

# Web Server Design

## Lecture 6 – Transfer and Content Encodings

Old Dominion University

Department of Computer Science

CS 431/531 Fall 2019

**Sawood Alam** <salam@cs.odu.edu>

2019-10-03

Original slides by Michael L. Nelson

# HTTP equivalent of “they’re / their / there”, “you’re / your”, etc.



Extending the analogy,  
“ur” is acceptable only when  
you know the rules, but  
breaking them provides  
some measurable comfort  
or convenience.

<http://theoatmeal.com/comics/misspelling>

# Common Encodings

- identity
  - no encoding at all; defined in 2616, removed in 7230
- gzip
  - extension: .gz (sometimes seen as x-gzip, deprecated)
- compress
  - extension: .Z (sometimes seen as x-compress, deprecated)
- deflate
  - extension: .zip
- chunked
  - breaks the body into a series of server-chosen “chunks”
  - optimization for dynamically produced content

# Identity Encoding

- The default, “no transformation” encoding
  - even though it was removed in 7230 and never really existed in the wild, it is a useful rhetorical construct
  - “applying the identity encoding to a resource is an \_\_\_\_\_ ??? \_\_\_\_\_ operation”

Hint: Applying identity encoding repeatedly makes no difference!

# Content-Codings

“Content coding values indicate an encoding transformation that has been or can be applied to a representation. Content codings are primarily used to allow a representation to be compressed or otherwise usefully transformed without losing the identity of its underlying media type and without loss of information. Frequently, the representation is stored in coded form, transmitted directly, and only decoded by the final recipient.”

– 3.1.2.1, RFC 7231

# Content Encoding vs. Transfer Encoding

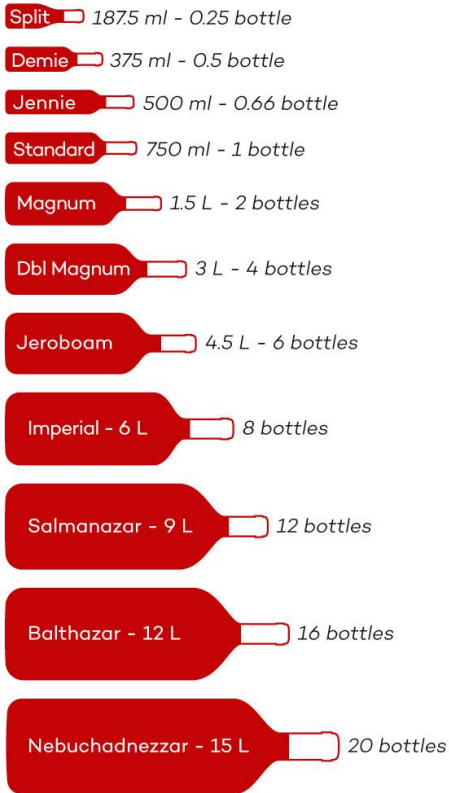
3.1.2.2, RFC 7231

Unlike Transfer-Encoding (Section 3.3.1 of [RFC7230]), the codings listed in Content-Encoding are a characteristic of the representation; the representation is defined in terms of the coded form, and all other metadata about the representation is about the coded form unless otherwise noted in the metadata definition. Typically, the representation is only decoded just prior to rendering or analogous usage.

If the media type includes an inherent encoding, such as a data format that is always compressed, then that encoding would not be restated in Content-Encoding even if it happens to be the same algorithm as one of the content codings.

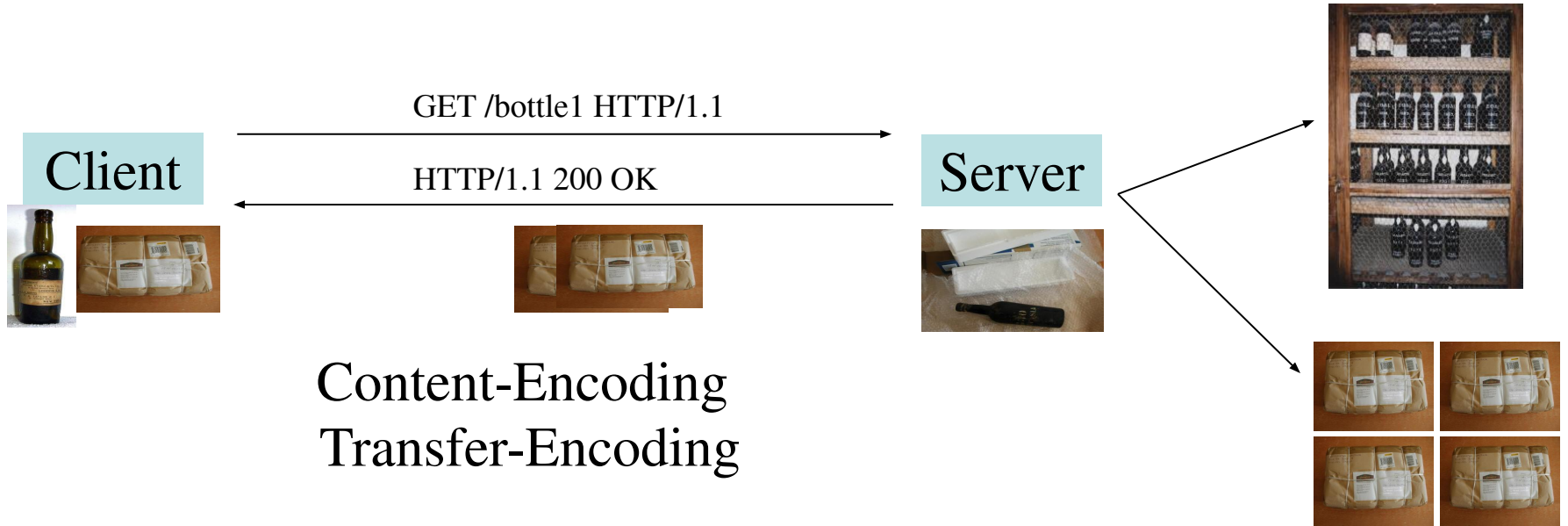
e.g., GIF uses LZW compression (“compress”),  
but this is not reflected in a Content-Encoding header

## WINE BOTTLE SIZES



The wine (liquid) is the  
Content-type;  
the bottle size is the  
Content-Encoding

# Content Encoding vs. Transfer Encoding





# Packaging/Shipping Doesn't Change the Representation



<https://mashable.com/2013/08/22/71-lbs-fedex/>

<https://about.van.fedex.com/blog/boeing-777-ecodemonstrator/>

# Encoding Can Mean Many Things

- Content type (or MIME type)
- Character encoding
  - “charset” attribute for textual MIME types
  - “utf-8” is the most popular charset, but there are many others
- Content encoding
- Transfer encoding

# Content-Encoding Example (Correct)

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/pubs/bollenj_adaptive.ps.gz HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 200 OK
Date: Mon, 20 Feb 2006 04:30:25 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2
PHP/4.3.4
Last-Modified: Thu, 25 Jul 2002 16:58:58 GMT
ETag: "1c16-139ea-3d402e52"
Accept-Ranges: bytes
Content-Length: 80362
Connection: close
Content-Type: application/postscript
Content-Encoding: x-gzip
```

```
Connection closed by foreign host.
```

# Content-Encoding Example (Incorrect)

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/pubs/bollenj_adaptive.ps.gz
HTTP/1.1
Host: www.cs.odu.edu
Connection: close
```

```
HTTP/1.1 200 OK
Date: Mon, 26 Feb 2007 02:06:25 GMT
Server: Apache/2.2.0
Last-Modified: Thu, 25 Jul 2002 16:58:58 GMT
ETag: "1c16-139ea-92cab880"
Accept-Ranges: bytes
Content-Length: 80362
Connection: close
```

**Content-Type:** application/x-gzip

**Wrong, Wrong, Wrong!!!!!!**

```
Connection closed by foreign host.
```

# Why is it incorrect?

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/adl98.ppt.gz HTTP/1.1
Host: www.cs.odu.edu
Connection: close
```

```
HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 02:21:29 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Mon, 25 Mar 2002 17:15:44 GMT
ETag: "33e94-39d06961d5000"
Accept-Ranges: bytes
Content-Length: 212628
Connection: close
Content-Type: application/x-gzip
```

the encodings are the same,  
the types are different

# Compress

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/ntrs.tar.Z HTTP/1.1
Host: www.cs.odu.edu
Connection: close
```

```
HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 02:31:45 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Thu, 12 Jun 2003 18:45:46 GMT
ETag: "7fffffff-3bfeb99a3ca80"
Accept-Ranges: bytes
Content-Length: 2147483647
Connection: close
Content-Type: application/x-compress
```

# Zip

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/michael-hany.zip HTTP/1.1
Host: www.cs.odu.edu
Connection: close
```

```
HTTP/1.1 200 OK
Date: Mon, 20 Feb 2012 03:16:35 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17
OpenSSL/0.9.8q
Last-Modified: Fri, 17 Feb 2012 20:43:13 GMT
ETag: "223e5a1-4b92efe4dfc40"
Accept-Ranges: bytes
Content-Length: 35906977
Connection: close
Content-Type: application/zip
```

# Transfer Encodings

Unlike Content-Encoding (Section 3.1.2.1 of [RFC7231]), **Transfer-Encoding is a property of the message, not of the representation**, and any recipient along the request/response chain MAY decode the received transfer coding(s) or apply additional transfer coding(s) to the message body, assuming that corresponding changes are made to the Transfer-Encoding field-value.



# Chunked Encoding

“The chunked transfer coding wraps the payload body in order to transfer it as a series of chunks, each with its own size indicator, followed by an OPTIONAL trailer containing header fields. **Chunked enables content streams of unknown size to be transferred as a sequence of length-delimited buffers**, which enables the sender to retain connection persistence and the recipient to know when it has received the entire message.”

4.1, RFC 7230

# Chunked Encoding Example

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET /~mln HTTP/1.1
Connection: close
Host: www.cs.odu.edu
```

```
Connection closed by foreign host.
HTTP/1.1 301 Moved Permanently
Date: Mon, 09 Jan 2006 19:32:24 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Location: http://www.cs.odu.edu/~mln/
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

12e

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>
<H1>Moved Permanently</H1>
The document has moved <A HREF="http://www.cs.odu.edu/~mln/">here</A>.<P>
<HR>
<ADDRESS>Apache/1.3.26 Server at www.cs.odu.edu Port 80</ADDRESS>
</BODY></HTML>
```

0

# Chunked Encoding Example 2

```
$ telnet www.cs.odu.edu 80
```

```
Trying 128.82.4.2...
```

```
Connected to xenon.cs.odu.edu.
```

```
Escape character is '^]'.
```

```
GET / HTTP/1.1
```

```
Host: www.cs.odu.edu
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 21 Feb 2006 03:54:31 GMT
```

```
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
```

```
Transfer-Encoding: chunked
```

```
Content-Type: text/html
```

```
5f6
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<!-- saved from url=(0036)http://www.cs.odu.edu/newcssite/new/ -->
```

```
<!-- saved from url=(0019)http://sci.odu.edu/ -->
```

```
<HTML xmlns:st1 = "urn:schemas-microsoft-com:office:smarttags">
```

```
<HEAD><TITLE>Department Of Computer Science</TITLE>
```

```
[lots of html deleted]
```

```
[demo this example to see the various "chunks"]
```

```
0
```

```
$ openssl s_client -connect www.cs.odu.edu:443
CONNECTED(00000003)
[much ssl deletia]
GET / HTTP/1.1
Host: www.cs.odu.edu
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:44:12 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close
```

```
d3
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Redirect</title>
<meta http-equiv="REFRESH" content="0;url=http://www.cs.odu.edu"></HEAD>
<BODY>
Redirecting!
</BODY>
</HTML>
```

```
0
```

```
closed
```

# Chunks visible in raw session

# User agents will sometimes hide the transfer-encoding chunk sizes!

```
$ curl -i www.cs.odu.edu
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:46:59 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
```

Transfer-Encoding header is present,  
but curl has suppressed the byte sizes  
(still hidden with “-v” too)

```
<html>
<meta http-equiv="refresh" content="0; URL='http://odu.edu/compsci'" />
</html>
```

```
$ curl -I www.cs.odu.edu
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 16:47:42 GMT
Content-Type: text/html
Connection: keep-alive
```

HEAD is less useful as well;  
we don't know the representation size  
(both Content-Length and  
Transfer-Encoding are absent)

# Multiple Transfer encodings are possible, but chunked (if present) is always the last

3.3.1, RFC 7230

The Transfer-Encoding header field lists the transfer coding names **corresponding to the sequence of transfer codings that have been (or will be) applied** to the payload body in order to form the message body.

[...]

**If any transfer coding other than chunked is applied to a request payload body, the sender MUST apply chunked as the final transfer coding to ensure that the message is properly framed.** If any transfer coding other than chunked is applied to a response payload body, the sender MUST either apply chunked as the final transfer coding or terminate the message by closing the connection.

For example,

Transfer-Encoding: gzip, chunked indicates that the payload body has been compressed using the gzip coding and then chunked using the chunked coding while forming the message body.

# TE Request Header & Transfer-Encoding Response Header

- Client specifies preferences for transfer encoding in the `TE` header
  - 4.3, RFC 7230
- Server marks the encoding used with the `Transfer-Encoding` header
  - 3.3.1, RFC 7230
- Both headers use the same encoding values available with `Content-Encoding`, plus the special `chunked` encoding and the `Trailers` value

# Trailers requested, but server isn't sending...

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET / HTTP/1.1
TE: gzip;q=1.0, Trailers
Host: www.cs.odu.edu

HTTP/1.1 200 OK
Date: Mon, 27 Feb 2006 15:52:33 GMT
Server: Apache/1.3.26 (Unix) ApacheJServ/1.1.2 PHP/4.3.4
Transfer-Encoding: chunked
Content-Type: text/html

5f6
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0036)http://www.cs.odu.edu/newcssite/new/ -->
<!-- saved from url=(0019)http://sci.odu.edu/ -->
[more html deleted]
```



# Time / Space Tradeoff

- Hard to find examples of compression used in transfer encoding
  - <https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/https://web.archive.org/web/20001019025549/http://webreference.com:80/internet/software/servers/>  
(read w/ curl or lynx; js rendering fails)  
<https://web.archive.org/web/20050114061329/http://www-128.ibm.com/developerworks/web/library/wa-httpcomp/>
  - idea: for very heavy volume web servers, answering the request quickly is more important than preserving bandwidth
- Complexity of management seems to be the limiting factor in compression with content encodings

# Trailer Response Header

- The “Trailer” response header lets the client know that additional headers will appear at the end of the chunked response
  - sections 4.1.2, 4.4, RFC 7230
  - headers can be reconstructed by downstream servers
  - headers that can never be trailers:
    - Transfer-Encoding
    - Content-Length
    - Trailer

# Trailer Example

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2004 11:15:03 GMT
Content-Type: text/html
Content-Length: 129
Expires: Sat, 27 Mar 2004 21:12:00 GMT
```

```
<html><body><p>The file you requested is
3,400 bytes long
and was last modified: Sat, 20 Mar 2004
21:12:00 GMT.
</p></body></html>
```

*“Expires:” response header covered in section 5.3, RFC 7234*

```
HTTP/1.1 200 OK
Date: Mon, 22 Mar 2004 11:15:03 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Trailer: Expires
```

```
29
<html><body><p>The file you requested is
5
3,400
23
bytes long and was last modified:
1d
Sat, 20 Mar 2004 21:12:00 GMT
13
.</p></body></html>
0
Expires: Sat, 27 Mar 2004 21:12:00 GMT
```

Two More Request  
Headers to Process

# Referer Request Header

5.5.2, RFC 7231

The "Referer" [sic] header field allows the user agent to specify a URI reference for the resource from which the target URI was obtained (i.e., the "referrer", though the field name is misspelled). A user agent **MUST NOT** include the fragment and userinfo components of the URI reference [RFC3986], if any, when generating the Referer field value.

Example:

Referer: <http://www.example.org/hypertext/Overview.html>

cf. `rel="noreferrer"` (also note the correct spelling)

<https://stackoverflow.com/questions/50773152/when-should-i-use-rel-noreferrer>

# User-Agent Request Header

5.5.3, RFC 7231

The "User-Agent" header field contains information about the user agent originating the request, which is often used by servers to help identify the scope of reported interoperability problems, to work around or tailor responses to avoid particular user agent limitations, and for analytics regarding browser or operating system use. A user agent SHOULD send a User-Agent field in each request unless specifically configured not to do so.

Example:

User-Agent: CERN-LineMode/2.15 libwww/2.17b3

```
$ curl -I "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/"
HTTP/1.1 503 Service Unavailable
Content-Type: text/html
Content-Length: 2671
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:19:09 GMT
[deletia]
```

# Lying with User-Agent

```
$ curl -I -A "mozilla" "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/"
HTTP/1.1 405 Method Not Allowed
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:19:32 GMT
[deletia]
```

```
$ curl -i -A "mozilla" -s "https://www.amazon.com/Mountain-Has-Fallen-EP/dp/B073JS3Y9Q/" | head -10
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Server: Server
Date: Wed, 17 Oct 2018 17:20:21 GMT
Strict-Transport-Security: max-age=47474747; includeSubDomains; preload
Vary: Accept-Encoding, User-Agent, X-Amazon-CDN-Cache
P3P: policyref="https://www.amazon.com/w3c/p3p.xml", CP="CAO DSP LAW CUR ADM IVAo IVDo CONo OTPo OUR
DELi PUBi OTRi BUS PHY ONL UNI PUR FIN COM NAV INT DEM CNT STA HEA PRE LOC GOV OTC "
Cache-Control: no-cache, no-transform
```

# I'm an iPhone, I swear!

```
$ curl -ILs https://en.wikipedia.org/ | grep -iE "^(http|location)"
HTTP/2 301
location: https://en.wikipedia.org/wiki/Main_Page
HTTP/2 200
```

```
$ curl -ILs -A "iphone" https://en.wikipedia.org/ | grep -iE "^(http|location)"
HTTP/2 301
location: https://en.wikipedia.org/wiki/Main_Page
HTTP/2 302
location: https://en.m.wikipedia.org/wiki/Main_Page
HTTP/2 200
```

cf. <https://www.business2community.com/instagram/post-instagram-computer-02013790>