Distributed cooperative HTTP Caching in Mobile Networks

Chris Drechsler, Gerd Windisch

Chair for Communication Networks

Chemnitz University of Technology

Outline

- Introduction
- Improved HTTP Caching Method
- Distributed Caching Architecture
- Distributed Cache Operation
- Summary

Introduction

- Key challenge for mobile network operators:
  - tremendous increase in mobile data traffic (dominant protocol: HTTP)

- Solution for HTTP traffic reduction in RAN and core:

  → **Caching at eNodeB site**

- Advantages:
  - no access to GTP-tunnel (S1-interface) required
  - access transport cost savings (compared to centralized caching at S/P-GW)
  - QoS/QoE improvement
- Disadvantages:
  - small population size (at eNodeBs) → low hit rate (caching efficiency)
  - higher cost for distributed cache deployment (smart proxies at eNodeBs)

Introduction

- Motivation:
  - increase caching efficiency → improved HTTP caching method
  - minimize cost of cache deployment
    - use of storage on UEs for building a distributed cache → free of charge from the operators perspective
    - minimal additional functionality in network elements

Outline

- Introduction
- **Improved HTTP Caching Method**
- Distributed Caching Architecture
- Distributed Cache Operation
- Summary

Improved HTTP Caching - HTTP Caching Efficiency

- Estimated efficiency potential of HTTP caching:
  - up to 68% HTTP traffic reduction (byte hit rate, BHR)
- Caching efficiency observed today:
  - only 10-20% (byte hit rate)
- Reasons for low caching efficiency:
  - difficult detection of duplicate payloads, example:

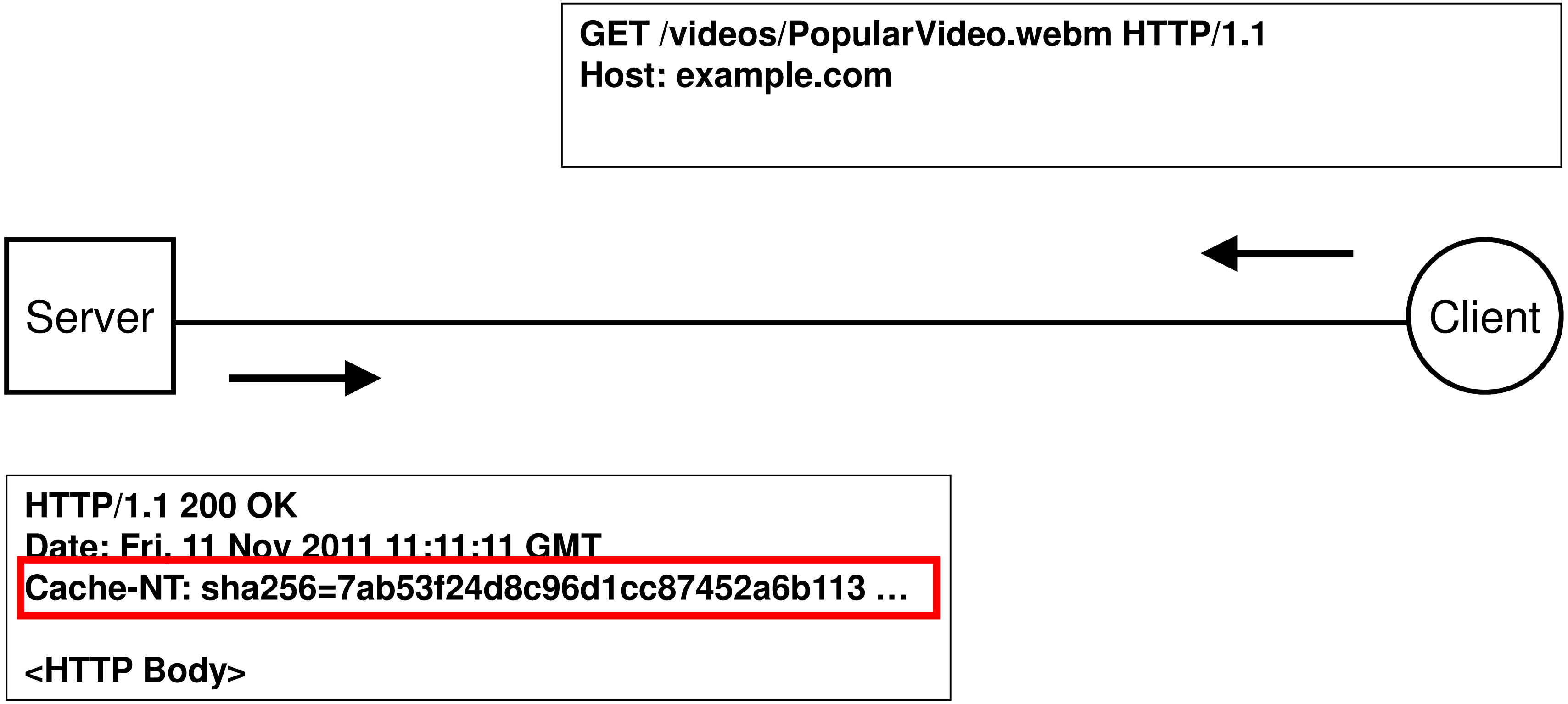
    http://s1.videoportal.com/PopularVideo.webm?userid=1111 vs.

    http://s2.videoportal.com/PopularVideo.webm?userid=2222
  - personalization
  - explicit suppression of caching by content producers
  - too small cache sizes

- → new caching method to improve the caching efficiency

Improved HTTP Caching - Basic Concept

- HTTP header field extension:



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

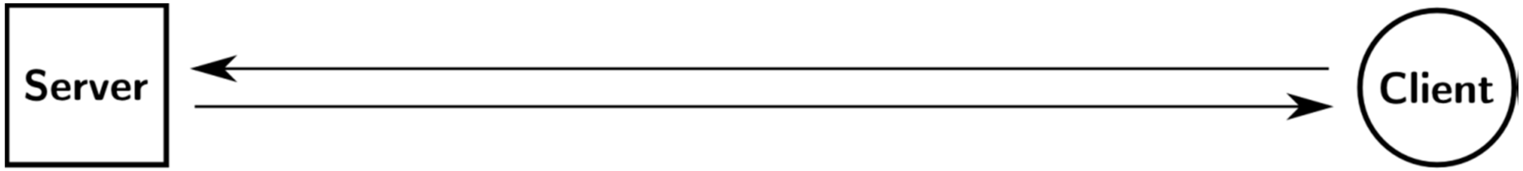
```
HTTP/1.1 200 OK
Date: Fri, 11 Nov 2011 11:11:11 GMT
Cache-NT: sha256=7ab53f24d8c96d1cc87452a6b113 …

<HTTP Body>
```

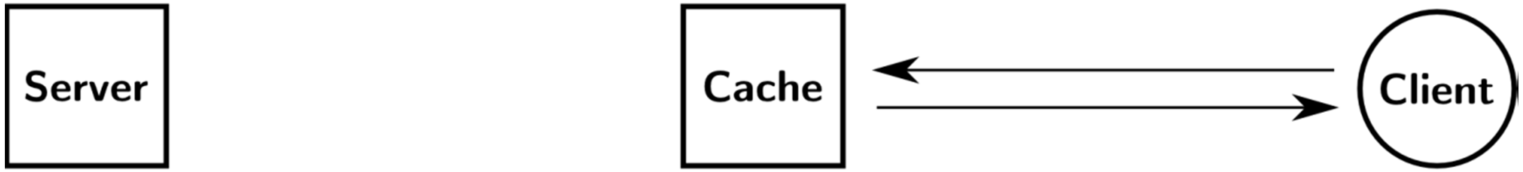
Improved HTTP Caching - Basic Concept

- Modified cache operation:

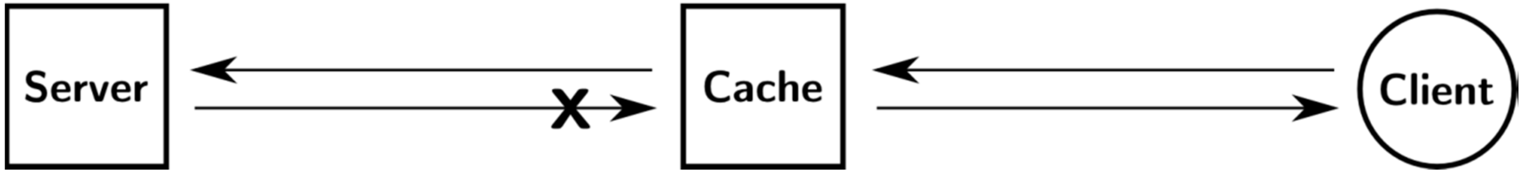
Without Caching:



Traditional Caching (example: cache hit):

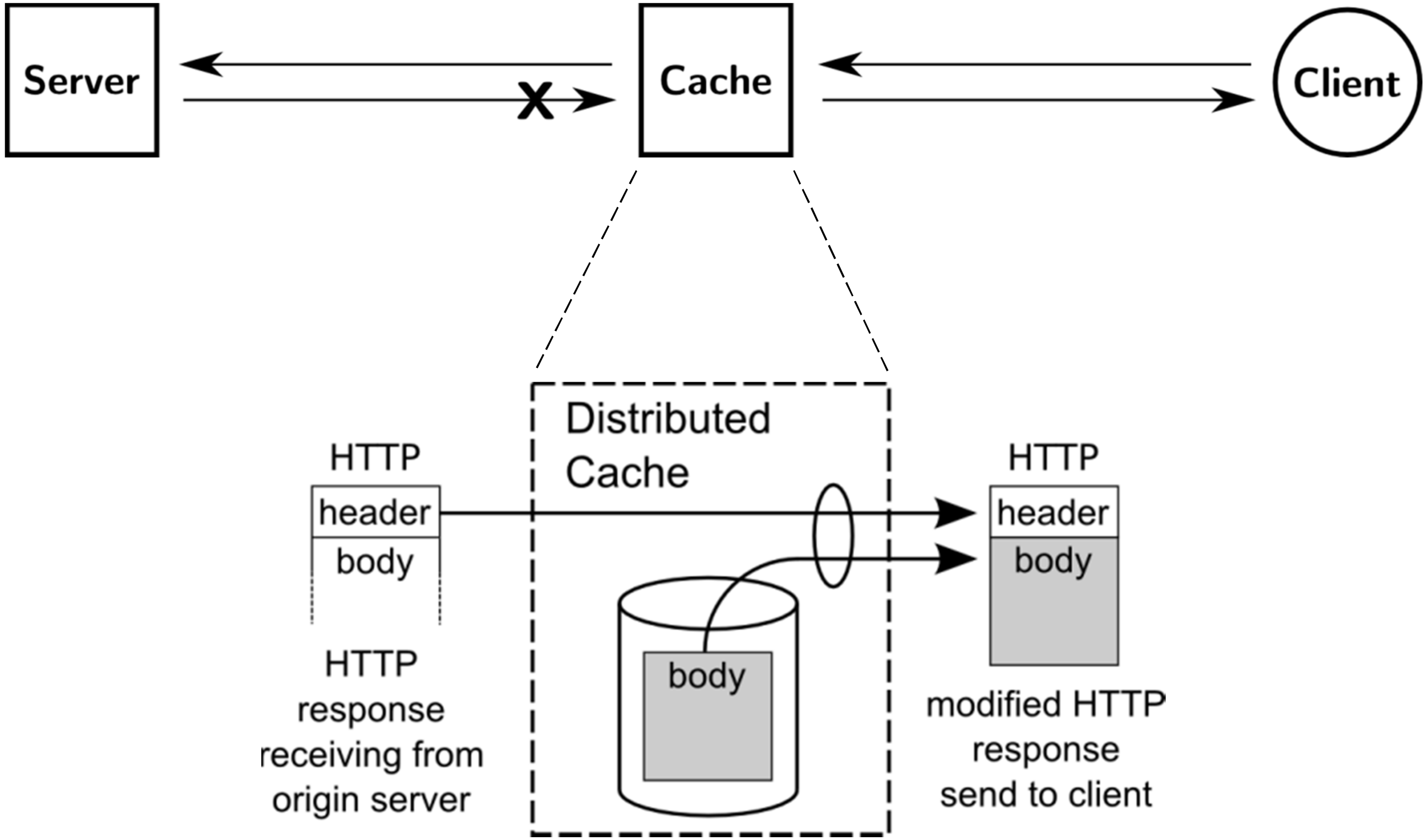


Modified Caching (example: cache hit):



Improved HTTP Caching - Basic Concept

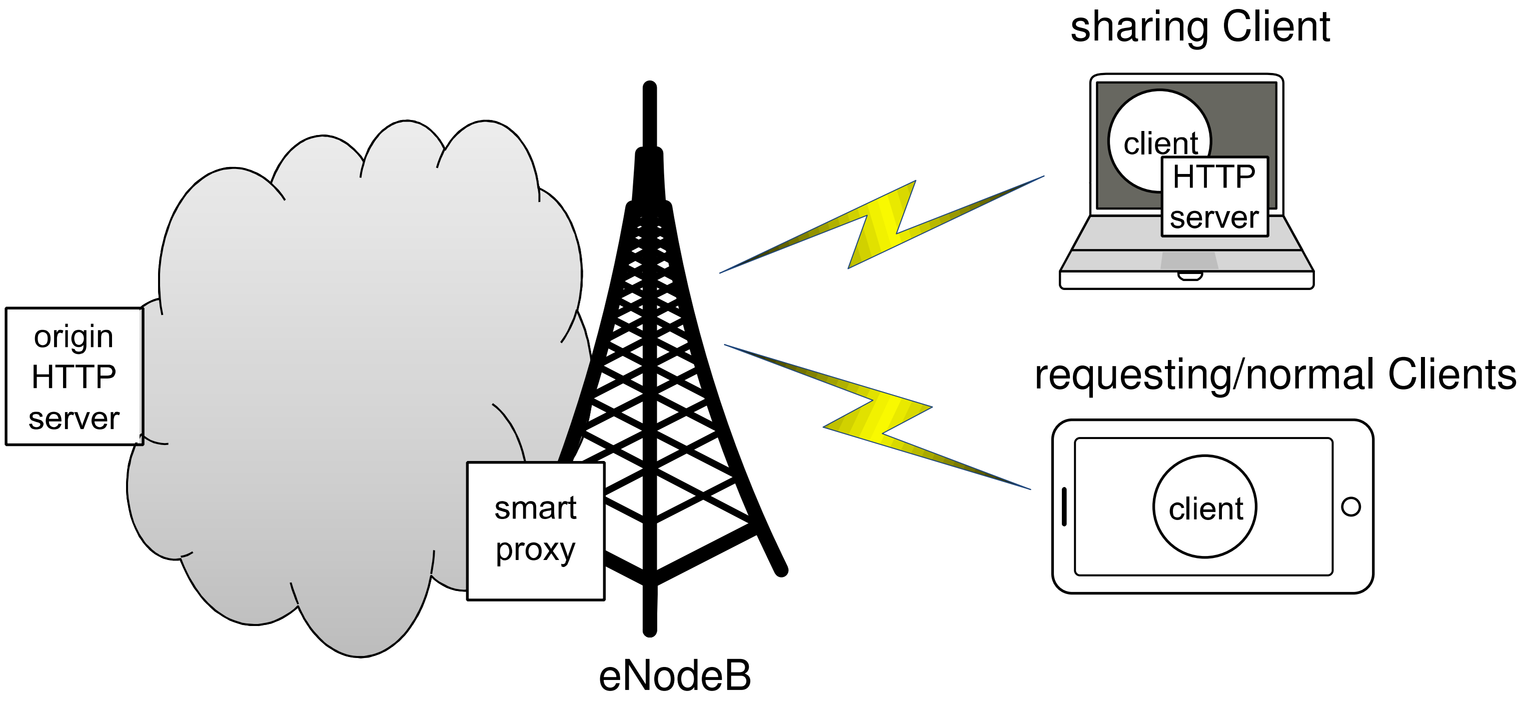
- Modified cache operation:



Outline

- Introduction
- Improved HTTP Caching Method
- **Distributed Caching Architecture**
- Distributed Cache Operation
- Summary

Distributed Caching Architecture - Overview

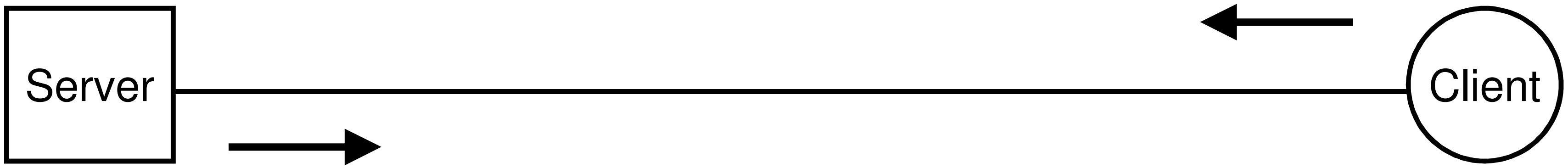


Distributed Caching Architecture - Origin HTTP server

Origin HTTP Server:

- acts like a normal HTTP server in the Internet
- one difference: adds the new HTTP header field

```
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>
```

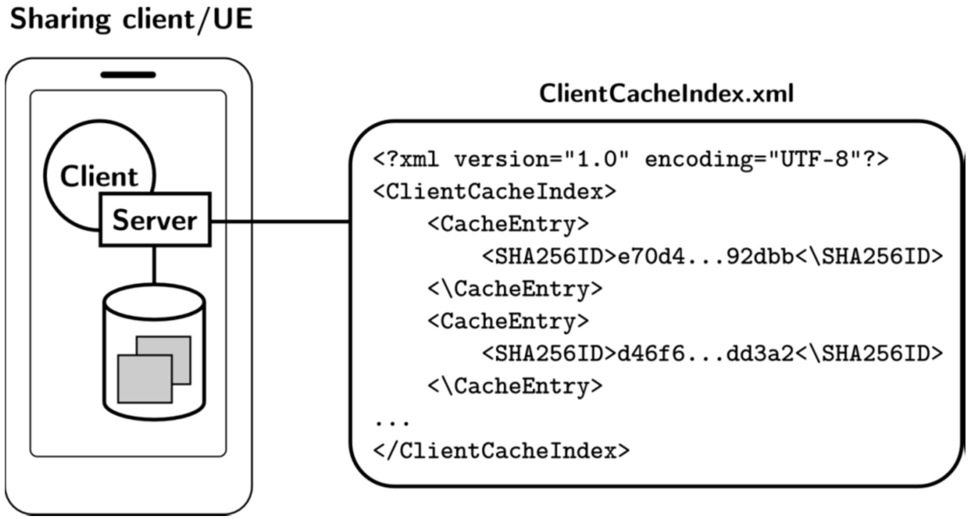
Distributed Caching Architecture - Clients

Requesting Clients:

- act like normal clients

Sharing Clients:

- sharing clients/UEs run a HTTP server
- they provide:
  - shared resources
  - index of shared resources



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

Distributed Caching Architecture - Smart HTTP Proxy

- Serves as a central element and lies in the data path between origin HTTP server and the clients/UEs
- Two basic functions:
  - builds an index of all shared resources (of all participating sharing clients within the cell coverage area of the eNodeB)
  - analyzes all incoming HTTP traffic and scans the HTTP header for the hash value (in the new header field)
- Can be easily implemented in eNodeBs as a software feature (no hardware upgrade needed)

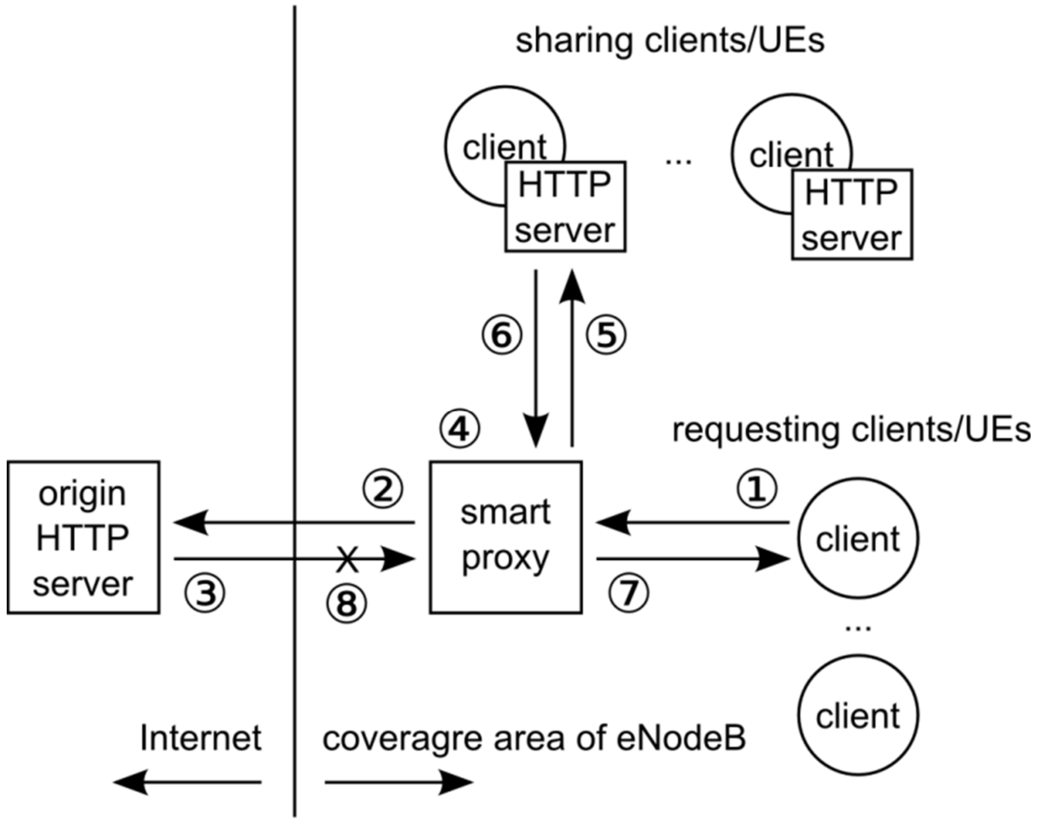
Distributed Caching Architecture - Smart HTTP Proxy

- Building the index of all shared resources (within the cell coverage area of the eNodeB)
    - not trivial since in a mobile environment the number of sharing clients/UEs is constantly changing
    - the index update is normally triggered periodically for active UEs
    - additional index updates are triggered by handover events and after attach or detach
    - special handling of idle mode UEs:
        - a timer is set when the UE goes idle
        - the UE and its shared resources are deleted from the index after the timer expires
        - the timer is reset with every TAU received from the UE

Outline

- Introduction
- Improved HTTP Caching Method
- Distributed Caching Architecture
- **Distributed Cache Operation**
- Summary

Distributed Cache Operation



1. Client HTTP request
2. Forwarding
3. Server HTTP response
4. Header analyzed, local copy available at one sharing client/UE
5. HTTP request (local copy)
6. HTTP response
7. HTTP response to client
8. Abort of HTTP transfer

Outline

- Introduction
- Improved HTTP Caching Method
- Distributed Caching Architecture
- Distributed Cache Operation
- **Summary**

Summary

- Novel approach for distributed cooperative caching
- The approach is based on three main concepts:
  - hash-based resource identification
  - distributed client-side caching
  - modified cache operation using smart proxies for cache control
- The approach is not limited to mobile networks, it could also be applied in fixed network scenarios

Questions?