Improving the Efficiency of HTTP Caching by Client-Cooperation

Chris Drechsler

Chair for Communication Networks

Chemnitz University of Technology

Outline

- Introduction
- Problem Statement
- Solution Approach
- Evaluation
- Conclusion

Introduction

- HTTP traffic accounts for more than 50 % of the whole Internet traffic and is still rising → high costs for network operators
- Solution for HTTP traffic reduction: caching of frequently requested content
- According to several studies the potential for HTTP traffic reduction by caching is up to 68%
- Problem: low efficiency of today's HTTP caches (less than 10 %)

→ Solutions to improve the caching efficiency required

Problem Statement

- Reasons for today’s low caching efficiency
  - Identification of resources via URLs only → same content might be available under different URLs and is not identified as identical by the cache
    - example:

      http://s1.videoportal.com/PopularVideo.webm

      vs.

      http://s2.videoportal.com/PopularVideo.webm
  - Personalization of HTTP messages
  - Explicit suppression of caching by content producers

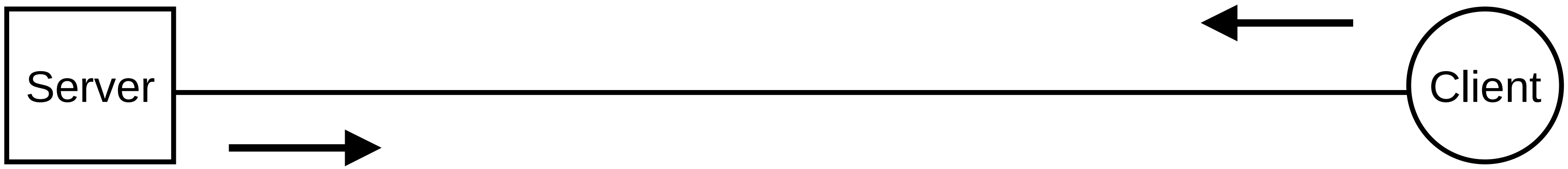
Solution Approach

- Our solution approach consists of 3 basic improvements:
  - HTTP header field extension
  - Modified cache operation
  - Cache size extension by client cooperation

Solution Approach - HTTP Header Field Extension

- HTTP header field extension:

```
GET /videos/PopularVideo.webm HTTP/1.1
Host: example.com
```

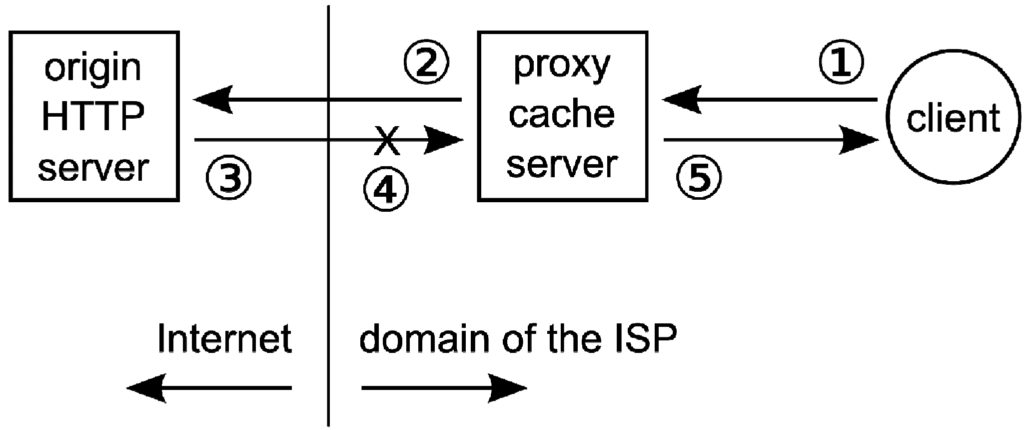


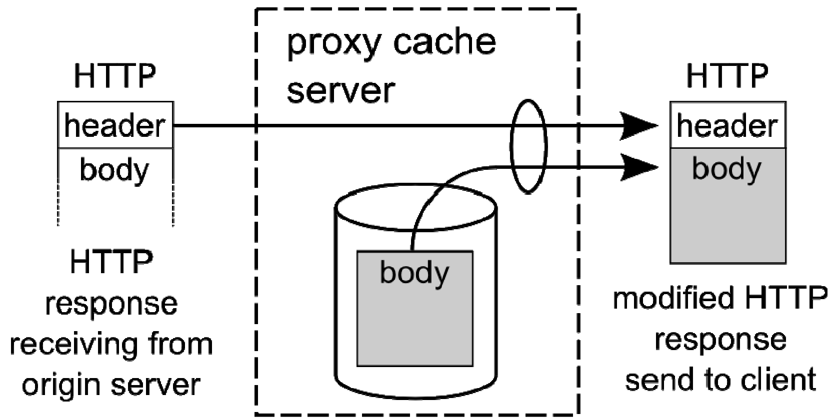

```
HTTP/1.1 200 OK
Date: Fri, 11 Nov 2011 11:11:11 GMT
Cache-NT: sha256=7ab53f24d8c96d1cc87452a6b113 …

<HTTP Body>
```

Solution Approach - Modified Cache Operation

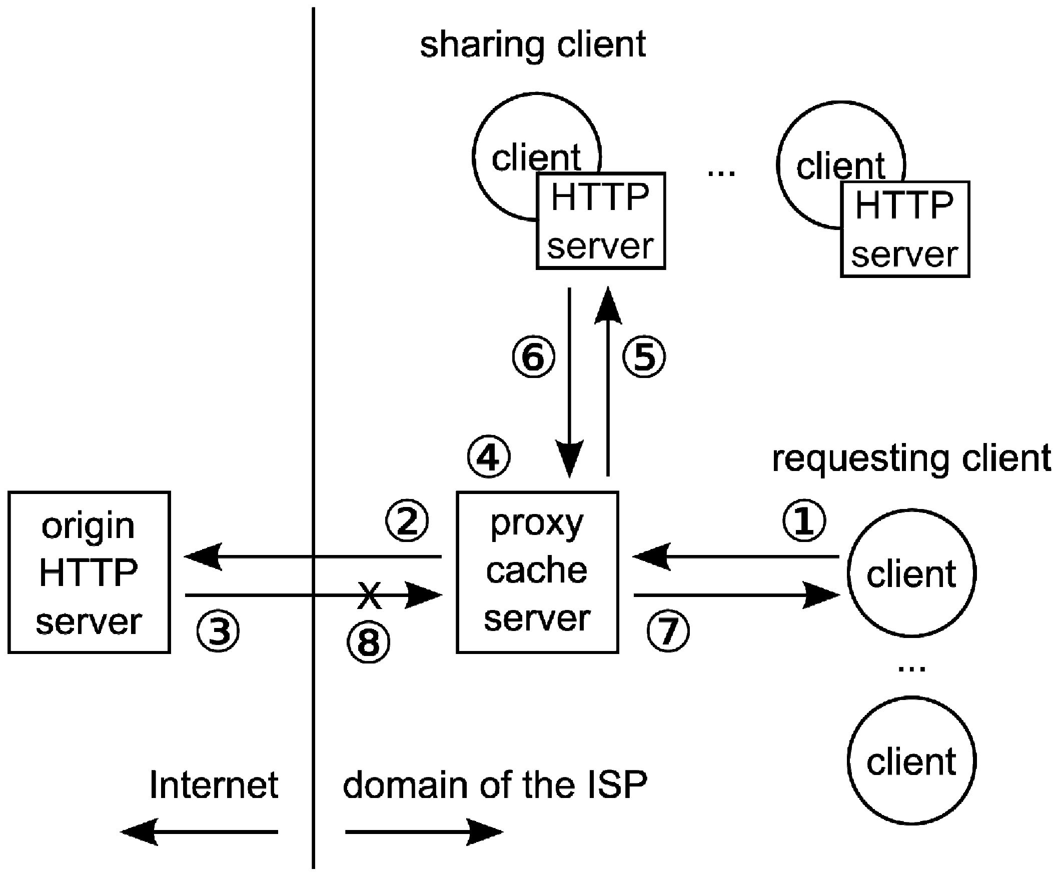
- Modified cache operation:





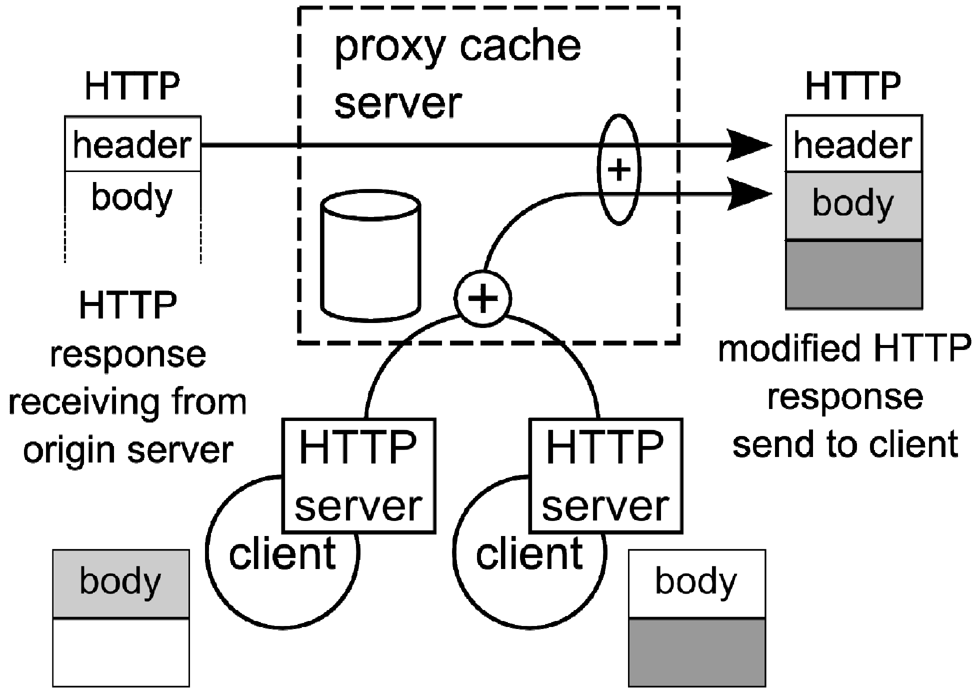
Solution Approach - Cache Size Ext. by Client Cooperation

- Clients can share their (downloaded) content within the operator's domain



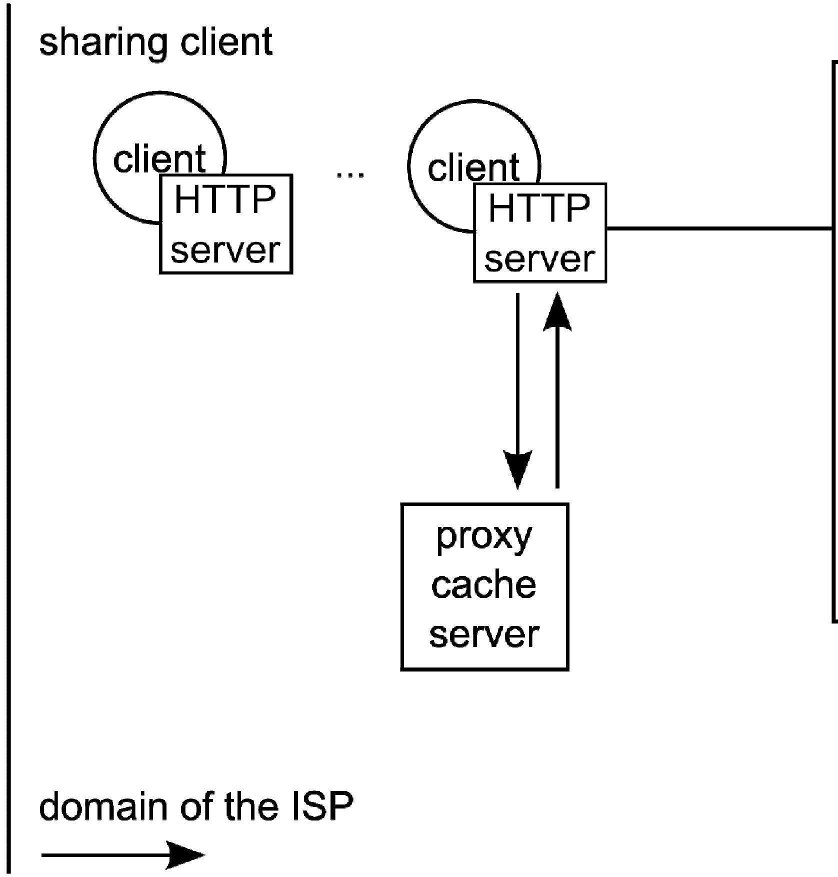
Solution Approach - Cache Size Ext. by Client Cooperation

- Concatenation of HTTP messages: the proxy cache server can request different pieces of the content resource from several sharing clients (via HTTP range request) and sends the concatenated HTTP response to the requesting client



Solution Approach - Cache Size Ext. by Client Cooperation

- The proxy cache server knows about the content resources on the clients by regularly querying each sharing client for its ClientCacheIndex → centralized index table




```xml
<?xml version="1.0" encoding="UTF-8"?>
<ClientCacheIndex>
   <CacheEntry>
      <SHA256ID>e70d4...92dbb<\SHA256ID>
      <ByteLength>4621287<\ByteLength>
   <\CacheEntry>
   <CacheEntry>
      <SHA256ID>d46f6...dd3a2<\SHA256ID>
      <ByteLength>2573648<\ByteLength>
   <\CacheEntry>
...
</ClientCacheIndex>
```

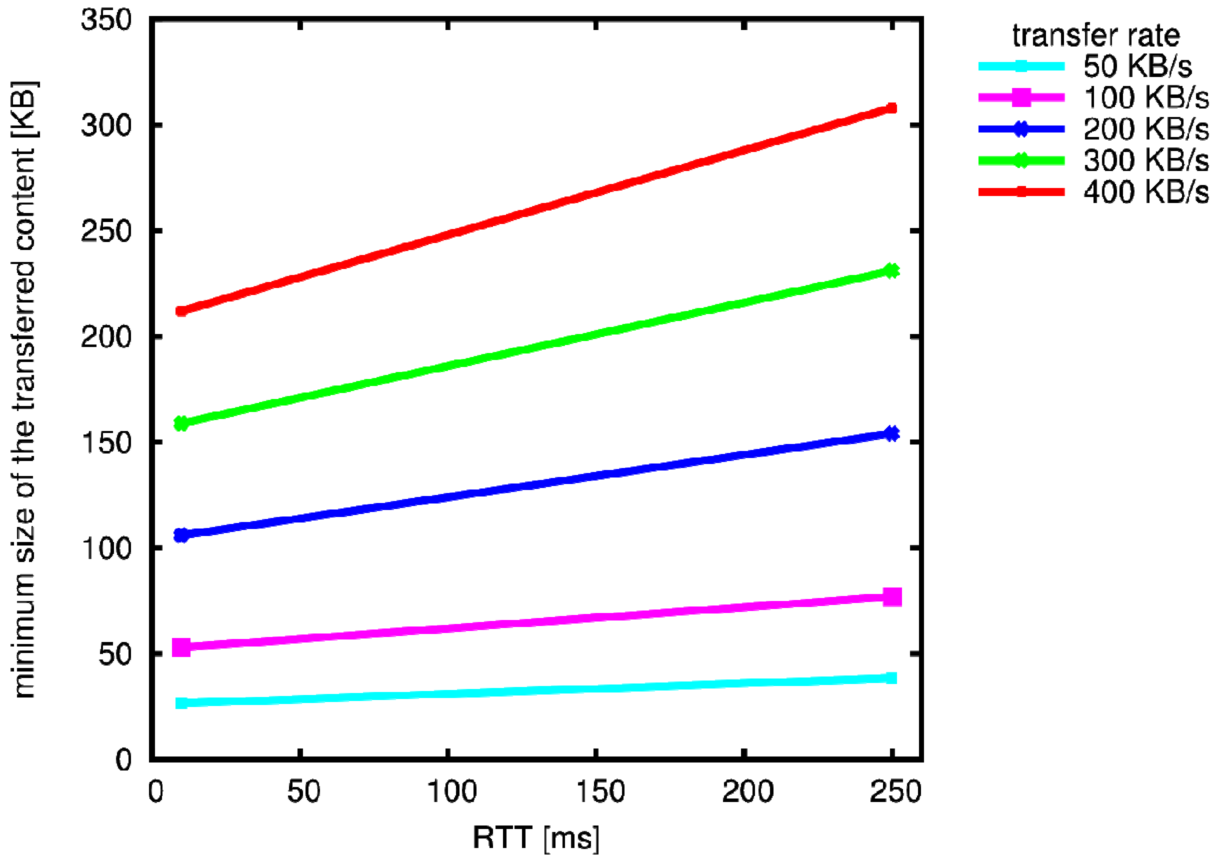
Evaluation - Critical Resource Size

- Question: is it always beneficial to stop the HTTP transfer if the requested resource is available within the ISPs domain?
- Aborting a transfer via the client does not stop sending data by the server immediately
- Aborting the HTTP transfer with the origin HTTP server and arranging a new transfer with the sharing client costs some time

→ the critical (minimum) resource size depends on the transfer rate and the RTT between origin HTTP server and proxy cache server

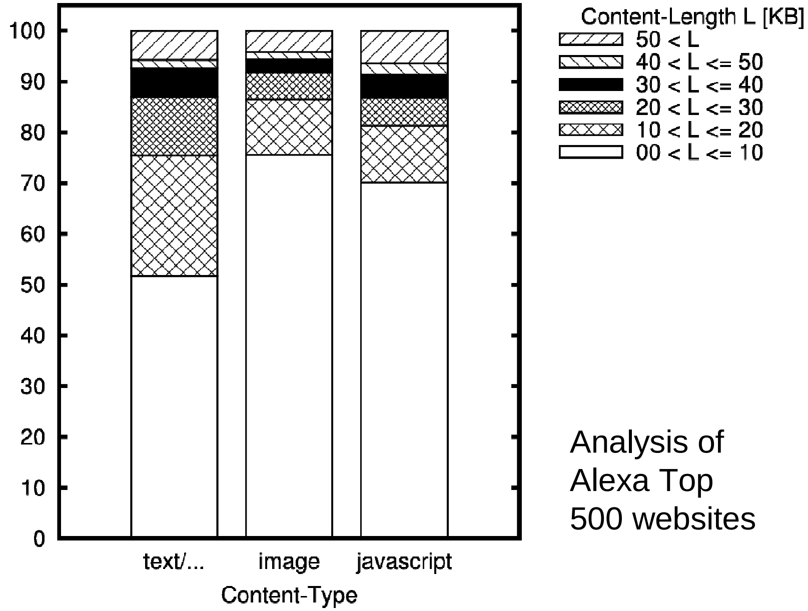
Evaluation - Critical Resource Size

- Critical (minimum) resource size vs. RTT and transfer rate



Evaluation - Caching Efficiency Improvement

- The gain in caching efficiency through our method depends on the content size (critical resource size) → analysis of real HTTP traffic wrt. content size
  - more than 80 % of the HTTP responses with content type *text, image and javascript* have a content length below 30 Kbyte → no caching efficiency improvement by our method
  - HTTP responses with video, audio, application/download content usually have content lengths larger than 0.5 Mbyte on (their share of the total transferred bytes (HTTP) is larger than 50%) → caching efficiency improvement possible
- Result: cache Byte hit rate about 34 %



Analysis of Alexa Top 500 websites

HTTP traffic	% avg. traffic
http/video	**31,9**
http/text-image	**25,9**
http/download	**16,2**
http/javascript	**5,8**
http/audio	**5,5**
...	**...**

HTTP traffic break-down [1]

Conclusion

- Contribution: improving the efficiency of HTTP caching
- Three basic concepts:
  - HTTP header field extension
  - Modified cache operation
  - Cache size extension by client cooperation
- Future work:
  - Implementation in a demo setup
  - Further performance analysis (scaling, overhead, timing behaviour, processing power, …)