CS305 Lab Tutorial
Lecture 6 CDN & DASH

HHQ. ZHANG
Dept. Computer Science and Engineering
Southern University of Science and Technology

SUSTech
Southern University of Science and Technology

Part A.

CDN & Web Cache

Content delivery network

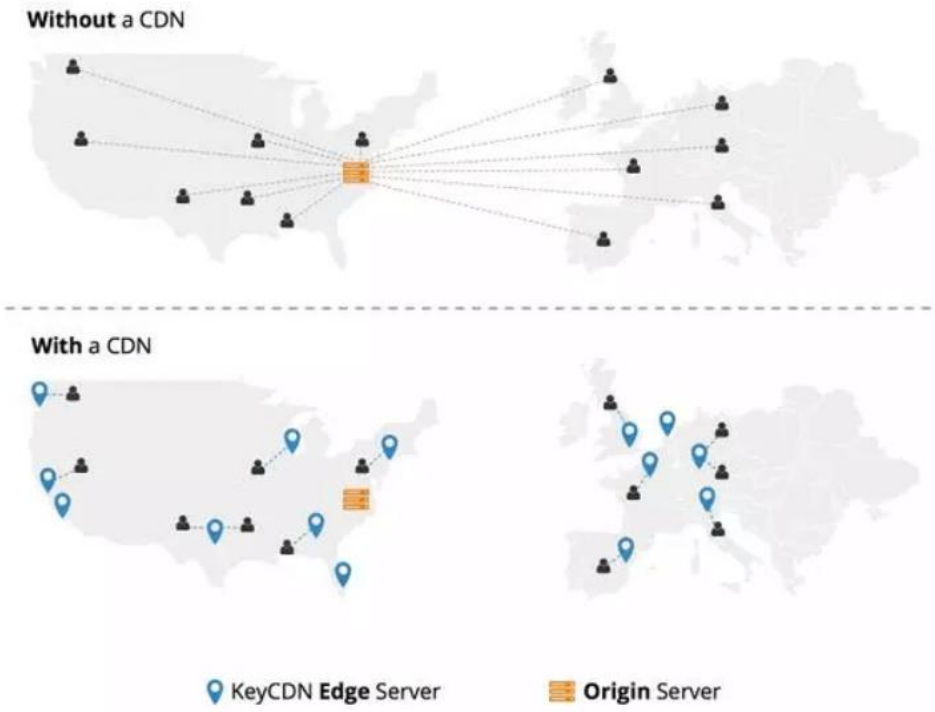
SUSTech
Southern University
of Science and Technology

CDN

- CDN is designed for cache content on a node closer to edge users to improve their experience.
- What are the scenarios of CDN?
  - Big flow website, such as: online video, games, pictures, audio, social, e-commerce, download stations, etc.
- CDN is suitable for a certain level of static resource access (html, js, css, jpg, gif, etc).

How does CDN work?

- A CDN server is actually a **reverse proxy cache server.**



SUSTech
Southern University
of Science and Technology

CDN Example

C:\WINDOWS\system32\cmd.exe

```
C:\Users\vivi>curl -I https://wx4.sinaimg.cn/mw690/6fa017c71y1fw55ou2gb2j21001qqb29.jpg
HTTP/1.1 200 OK
Server: edge-esnssl-1.12.1-12.1
Date: Sun, 14 Oct 2018 07:42:30 GMT
Content-Type: image/jpeg
Content-Length: 302723
Connection: keep-alive
x-oss-request-id: 5BBFF2EEB4DE0B1E77F7AEB5
ETag: "C3893605B7AD4E4A34D1D4E0E5C008CB"
Last-Modified: Fri, 12 Oct 2018 00:55:57 GMT
x-oss-object-type: Symlink
x-oss-storage-class: Standard
x-oss-hash-crc64ecma: 2801485700556859094
Via: cache48.12cm12-1[0,200-0,H], cache16.12cm12-1[0,0], cache8.cn1009[0,200-0,H], cache14.cn1009[1,0],
http/1.1 cmcc.guangzhou.ha2ts4.137 (ApacheTrafficServer/6.2.1 [cMsSf ])
Age: 196728
Ali-Swift-Global-Savetime: 1539306777
X-Swift-SaveTime: Fri, 12 Oct 2018 01:12:57 GMT
X-Swift-CacheTime: 8640000
Timing-Allow-Origin: *
EagleId: b7f0d5a215395029501395216e
X-Cache: MISS.137
X-Via-CDN: f=edge,s=cmcc.guangzhou.edssl.95.nb.sinaedge.com,c=183.232.197.103;f=edge,s=cmcc.guangzhou.ha
2ts4.103.nb.sinaedge.com,c=183.232.24.95;f=Edge,s=cmcc.guangzhou.ha2ts4.137,c=183.232.24.103;f=alicdn,s=
cache14.cn1009,c=183.232.24.137;
Access-Control-Allow-Origin:
X-Via-Edge: 153950295013267c5e8b7de18e8b734b573f6
```

A static resource which is cached on a CDN node of aliyun

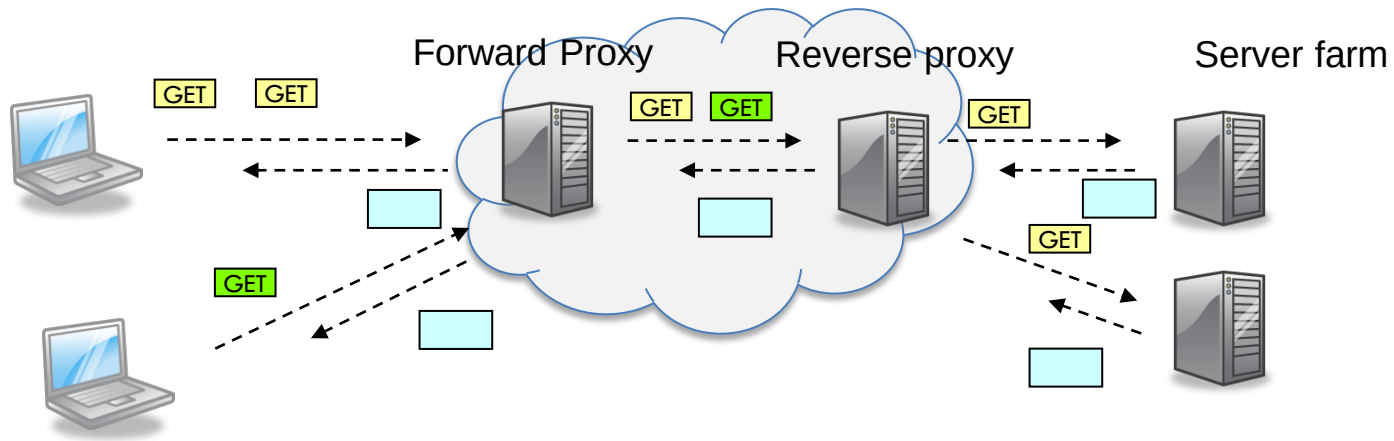
SUSTech
Southern University
of Science and Technology

Web proxies

- Web proxies are intermediaries between web clients and web servers that fulfill transactions on clients' behalf.
- A client sends a request to the proxy, which forwards the request to the server. When the proxy receives a response from the server, it forwards the response back to the client.
  - Proxies act like servers to web clients
  - Proxies act like clients to web servers

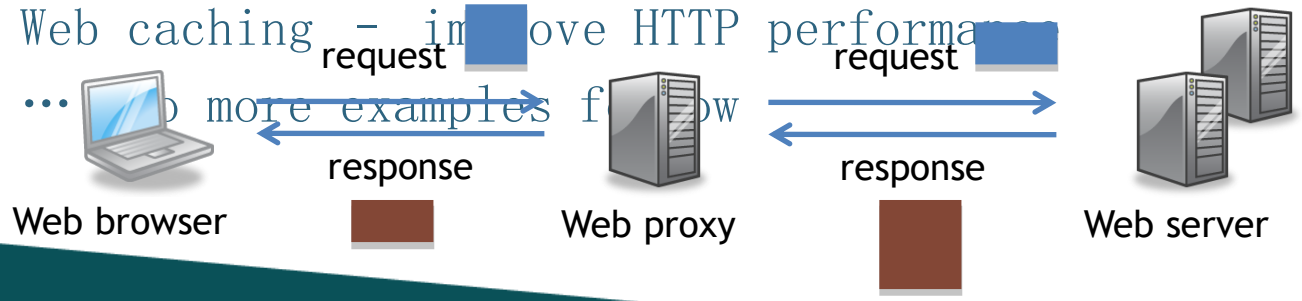
Web proxies

- There can be multiple proxy servers between a browser and the origin web server (which produces HTTP responses)
- Proxy servers are transparent to end-users.



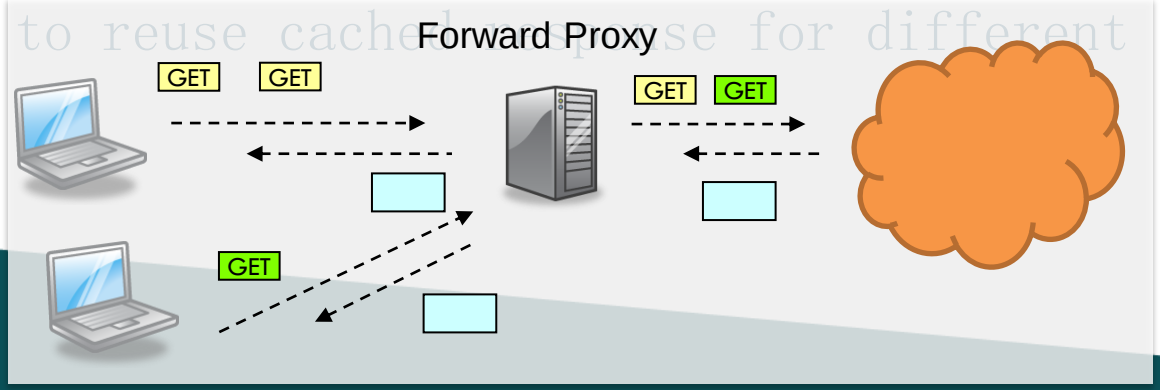
Functions of Proxies

- A web proxy may scan and change requests and responses in transit. It may also route a request to different servers or handle the request itself.
- Some possible functions:
  - Content filtering - block access to inappropriate content
  - Security firewall - block malicious software like virus
  - Web caching - improve HTTP performance
  - … more examples follow



Forward proxy

- A forward proxy acts on behalf of a client (or other forward proxy) to access web servers in the Internet
- Forward proxy server can concentrate HTTP traffic. Requests from different sessions may be transmitted in one TCP connection
  - Reduce outgoing bandwidth usage and concurrent TCP connections
- Possible to reuse cached response for different users



SUSTech
Southern University of Science and Technology

More about forward proxy

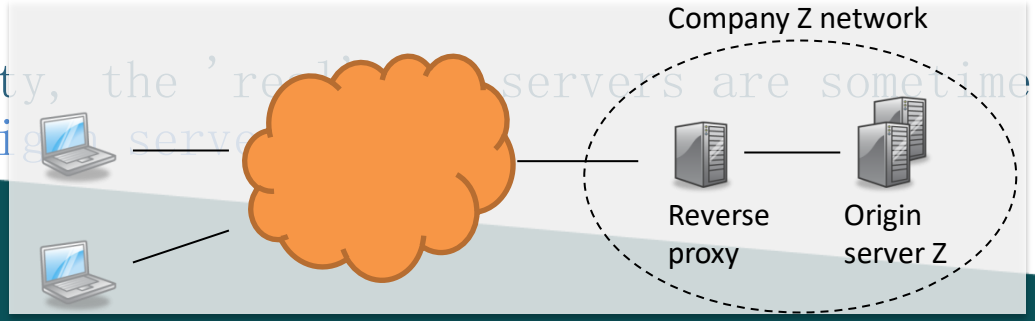
- A forward proxy often hides the IP of the clients
  - TCP connection between the proxy and the server, but not between the client and server
- Some proxies add a request header X-Forwarded-For to reveal client's IP address
  - Experiment*: http://whatismyipaddress.com/

*Notice: If you are behind a NAT, the result will not be correspond to your IP address, which doesn't mean you are behind a HTTP proxy.

SUSTech
Southern University of Science and Technology

Reverse proxy

- Reverse proxy is a proxy server that retrieves resources on behalf of a client from one or more servers
  - Usually, a reverse proxy only connects to web servers of a web site
  - Popular software: Nginx, lighttpd
- For client, the reverse proxy works as the web server of the company
  - Client cannot connect to the 'real' web servers behind. The reverse proxy uses IP address of a web site.
  - For clarity, the 'real' web servers are sometimes called origin servers



SUSTech
Southern University of Science and Technology

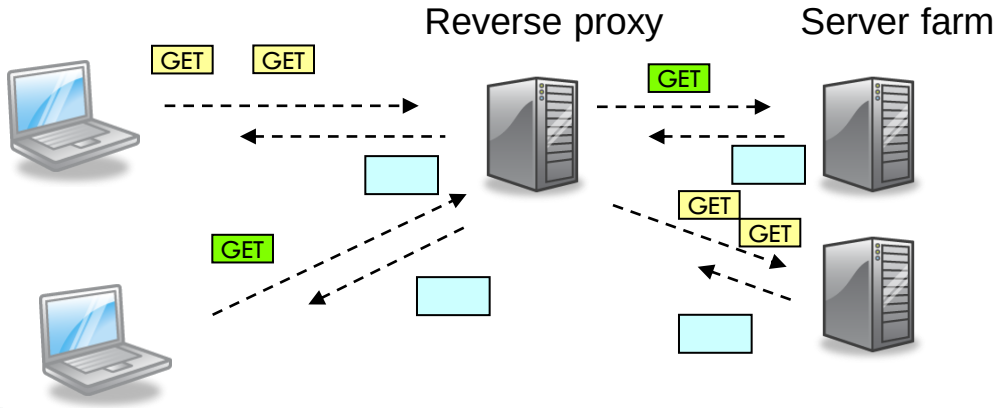
Functions of reverse proxy

- Reduce workload of origin servers
  - Caching
  - Load balancing
  - Serving static resources (e.g. images)
  - HTTP Compression and encryption

- Protect against common web-based attacks

Reverse Proxy for load balancing

- A reverse proxy usually sits before a server farm
  - Each server in the farm duplicates databases, programs and static resources (e.g. images)
  - The reverse proxy dispatches request from the same user to one of the servers in the server farm.
  - Each server maintains client sessions.



SUSTech
Southern University of Science and Technology

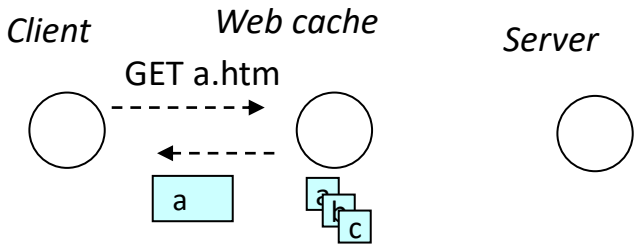
Caching is everywhere (Examples)

Tommy is accessing a web site through Chrome on his MacBook.

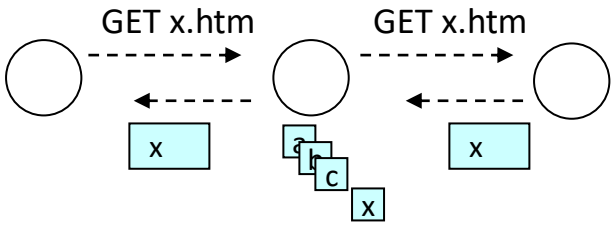
- Web server: It will cache rendering result (for dynamic pages).
- CDN server: It will cache static content.
- Web browser: It will cache web resources*.
- Operating System: It will likely cache file of browser cache in memory.
- Hybrid Drive: It will cache recent access blocks of HDD in SSD.

Web caching, cache hit / cache miss

- Some resources are retrieved frequently when users browse the web
  - e.g. images, popular web pages, JavaScript libraries
- Web cache saves such resources (in memory or file system) and use them to satisfy future requests from clients.
  - Cache hit: the requested resource is available in the cache
    - return the cached copy as a response



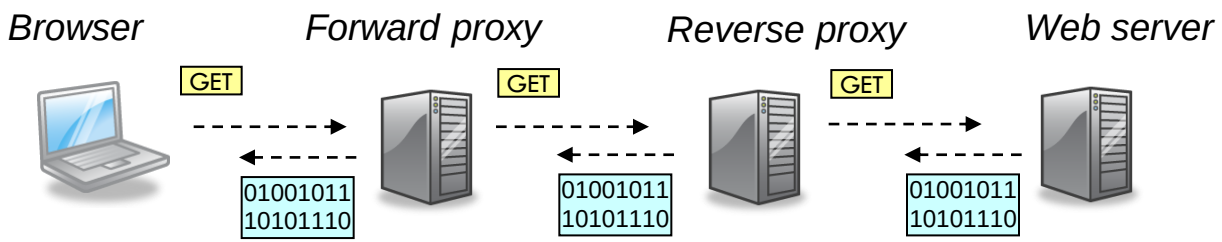
  - Cache miss: the requested resource is not available
    - forward the request to the web server
    - save the response in cache
    - return the response to the client



SUSTech
Southern University of Science and Technology

Varieties of web caching

- Web request / response travel through several machines from a client to a server. Web caching is done in several places:
  - Web browser(e.g: firefox about:cache)
  - Web proxies:
    - forward proxy (Cache server)
    - reverse proxy
  - Web server



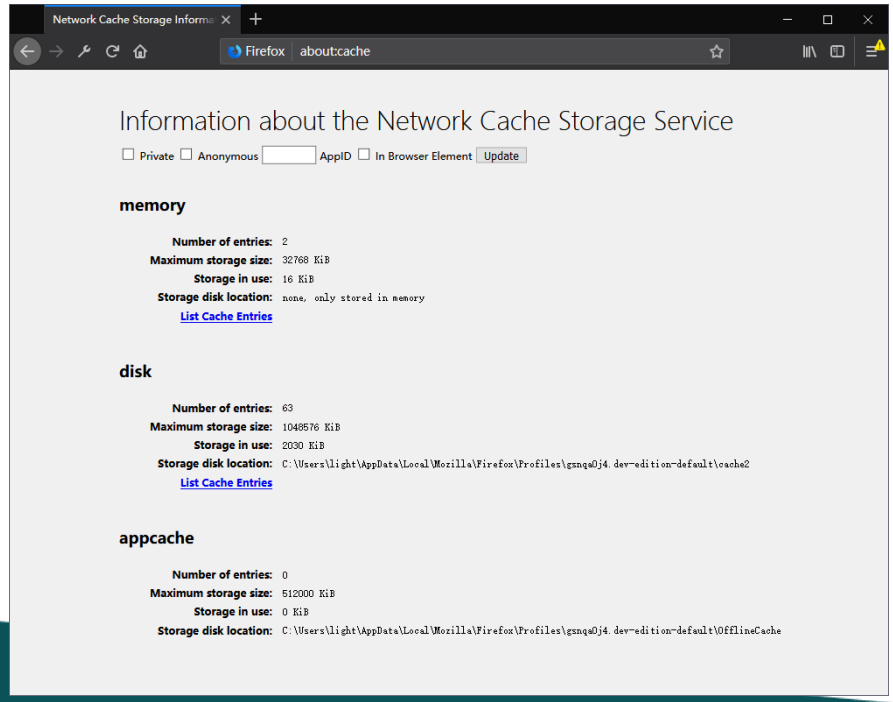
Web browser cache

- Built-in caching of browsers. It saves cached copies in memory and disk on the client machine.
- Can cache private resources (response with Cache-Control: private)

Experiment:

*Both chrome://cache and chrome://view-http-cache have been removed sience chrome 66.

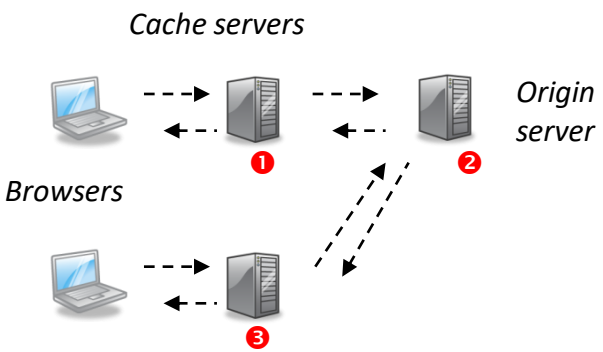
*Firefox: about: cache



SUSTech
Southern University of Science and Technology

Web cache consistency

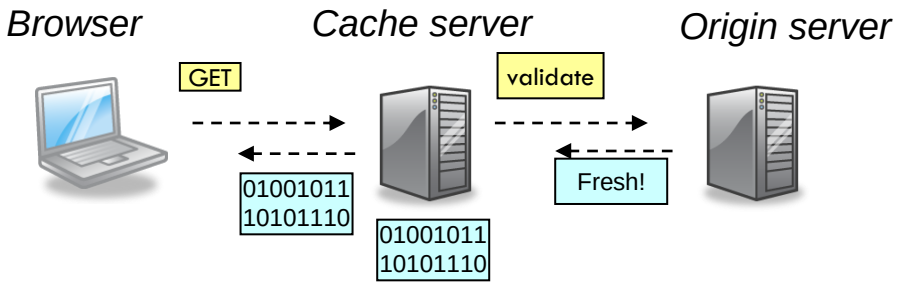
- Resources in an origin server may be modified after a cache server saves a copy
- If the cache server returns such an outdated copy, clients will have inconsistent view of the web site



1. Cache server A saves a copy of a resource at 8:00
2. The resource is modified at origin server at 9:00
3. Cache server B retrieves the resources and saves a copy at 10:00

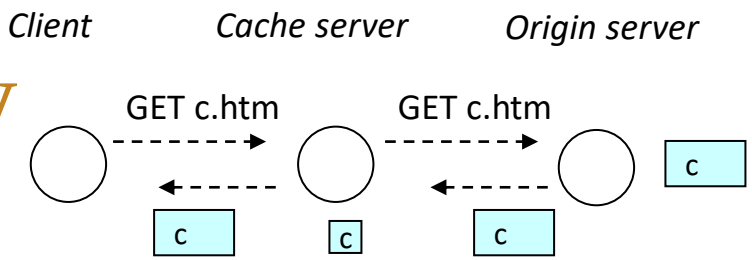
1)Ensuring consistency by validation

- Validation: a cache server inquires the origin server whether the cached copy is 'the same' as the resource in the origin server
- The cache server may validate a cached copy before satisfying client's request
  - But validation for every request would be too expensive.

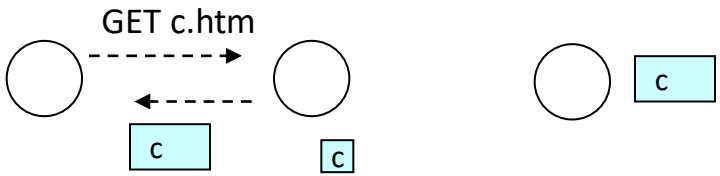


2)Fresh cached copy

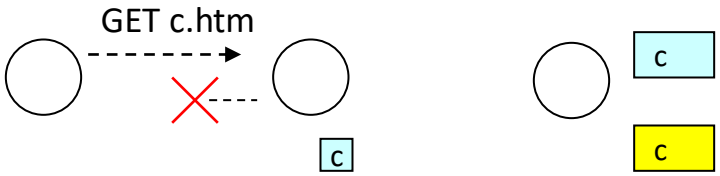
- The origin server may specify an expiration time of its response. Before a cached copy expires, it remains fresh.
  - The resource is not likely to change before it expires
- The cache server considers it safe to satisfy a client's request with a fresh cached copy.
- If a cached copy has expired, it becomes stale.
- It is likely that the resource in the origin server has changed.
  - The cache server cannot satisfy a client's request with a stale cached copy.
  - But the stale cached copy may still be the same as the resource in the origin server



Cached copy expires at 14:00



At 13:30, the cached copy is still fresh. So the cache server can return the copy to the client.

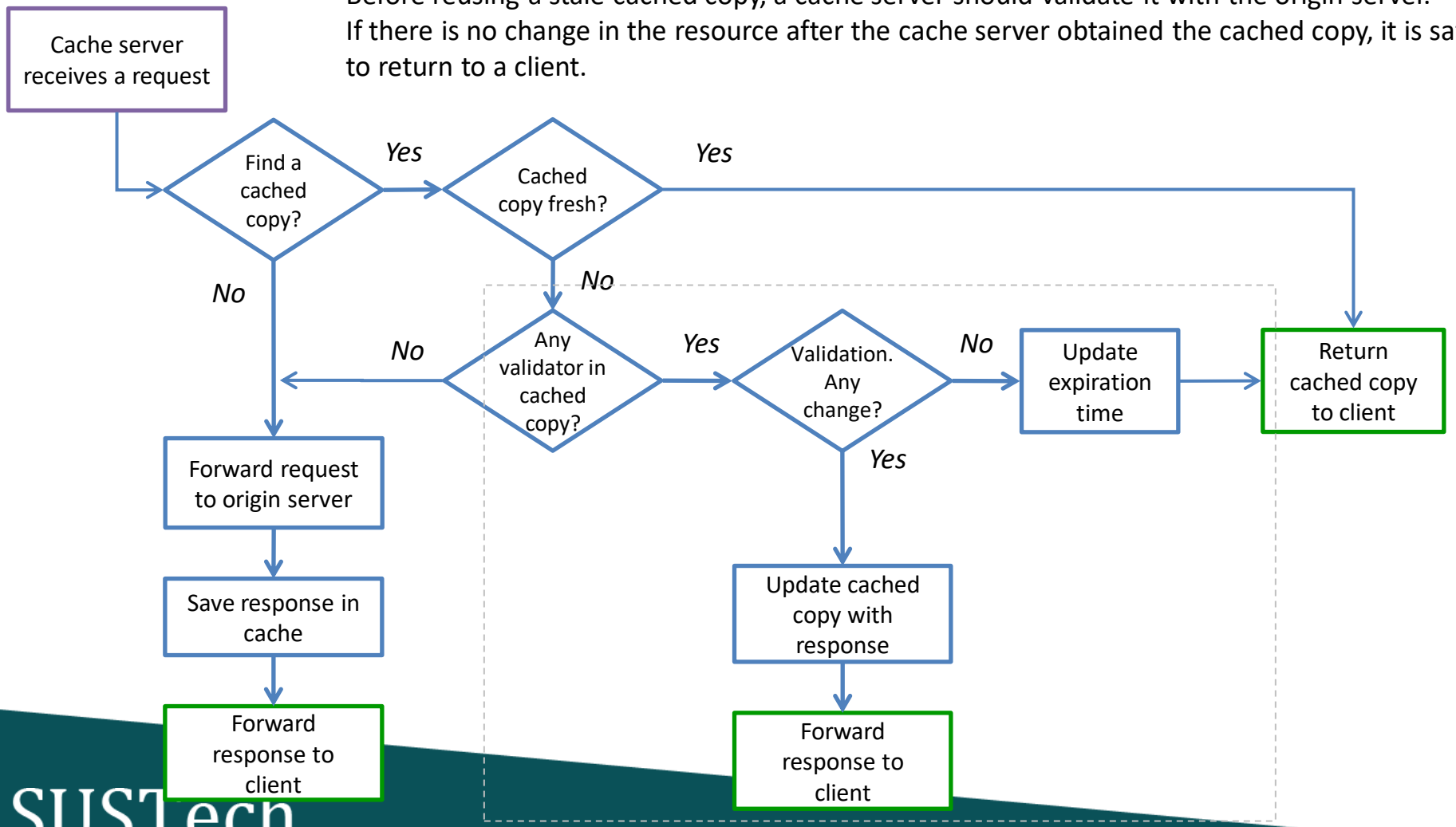


At 15:00, the cached copy has expired. The cache server cannot use the copy immediately.

SUSTech
Southern University of Science and Technology

Operation of Cache Server

Before reusing a stale cached copy, a cache server should validate it with the origin server. If there is no change in the resource after the cache server obtained the cached copy, it is safe to return to a client.



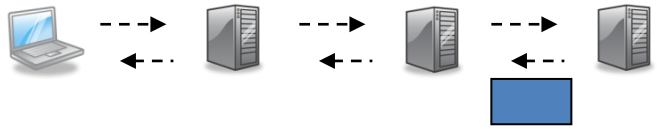
SUSTech
Southern University of Science and Technology

Cache support in HTTP

- "The goal of caching in HTTP/1.1 is to eliminate the need to send requests in many cases, and to eliminate the need to send full responses in many other cases." RFC 2616
- Cache-related headers in HTTP
  - HTTP/1.0: Date, Expires, Last-Modified, If-Modified-Since
  - HTTP/1.1: Cache-Control, Etag, If-None-Matched, Vary

Controlling expiration / freshness

- An origin server should indicate whether its resources can be cached. (cacheability)
- If a resource can be cached, the origin server should indicate the expiration time of a response.
  - In case none is provided, the cache server uses some heuristics to estimate.
  - Nonetheless, a cache server may not strictly observe the expiration date set by origin server



Cacheability of resources

- A resource is cacheable if a cache server can save a copy and later use it to satisfy client's request
  - Generally, GET responses may be cached, but not for POST.
- An origin server defines the cacheability in the response header Cache-Control

	Private cache (e.g. browser cache)	Shared cache (e.g. cache server)
Cache-Control: no-store	✗	✗
Cache-Control: private	✓	✗
Cache-Control: public	✓	✓

SUSTech
Southern University of Science and Technology

Expiration related headers

- An origin server uses these headers to set the freshness time of a resource.
  - HTTP/1.1 headers are not understood by HTTP/1.0 cache.

Header (Response)	Meaning
Date:	Time that the server generates the response
Expires:	Time that this page will expire
Cache-Control: max-age=n	The response will remain fresh for n seconds
Cache-Control: must-revalidate	The cache server must observe the expiration time set in other headers.
Cache-Control: no-cache	The cached response cannot be used without validation

HTTP/1.1

SUSTech
Southern University of Science and Technology

Cache-Control: max-age

- A server can also indicate the maximum time (in sec) that a response remains fresh using max-age.
- Similar to Expires:, but max-age is relative to response time.

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2009 01:46:57 GMT
Cache-Control: max-age=3600
Last-Modified: Tue, 17 Feb 2009 05:14:05 GMT
…
```

This response expires 1 hour later, i.e. at 18 Feb 2009 02:46:57 GMT.

Cache-Control: must-revalidate

- In some cases, a cache server may choose to return a stale cached copy to client without validation
  - A cache server may be configured to compute expiration time using its own heuristics. It may not observe the expiration time set by origin servers.
  - A cache server may fail to connect to the origin server for validation. It may choose to return the stale cached copy to clients.
- An origin server uses Cache-Control: must-revalidate to force a a cache server to validate before using a stale cached copy. If it cannot successfully validate the copy, the cache server should return the 504 (Gateway Timeout) error.

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2009 01:35:00 GMT
Cache-Control: max-age=1800, must-revalidate
…
```

SUSTech
Southern University of Science and Technology

Validation before each reuse

- An origin server can force a cache server to validate a cached copy for each request using Cache-Control: no-cache. This disallows using the cached copy without validation.
  - No-cache doesn't prohibit the cache server to save the response.

```
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2009 01:35:00 GMT
Cache-Control: no-cache
Last-Modified: Tue, 17 Feb 2009 05:14:05 GMT
…
```

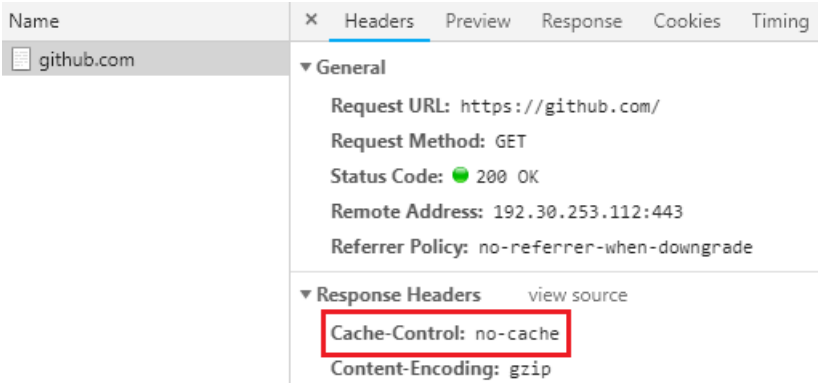
SUSTech
Southern University of Science and Technology

Example

Cache-Control header	When to validate?
Cache-Control: no-cache	Must validate before reusing
Cache-Control: max-age=60	No need to validate if reuse it within 1min of retrieval. SHOULD validate if expired.
Cache-Control: max-age=60, must-revalidate	No need to validate if reuse it within 1min of retrieval. MUST validate if expired.
Cache-Control: max-age=0, must-revalidate	*What does this mean?*

Exercise

- Check the cacheability of web resources with https://github.com/
- What kinds of resources should have a small / large maxage?



SUSTech
Southern University of Science and Technology

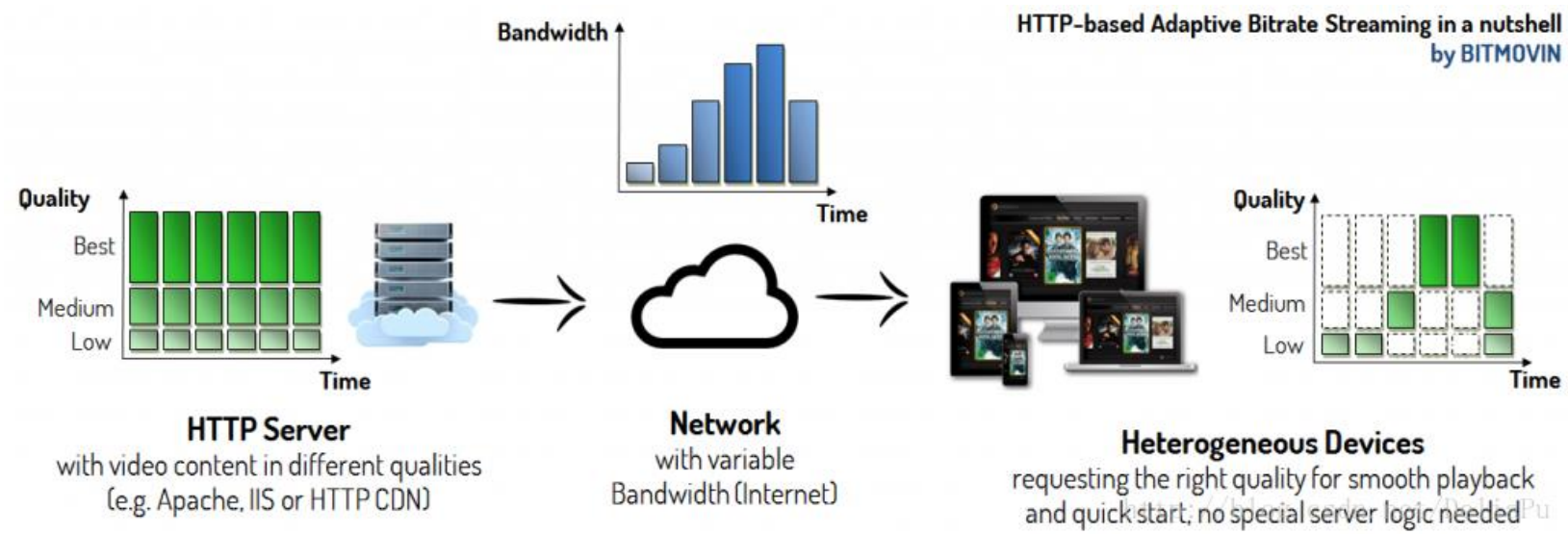
Part. B

DASH

Dynamic Adaptive Streaming over HTTP

SUSTech
Southern University
of Science and Technology

Part A.2 DASH

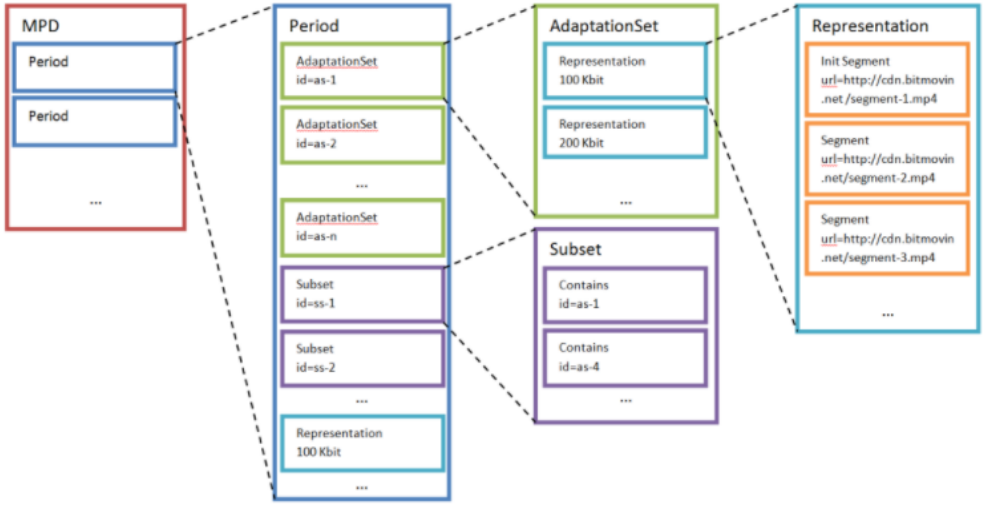


mpd & m4s

Index of /ftp/datasets/DASHDataset2014/ElephantsDream/15sec

Name	Last modified	Size	Description
Parent Directory		-	
ElephantsDream_15s_onDemand_2014_05_09.mpd	16-Oct-2014 16:04	8.7K	
ElephantsDream_15s_simple_2014_05_09.mpd	16-Oct-2014 16:04	4.3K	
ed_45826bps/	15-Sep-2014 13:34	-	
ed_89324bps/	15-Sep-2014 13:34	-	
ed_127632bps/	15-Sep-2014 13:33	-	
ed_177678bps/	15-Sep-2014 13:33	-	
ed_217728bps/	15-Sep-2014 13:33	-	
ed_253741bps/	15-Sep-2014 13:33	-	
ed_313190bps/	15-Sep-2014 13:33	-	
ed_361788bps/	15-Sep-2014 13:34	-	
ed_503134bps/	15-Oct-2014 14:25	-	
ed_570572bps/	15-Oct-2014 14:25	-	
ed_790369bps/	15-Sep-2014 13:34	-	
ed_1006812bps/	15-Sep-2014 13:33	-	
ed_1186142bps/	15-Sep-2014 13:33	-	
ed_1431961bps/	15-Sep-2014 13:33	-	
ed_2065452bps/	15-Sep-2014 13:33	-	
ed_2362158bps/	15-Sep-2014 13:33	-	
ed_2858043bps/	15-Sep-2014 13:33	-	
ed_3272272bps/	15-Sep-2014 13:34	-	
ed_3577865bps/	15-Sep-2014 13:34	-	
ed_3950139bps/	15-Sep-2014 13:34	-	

ms4 files in different rate



Index of /ftp/datasets/DASHDataset2014/ElephantsDream/15sec/ed_45826bps

Name	Last modified	Size	Description
Parent Directory		-	
ElephantsDream_15s1.m4s	09-Sep-2014 12:55	85K	
ElephantsDream_15s2.m4s	09-Sep-2014 12:55	90K	
ElephantsDream_15s3.m4s	09-Sep-2014 12:55	87K	
ElephantsDream_15s4.m4s	09-Sep-2014 12:55	87K	
ElephantsDream_15s5.m4s	09-Sep-2014 12:55	86K	
ElephantsDream_15s6.m4s	09-Sep-2014 12:55	87K	
ElephantsDream_15s7.m4s	09-Sep-2014 12:55	85K	
ElephantsDream_15s8.m4s	09-Sep-2014 12:55	84K	

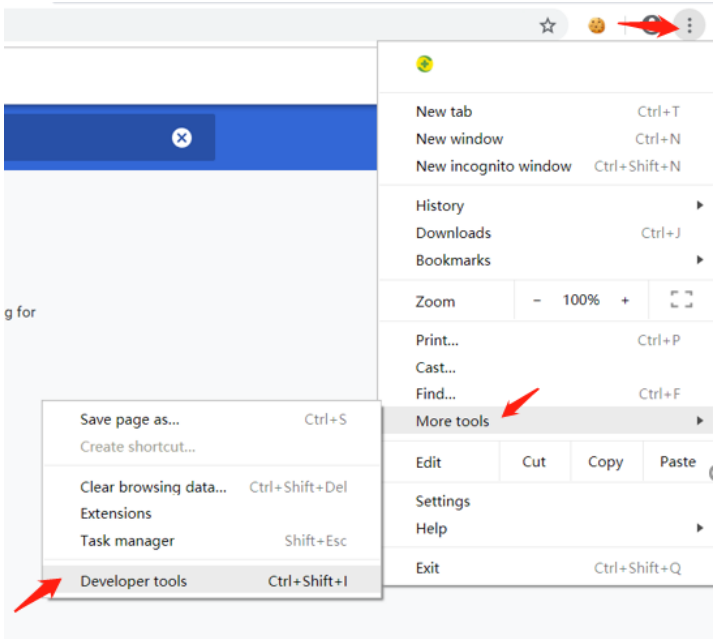
http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset2014/

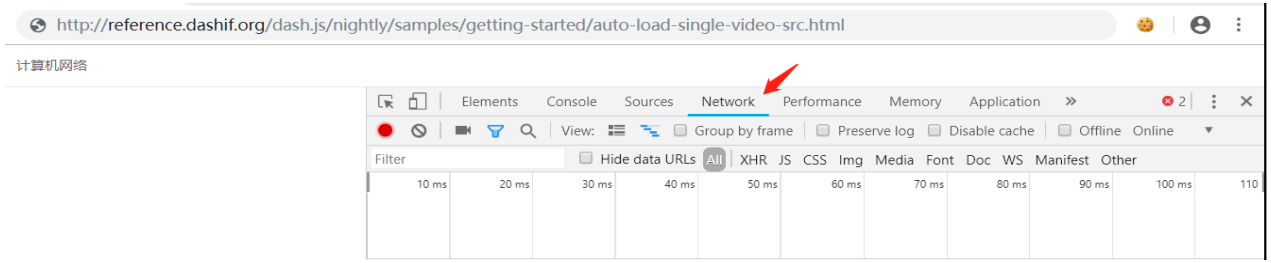
SUSTech
Southern University of Science and Technology

practice

1. open “chrome”
2. open the “developer tools” of chrome
3. visit the url: https://allen8101070.github.io/ITMAN_DASHjs/index.html

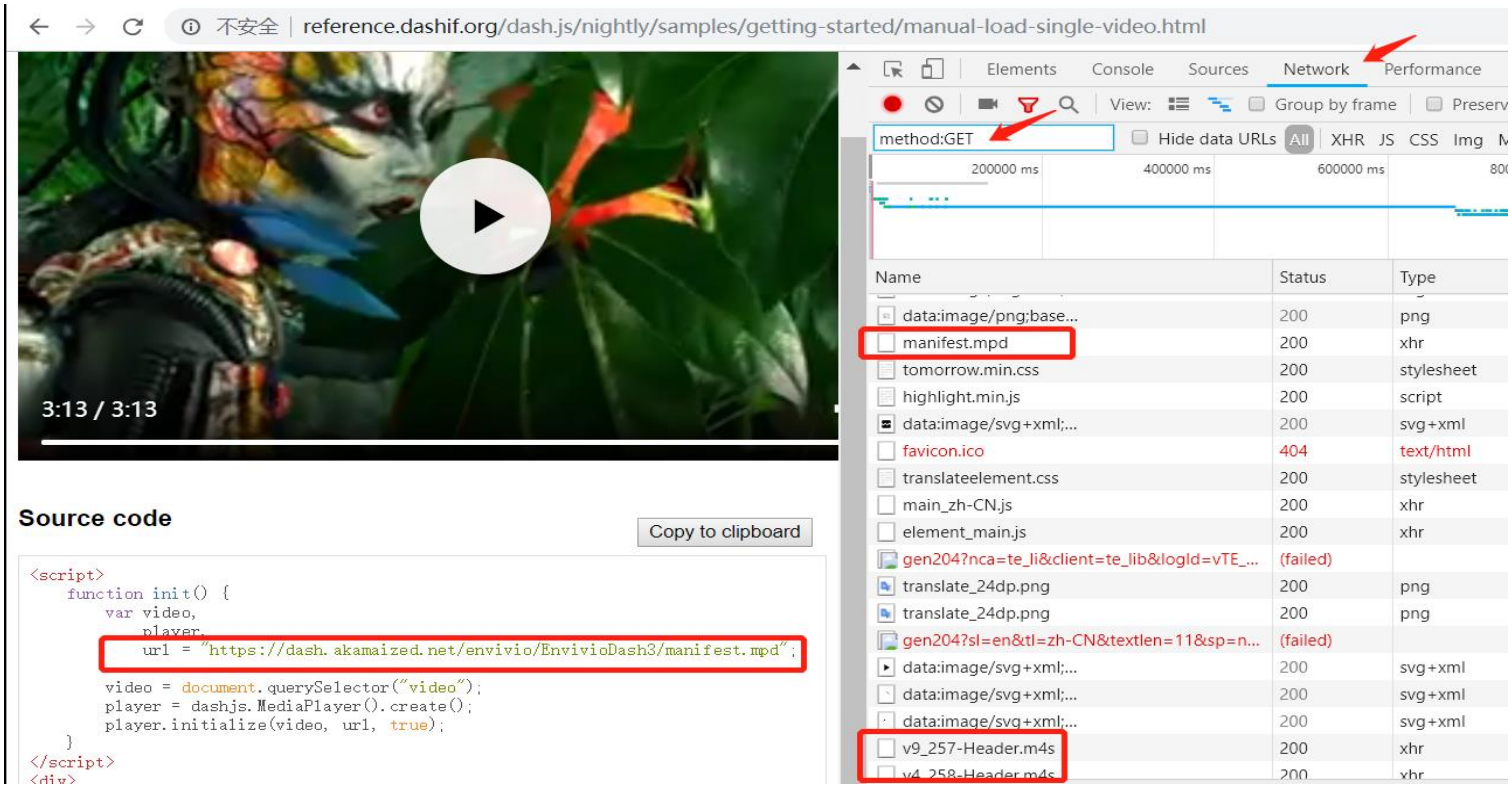
4. Observe what happened on the ‘Network’ view of “developer tools”





SUSTech
Southern University of Science and Technology

Testing result



SUSTech
Southern University
of Science and Technology

lab 6

- Please finish the lab according to this file
  - submit the **report** of lab 6.
  - submit your source code in zip file. (**6.3.zip**)
    - comments is MUST

SUSTech
Southern University of Science and Technology

lab 6.1 finding a CDN user

- Using curl to Get a resource from web which using CDN to upgrade the accessing speed and balance the traffic load
  - How can you tell that this web is using CDN
  - Using nslookup/dig to find the ip address the this web sit by your computer
  - Ask a friend who is in another province ,ask him/her to practice the same thing(using nslookup/dig to find the ip of the same web site which using CDN, find the ip address of this web sit)
  - Record the result in your report

lab 6.2 loading a Dash resource

- Using dash.js to load a dash resource
- Open “Network” view in ’developer tools’ of browse(such as chrome) to observe
  - Is there any ‘mpd’ files, What’s its name, what is the description of ‘mpd’ in mime
  - Is there any ‘m4s’ files, what’s its related rate, will the files’ ‘rate’ change along with the changing of network condition(especially the bandwidth)
- Reference:
  - A html embedded a dash.js which maybe helpful for loading a ‘mpd’ file
    - https://allen8101070.github.io/ITMAN_DASHjs/index.html
  - A dataset of dash resources
    - http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset2014/

SUSTech
Southern University of Science and Technology

lab 6.3

- Using multi thread and TCP socket to rewrite the http server which is asked in lab assignment 3.3:
  - Based on Assignment 3.3, implement following features:
  - Range Header support
    - With this feature implemented, user can pause and resume download file from the server.
  - Session Cookie support:
    - Remember last folder user visited, response with 302 Found if user access root directory.
      Example:
      Request: GET http://localhost:8080
      Response: 302 Found, Location: http://localhost:8080/lastdir

Reference: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Location