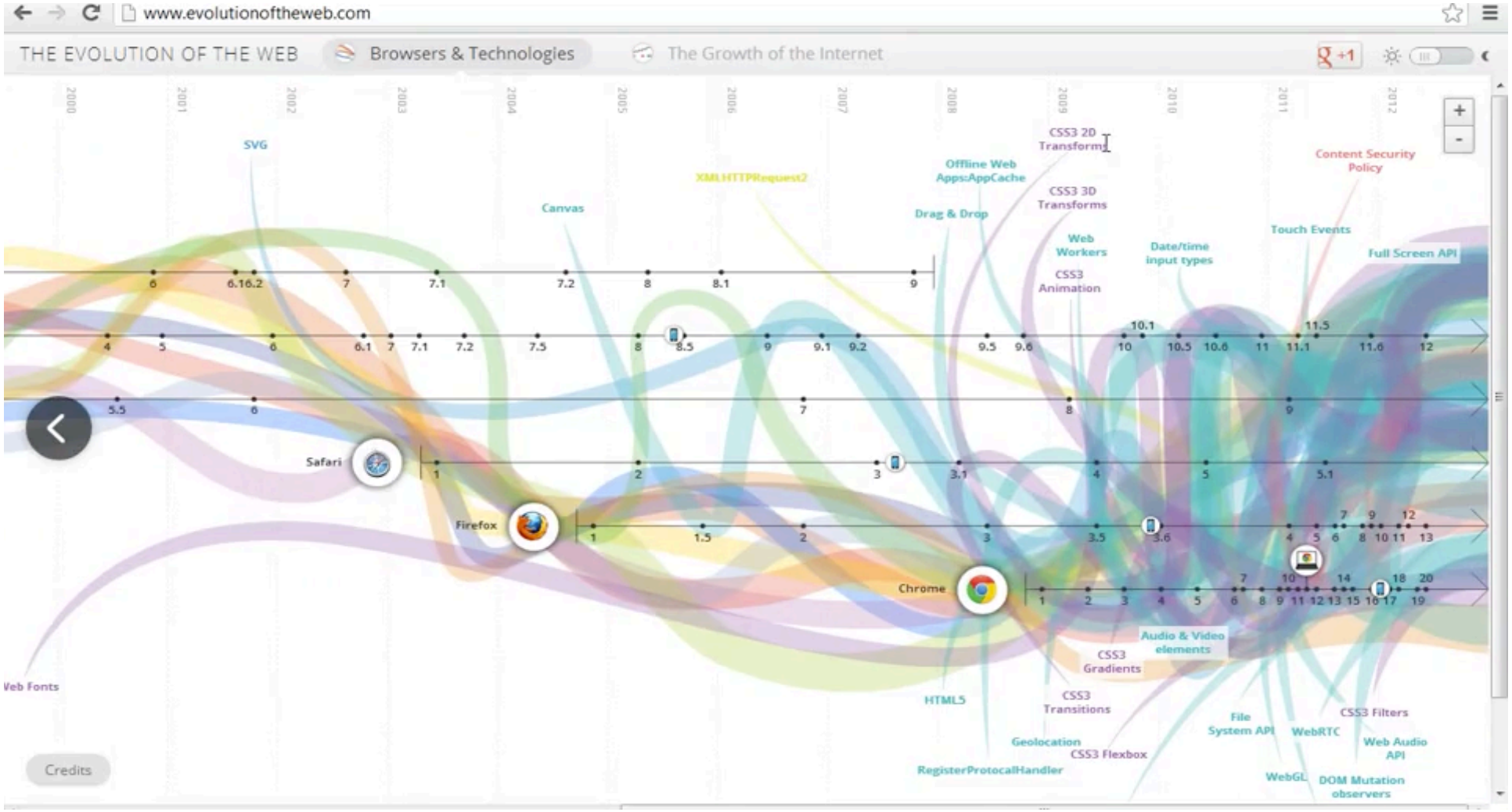
- Better use of the network
  - Google SPDY (speedy), HTTP/2.0
- Better content structure
  - mod_pagespeed server extension



Future of HTTP



Future of HTTP



Google SPDY

- A set of HTTP improvements
  - Supports multiplexed (parallel) HTTP requests on one TCP connection
  - Client priorities for parallel requests
  - Compressed HTTP headers
  - Server push of resources
- Basis for HTTP/2 effort
  - Default in later versions of Chrome and Firefox

mod_pagespeed

- The way pages are written and the contents are organised affects how quickly they load
  - Depends on the programmer
  - Can we automate this?
- Have the server re-write pages to help them load quickly
- Generate optimized pages on the fly
  - optimize javascript
  - Flatten multi-level CSS files
  - Resize images depending on the client
  - 100s of specific rules