Introduction

- **The transport layer builds on the network layer to provide da transport from a process on a source machine to a process on destination machine with a desired level of reliability that independent of the physical networks currently in use.**
- **In this chapter, we will study the transport layer in deta including**
  - **Services**
  - **How to tackle issues of reliability, connections**
  - **Congestion control**
  - **Protocols such as TCP and UDP.**

Services Provided to Upper Layer

- The goal of the transport layer is to provide efficient, reliable, and cost-effective data transmission service to its users, normally processes in the application layer.
- Just as there are two types of network service, connection-oriente and connectionless, there are also two types of transport service.
  - Connection-oriented Transport Service (by TCP)
  - Connection-less Transport Service (by UDP)

Services Provided to Upper Layer

- Functionalities
  - Process-to-Process Communication
  - Addressing: Port Numbers
  - Encapsulation and Decapsulation
  - Multiplexing and Demultiplexing
  - Flow Control
  - Error Control

Process-to-Process Communicatio

ADDRESSING: PORT NUMBERS

Encapsulation and Decapsulation

FLOW CONTROL

a. Pushing

b. Pulling

Error Control

- In the Internet, since the underlying network layer (IP) unreliable, we need to make the transport layer reliable if t application requires reliability. Reliability can be achieved to ac error control services to the transport layer. Error control at t transport layer is responsible for
  - Detecting and discarding corrupted packets.
  - Keeping track of lost and discarded packets and resending them.
  - Recognizing duplicate packets and discarding them.
  - Buffering out-of-order packets until the missing packets arrive.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to con
CONNECT	CONNECTION REQ.	Actively attempt to establish a conne
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	Request a release of the connection

Transport Service Primitives

Note: For lack of a better term, we will us
the term segment for messages sent fro
transport entity to transport entity.

Elements of Transport Layer

- The following is also called as the elements of transpor layer
  - Addressing: Port Numbers
  - Multiplexing and Demultiplexing
  - Flow Control
  - Error Control
  - Connection-Establishment and Release (Will be discussed in TCP)

Connection-le
Protocols

- No numbering for each frag
  the two transport layers c
  coordinate with each other.
- Fragments may arrive in di
  order.
- The situation would be worse
  of the fragments were lost.
- We can say that no flow c
  error control, or congestion c
  can be effectively implemente
  connectionless service.

Connection-Oriented Protocols

- At the transport layer, conn
oriented service involves only th
hosts; the service is end to enc
means that we should be able to r
connection-oriented protocol a
transport layer over eithe
connectionless or connection-or
protocol at the network layer.
- We can implement flow control,
control, and congestion contro
connection-oriented protocol.

Transmission Control Protocol

- The Internet has two main protocols in the transport layer, a connectionless protocol and a connection-oriented one. The protocols complement each other
- TCP (Transmission Control Protocol) was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork.
- An internetwork differs from a single network because different parts may hav wildly different topologies, bandwidths, delays, packet sizes, and other parameters.
- TCP was designed to dynamically adapt to properties of the internetwork and t be robust in the face of many kinds of failures.

The TCP Service Model

- All TCP connections are full duplex and point-to-point.
- TCP does not support multicasting or broadcasting.
- A TCP connection is a byte stream, not a message stream.

The TCP Segment

- A TCP segment consists of a fixed 20-byte header (plus an optional pa followed by optional headers followed by zero or more data bytes.
- Two limits restrict the segment size.
  - First, each segment, including the TCP header, must fit in the 65,51 byte IP payload.
  - Second, each link has an MTU (Maximum Transfer Unit). Each segme must fit in the MTU at the sender and receiver so that it can be sent a received in a single, unfragmented packet. In practice, the MTU generally 1500 bytes (the Ethernet payload size) and thus defines t upper bound on segment size.

The TCP Segment Header

- Every segment begins with a fixed-for 20-byte header.
- Optional Headers
- Data of size 65535-20-20=65495.
- Segments without any data are legal a are commonly used for acknowledger and control messages.

TCP Segment Header

- ***Source port & Destination port:*** identify the local end points of t connection.
- ***Sequence Number:*** Numbering of the segments.
- ***Acknowledgement number:*** the next in-order byte expected, not the la byte correctly received.
- ***TCP header length:*** Number of 32-bit words are contained in the T header.
- Next comes a ***4-bit field that is not used***.
- Next 8 bits are used as **FLAGS**.

TCP Segment Header

- ***Window size:*** Tells how many bytes may be sent starting at the by acknowledged.
- ***Checksum:*** Error Control Information for reliability.
- ***Urgent pointer:*** Indicate a byte offset from the current sequence numb at which urgent data are to be found. This facility is in lieu of interru messages.
- ***Options:*** Provides a way to add extra facilities not covered by the regu header including negotiations during connection establishment bett reliability etc..

FLAGS

- 8 flags of each one bit, set to “0” when not used and “1” when used.
- ***CWR & ECE:*** Used to signal congestion when ECN is u
- ECE is set to signal an ECN-Echo to a TCP sender to to slow down when the TCP receiver gets a conge indication from the network.
- CWR is set to signal Congestion Window Reduced the TCP sender to the TCP receiver so that it know sender has slowed down and can stop sending the Echo.

FLAGS:

- ***URG:*** Set to 1 if the Urgent pointer is in use.
- ***ACK:*** Set to 1 to indicate that the Acknowledgement number is valid, else th Acknowledgement field is neglected.
- ***PSH:*** PuSHed data, i.e., The receiver is hereby kindly requested to deliver t data to the application upon arrival and not buffer it until a full buffer h been received.
- ***RST:*** Used to abruptly reset a connection that has become confused due to host crash or some other reason.
- ***SYN:*** Used to establish a connection.
- ***FIN:*** Used to release a connection.

TCP Communication

- A Connection-oriented transport service
  - Connection Establishment
  - Data Transfer
  - Connection Release

TCP Connection Establishment:

TCP connection establishment in the normal case.

Simultaneous connection establishment on both sides.

TCP Connection Establishment:

Data Transfer

TCP CONNECT RELEAS

TCP CONNECT RELEAS

TCP CONNECTION

User Datagram Protocol

- UDP is a connectionless protocol unreliable protocol.
- It does almost nothing beyond sending packets betwe applications, letting applications build their own protocols on top needed.
- UDP provides a way for applications to send encapsulated datagrams without having to establish a connection.
- UDP transmits segments consisting of an 8-byte header followed the payload.

UDP Segment Header

- The two ports serve to identify the endp
  within the source and destination machines.
- The UDP length field includes the 8-byte he
  and the data. The minimum length is 8 byte
  cover the header. The maximum length is 65
  bytes
- An optional Checksum is also provided for
  reliability. It checksums the header, the data
  a conceptual IP pseudo header.

Congestion Control

- If the transport entities on many machines send too many packets into the netwo too quickly, the network will become congested, with performance degraded packets are delayed and lost.
- Controlling congestion to avoid this problem is the combined responsibility of t network and transport layers.
- Congestion occurs at routers, so it is detected and handled at the network layer.
- However, congestion is ultimately caused by traffic sent into the network by t transport layer.
- The Internet relies heavily on the transport layer for congestion control, and spec algorithms are built into TCP and other protocols.

Congestion Control

- Congestion control can be achieved by
  - Desirable Bandwidth Allocation.
  - Regulating the Sending Rate.

Desirable Bandwidth Allocation

- The goal is more than to simply avoid congestion.
- It is to find a good allocation of bandwidth t transport entities that are using the network.
- For both goodput and delay, performance begi degrade at the onset of congestion. Intuitively, w obtain the best performance from the network allocate bandwidth up until the delay starts to rapidly

Desirable Bandwidth Allocation

- How to divide bandwidth between different transport senders
- Ans: give all the senders an equal fraction of the bandwidth. B not right.
- It needs several considerations.
- Hence, the right answer is **“max-min fairness”.**
- The max-min fairness is said to be achieved by an allocation only if the allocation is feasible and an attempt to increa allocation of any participant necessarily results in the decrease allocation of some other participant with an equal or s allocation

Desirable Bandwidth Allocation

- The bandwidth needed by a given connection will vary over time the connections are always coming and going.
- A good congestion control algorithm should rapidly converge to t ideal operating point, and it should track that point as it changes ov time.
- If the convergence is too slow, the algorithm will never be close to t changing operating point.
- If the algorithm is not stable, it may fail to converge to the right poi in some cases, or even oscillate around the right point.

Regulating the Sending Rate

- How do we regulate the sending rates to obtain a desirable bandwid allocation?
- The sending rate may be limited by two factors.
  - The first is flow control, in the case that there is insufficient buffering at the receiver.
  - The second is congestion, in the case that there is insufficient capacity in the network.
- The way that a transport protocol should regulate the sending rate depends on t form of the feedback returned by the network.
- The feedback may be
  - Explicit or Implicit
  - Precise or Imprecise.

Protocol	Signal	Explicit?	Precise?
XCP	Rate to use	Yes	Yes
TCP with ECN	Congestion warning	Yes	No
FAST TCP	End-to-end delay	No	Yes
Compound TCP	Packet loss & end-to-end delay	No	Yes
CUBIC TCP	Packet loss	No	No
TCP	Packet loss	No	No

Regulat the Sen Rate

Chiu and Jair studied the binary cong feedback and that AIMD (A Increase Mult Decrease) appropriate co to arrive at the and fair operat

DOMAIN NAME SYSTEM

- Remembering and using the IP addresses to access a servi
hard for people.
- Moreover, what happens? if a company moves the server t
different location with different IP address.
- Consequently, high-level, readable names called URLs were
introduced in order to decouple machine names from mac
addresses.
- But, since a network can only use numerical addresses, so
mechanism is required to convert the URLs to IP addresses

DOMAIN NAME SYSTEM

- For this purpose Domain Name System (DNS) was introduced in 1983.
- The simple working of a DNS was shown in the Fig.
- When a user enters an URL to access a http server, at first a request will be sent to the DNS server.
- The DNS server will send the replay with the IP address of the requested URL.
- Now a http request will be sent to the web server using the IP address.

DNS: Name Space

- Managing a large and constantly changing set of names is a nontrivial problem.
- Hence, like in postal system, the DNS name space is structured in hierarchical manner.
- For the Internet, the top of this naming hierarchy is managed by ICANN (Internet Corporation for Assigned Names and Numbers).
- Conceptually, the Internet is divided into over 250 top-level domains, where each domain covers many hosts.
- Each domain is partitioned into subdomains, and these are further partitioned, and so on.

DNS: Domain Resource Records

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

- Every domain, can have a set of resource record associated with it. These records are the DNS database.
- A resource record is a five-tuple.
  - Domain name| Time to live | Class | Type | V
  - ***Domain name***: The domain to which this reco applies.
  - ***Time to live***: Gives an indication of how stable record is.
  - ***Class***: For Internet information, it is always IN
  - ***Type***: The kind of record.
  - ***Value***: This field can be a number, a domain n or an ASCII string.
- Ex: cs.mit.edu | 86400 | IN | CNAME | csail.mit.edu

DNS: Name Servers

- The DNS name space is divided into nonoverlapping zones, so that the entire DNS informat is distributed among these zones.
- The process of looking up a name and finding an address is called **name resolution**.
- When a **resolver** has a query about a domain name, it passes the query to a local name serv
- If the domain being sought falls under the jurisdiction of the queried name server, it will se the record back to the resolver.
- If not, i.e., when the domain is in a remote location, the local name server begins a rem query from the top of the hierarchy, i.e., it queries one of the root server and follow **recursive querying**.
- The remote query process is as shown in the figure.

DNS: Name Servers

DNS Message

0
16
31
Header
Identification
Flags
Number of question records
Number of answer records
(All 0s in query message)
Number of authoritative records
(All 0s in query message)
Number of additional records
(All 0s in query message)
Question section
Answer section (Resource Records)
Authoritative section
Additional section

ELECTRONIC MAIL (E-Mail)

- Email, like most other forms of communication, has developed its own conventions and style
- The email protocols have evolved during the period of their use, too.
- The first email systems simply consisted of file transfer protocols, with the convention that first line of each message (i.e., file) contained the recipient’s address.
- Later, email diverged from file transfer and many features were added, such as the ability send one message to a list of recipients, sending multimedia messages, etc.

Architecture and Services

Architecture and Services

Architec
and Serv

E-Mail: User Agent

- A user agent is a program (sometimes called an email reader) that accepts a variety commands for composing, receiving, and replying to messages, as well as for manipulat mailboxes.
- The lines use the From, Subject, and Received fields, in that order, to display who sent message, what it is about, and when it was received.
- After a message has been read, the user can decide what to do with it. This is called messa disposition.
- Filing can be done automatically by the user agent as well, before the user reads the messag

E-Mail: User Agent

Message Formats

- The Internet Message Format
  - Messages consist of a primitive envelope some number of header fields, a blank line, and then the message body.

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	Email address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display

Message Transfer

- The mail transfer is done with the SMTP (Simple Mail Transfer Protocol).
- Within the Internet, email is delivered by having the sending computer establish a T connection to port 25 of the receiving computer. Listening to this port is a mail server t speaks SMTP.
- SMTP is a simple ASCII protocol.
- A simple SMTP uses various commands
  - HELO
  - RCPT
  - FROM

Message Transfer : SMTP

- Commands are sent from the client to the server.
- The format of a command is shown below:

Keyword: argument(s)

Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing li
HELP	Command name	Asks the recipient to send information abo the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *or* the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *and* the mailbox of the recipient

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command

Message Transfer : SMTP

- Responses are sent from the server to the client.
- A response is a three digit code that may followed by additional textual informatio

Message Transfer: SMTP

Code	*Description*
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Message Transfer

- Mail Submission:
  - User agents often run on laptops, home PCs, and mobile phones.
  - Mail transfer agents run on ISP and company servers.
  - SMTP is normally used for mail submission with the AUTH extension.
- Message Transfer:
  - Once the sending mail transfer agent receives a message from the user agent, it will deliver it to receiving mail transfer agent using SMTP.
  - To determine the correct mail server to contact, DNS is consulted.
  - The sending mail transfer agent then makes a TCP connection on port 25 to the IP address of the r server to reach the receiving mail transfer agent, and uses SMTP to relay the message.
  - The receiving mail transfer agent will then place mail in the correct mailbox for receiver to read it later.

Message Transfer

- The process of transferring a mail message occurs in three phases: connection establishment, mail tran and connection termination.

Message Access

- Final Delivery:
  - Our mail message is almost delivered. It has arrived at receiver’s mailbox.
  - All that remains is to transfer a copy of the message to receiver’s user agent for display, when receiver comes online.
  - Here, the job of the user agent is to present a view of the contents of the mailbox, and to allow mailbox to be remotely manipulated.
  - The final delivery of message will be done by:
    - Post Office Protocol (POP v3).
    - Internet Message Access Protocol (IMAP v4).

Final Delivery

- POPv3:
  - The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
  - The client opens a connection to the server on TCP port 110.
  - It then sends its user name and password to access the mailbox.
  - The user can then list and retrieve the mail messages, one by one.
  - POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.

Final Delivery

- IMAP4:
  - IMAP4 is similar to POP3, but it has more features;
  - POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server.
  - In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.
  - IMAP4 provides the following extra functions:
    - A user can check the e-mail header prior to downloading.
    - A user can search the contents of the e-mail for a specific string of characters prior to downloading.
    - A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
    - A user can create, delete, or rename mailboxes on the mail server.
    - A user can create a hierarchy of mailboxes in a folder for e-mail storage.

MIME

- The Multipurpose Internet Mail Extensions
  - In the early days of the ARPANET, email consisted exclusively of text messages written in English and expressed in ASCII.
  - In the 1990s, the worldwide use of the Internet and demand to send richer content through the mail system meant that this approach was no longer adequate.
  - The solution was the development of MIME (Multipurpose Internet Mail Extensions).

E-mail header
MIME-Version: 1.1 Content-Type: type/subtype Content-Transfer-Encoding: encoding type Content-ID: message ID Content-Description: textual explanation of nontextual contents
E-mail body

MIME

- MIME defines five headers, which can be added to the original e-mail header sectio to define the transformation parameters:

Message Formats

Content-Type: This header defines the type of data used in the body of the message.

- The content type and the content subtype are separated by a slash.

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Type	*Description*
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

Messag
Formats

- Content-Tran
  Encoding Thi
  defines the m
  used to enco
  messages int
  1s for transp

Web-Based Mail

- E-mail is such a common application that some websites today provide this service to anyone who accesses the site. Three common sites are Hotmail, Yahoo, and Google mail.

Case 1: Only receiver uses HTTP

Case 2: Both sender and receiver use HTTP

E-Mail Security

- E-mail exchanges can be secured using two application-layer securities designed in particula for e-mail systems.
- Two of these protocols are:
  - Pretty Good Privacy (PGP)
  - Secure/Multipurpose Internet Mail Extensions (S/MIME)

PRETTY GOOD PRIVACY (PGP)

MSKR

TELNET: TErmina NETwor

- A remote l protocol o Applicatio
- It is a non- protocol w an attacke eavesdrop can even g username password for loggin server.

TELNET: NVT Chars

TELNET

- TELNET lets the client and server negotiate options before or during the use of the service.
- The operating system (UNIX, for example) defines an interface with user-friendly commands.

Command	*Meaning*	*Command*	*Meaning*
open	Connect to a remote computer	**set**	Set the operating parame
close	Close the connection	**status**	Display the status inform
display	Show the operating parameters	**send**	Send special characters
mode	Change to line or character mode	**quit**	Exit TELNET

World Wide Web

- First proposed by Tim Berners-Lee in 1989 at CERN for sharing Nuclear Research.
- The commercial Web started in 1990.
- Web is a Repository of Documents known as ***Webpages***.
- The link of a webpage is achieved by ***hypertext*** where now a days was changed to ***hypermedia.***

World Wide Web

- Client: Browser
- Distributed Architecture of S

World Wide Web

- URL:

protocol://host/path	**Used most of the time**
protocol://host:port/path	**Used when port number is needed**

- Web Documents:
  - Static Documents
  - Dynamic Documents
  - Active Documnets

HyperText Transfer Protocol (HTTP)

- The HyperText Transfer Protocol (HTTP) is used to define how the client-server programs can written to retrieve web pages from the Web.
- The server uses the port number 80 and the client uses a temporary port number.
- HTTP uses the services of TCP.
- During a HTTP communication, if the required objects are located in different servers, then client need to establish different TCP connections with each server.
- If the objects are in the same server:
  - Nonpersistent Connection: A new TCP connection is used for each object.
  - Persistent Connection: One TCP connection is used for all objects.

HyperText Transfer Protocol (HTTP)

HTTP: Message Formats

- Request Message:
  - Request line
    - Method : the request type
    - URL
    - Version : Used HTTP Version
  - After the request line, we can have zero or more request header lines.

HTTP: Message Formats

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

HTTP: Message Formats

- Request Message:
  - After the request line, we can have zero or more request header lines. Each header line sends additional information from the client to the server.
  - For example, the client can request that the document be sent in a special format.
  - The body can be present in a request message. Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Header	*Description*
User-agent	Identifies the client program
Accept	Shows the media format the client can accep
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can ha
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication proto
Cookie	Returns the cookie to the server (explained la
If-Modified-Since	If the file is modified since a specific date

HTTP: Message Formats

- Response Message
  - Status line
    - Version: HTTP Version
    - Status Code: 3 Digit Code
      - Up to 100 : Informational
      - 101 to 200: Successful Request
      - 201 to 300: Redirecting Client
      - 301 to 400: Error in Client side
      - 401 to 500: Error in Server Side
    - Status phrase: Explanation of code in text.

HTTP: Message Formats

- Response Message
  - Header lines
    - Additional information from server to client.
  - Body (Optional)
    - The document to be sent from the server to the client.

Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protoc
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to anothe
Accept-Ranges	The server will accept the requested byte-rang
Last-modified	Gives the date and time of the last change

HTTP: Example