1. If a website domain *green.com* has been setup such that it is running from a server machine in Virginia, USA. Is it possible for a subdomain *mail.green.com* to run from Netherlands ? Yes/No. Validate your claim.

   **ANSWER :**

   Yes. Domain names are hierarchical and it is possible to setup DNS with an A record for green.com to point to a machine/IP in USA eg. 54.6.8.22 and its subdomain mail.green.com can be setup to point to another machine/IP in Netherlands eg. 117.20.2.2. In this case mail.green.com can be a mailserver in which case a MX record can be setup to offload mail traffic to different geographical regions of the world.

   But note, only the main domain falling in top level domain – eg. green.com – has to be registered/purchased from the ISP. All subdomain setup is completely within the control of the person/company purchasing the main domain.

2. TCP RST flag is sent from endpoint A to B indicating that the TCP connection between them has to be brought down right away. Is it possible for endpoint C to intentionally send a TCP RST segment to endpoint B to bring down A-B's established TCP connection? Yes/No. Validate your claim and if Yes, how is this problem mitigated?

   **ANSWER :**

   Yes, it is possible but quite difficult practically. That would mean Endpoint C is able to sniff packets going between endpoint A and B. (at least one end)., so that Endpoint C knows the sequence numbers and Ack numbers exchanged so that it can generate a real looking RST packet. In fact many stateful firewalls and routers do generate TCP RST to forcefully end connections. Yes it is a problem of security for which IPSEC and RFC 2385, 5961 propose many improvements.

3. I did a ping to website www.cctv.com - which is a domain registered in China - from my Ubuntu 12. machine in India. It went to 58.26.1.65. Then I did a ping immediately to cctv.com, it resolved to 202.108.8.82. Explain this anomaly.

   **ANSWER :**

   No anomaly here. As can be seen, it is not necessary that www.domain.com and domain.com resolve to the same IP. But in general in common websites, both seem to be pointing to the same website. Also note, given load balancing of servers, it might be quite possible that both those IPs represent a webserver pool or CDN pool for that domain – cctv.com

4. I ran a Wireshark trace on my Windows 7 machine with a LAN IP 192.168.3.100. I saw a lot of ARP requests like 'Who has 192.168.1.165? Tell 192.168.3.9', etc. with different IPs which are not my machine's IP. What is going on? Good/Bad.? Do you propose any changes to my network?

   **ANSWER :**

   This is quite normal mode of LAN/Layer2 operation. ARP Request is broadcasted which is the reason these packets are received/seen by all the machines in the LAN. As can be seen even though both the ARPer's IP and ARPing IP are different from the machine's IP, this traffic will be seen. If too much of ARP flooding for a particular IP is seen, may be that can be investigated. . But, no specific changes are required for the network in question.

** Note the LAN subnet mask is not given in the question. So it can be argued that the network configuration may not be ok – may be overlapping – as both 192.168.1.0 and 192.168.3.0 networks are framed in the same medium, but that does not hold much water as it is quite possible that both the subnets are part of the same LAN and machine 192.168.1.165 may be able to directly talk (MAC level) to 192.168.3.9.

5. My friend Ram says if we do DNS lookup over TCP instead of UDP, things will be much faster, as in TCP there is no data loss. Do you agree/disagree with Ram? Why ? Explain.

**ANSWER :**

No. I do not agree with him and moreover this is a common misconception that TCP is faster than UDP. It is not the case as technically both TCP and UDP segments are carried by Layer 3 IP packets and are limited by the Layer 1 Line badwidth. In fact, if anything, TCP has head of line blocking and until the connection is setup the actual data is not sent out, wherein UDP may turn out to be faster in such cases, which is also one of the reasons why DNS lookup over UDP is favored and is effective.

6. Assume that initially Endpoints A and B are in TCP-connected state. Endpoint A is closing the connection to Endpoint B by sending a TCP FIN packet. Assume this packet never makes it Endpoint B. What happens further at both endpoints in terms of TCP state machine.

**ANSWER :**

When A is sending a TCP FIN to B, it indicates A is not going to send any more data to B. Endpoint A goes into a FIN_WAIT_1 state and if does not a get back an ACK, it retransmits the FIN. Interesting to note that in FIN_WAIT_1 state, endpoint A is not sending any data but can continue to receive data from endpoint B. So endpoint B will continue to be in ESTABLISHED state until it eventually receives the FIN from endpoint A.

**** No mention of ESTABLISHED, FIN_WAIT will mean 0 mark credit.**

7. A Layer-2 network employing CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time is 25.6 micro seconds, what is the minimum size of the frame?

**ANSWER :**

Minimum size of the frame should be selected such that, transmission time of such a frame should be at a minimum, TWICE the Maximum propagation time on the medium. This is because, when an endpoint A is sending out a frame it will hear/sense a collision before it finishes its transmission and that is what makes CSMA/CD work.
Therefore,
( 10 x 10 ^ 6 bits/sec ) x ( 25.6 x 10 ^ -6 sec ) = 256 bits = 32 bytes
Minimum frame size = 32 x 2 = 64 bytes

8. In my company email id, my@company.com, I receive a lot of mails from my friends and colleagues. What do you suggest to me, so that I can figure out if a mail I received, has indeed been sent by the sender - *sender@senderdomain.com* as claimed in the mail - Authenticity.

**ANSWER :**

Authenticity of Email is an interesting subject area these days. In general email is not secure. To the big part, the blame is on the overly simplistic SMTP which is an old protocol which has for most part remained the same all these years to carry email traffic between mail servers.

So I suggest you to probably check with the sender - the human behind the email id – whether he did indeed sent the mail, before reacting on it. Of course, if possible look into the PGP security to secure email traffic which requires both the sender and receiver are in agreement apriori the email transmission.

On the other hand, if you are receiving mails from another company with domain company.com, you can invest in setting up SPF records in the DNS and/or Reverse DNS PTR records for company.com's mailserver, and implement server solutions on both sides which check for SPF records on the fly.

Note, it is quite trivial for an imposter to send a plain mail to your email id claiming as being sent by sender@senderdomain.com as it is just a From: header field in the email.

```
Message-ID: <20cf305792797592345234623462@google.com>
Date: Sun, 02 Feb 2014 18:02:12 +0000
Subject: News for you
From: chris@news.com
To: my@company.com
…
```

**** Email security is interestingly still an interesting/evolving area and many other solutions are suggested these days. So if the answer is talking about any other relevant mechanisms like anti SPAM filters like SPAM-Assassin, etc. marks can be credited.**

9. Differentiate a router asking an endpoint to go slow with data transmission and the remote endpoint doing so. How is this achieved? Which protocol(s) is involved? Illustrate with examples.

**ANSWER :**

An overloaded Router can ask an endpoint to go slow by sending ICMP Source quench messages – Network Layer Congestion Control mechanism.

Remote Endpoint can ask a source endpoint to go slow by setting the Receiver WINDOW size accordingly in TCP Header – This is called Flow control. Of course this is not possible if the transport layer protocol employed is UDP.

10. The SSH daemon running on Linux has a config file */etc/ssh/ssh_config* where we can set '*ServerAliveInterval*' with a value like 30 seconds such that server sends a No-Op packet to the connected client to detect if the session is still active. How is TCP Keepalive different from this scheme? Also, given that SSH runs on top of TCP, does it make sense to have both enabled? Why? Why not?

**ANSWER:**

TCP Keepalive is transparent to Application layer keep-alive and will not require any additional application layer logic and the Application layer protocol can be kept free from connection maintaining or keep alive logic which is quite cumbersome.

But on the flipside TCP Keepalive is usually differently implemented on different host Operating systems. So the mileage may vary significantly and affect the proper functioning of the Application layer protocol which requires long running connections.

TCP Keepalive packet has the SEQ number as ( SEQ next – 1) where SEQ next is the next expected SEQ number by the other side endpoint as indicated in the most recent ACK. So in that regard it is like sending the last 1 byte again, which is evidently treated as Keepalive and ignored by the remote side TCP endpoint.

It sure sounds like an overkill to have both SSH and TCP enabled as once the SSH keepalive message is sent out, it will reset the TCP Keepalive timers as even an SSH keepalive is a payload for TCP just like any other SSH message exchanged.

11. I booted my Windows system and assigned it an IP via Network Adapter settings as 192.168.3.100 and immediately it threw an alert box which read 'Windows has detected an IP Address conflict' and did not allow me to set that IP. I was intrigued. Explain how it detected.

**ANSWER :**

My Windows system most likely sends an ARP with the given IP as the ARPing IP. Interestingly, this is one of the scenarios, where the sender system knows the ARPing IP, but rather wants to check and see if any system has been assigned this IP. It waits for a while and if a reply is received from any other system in the LAN, it clearly means this IP is being used in which case the host OS, throws the 'IP Conflict' or 'IP already in use' error message.

**** Even though this is a very common scenario, the correct technical reasoning behind this is what happens at Layer 3 and Layer 2 with the help of ARP mechanism.**
**** Any answer without mention of ARP will deserve only 0 mark credit.**

12. Suppose two machines A and B are both behind DIFFERENT NATs. Is it at all possible that A can setup a TCP connection to B or vice-versa. Yes/No. Explain why/why-not? If you side with 'No', how do you propose a scheme so that A and B can transfer a file between them in best possible time.

**ANSWER :**

**NO. Not possible.**

In the first case if A's IP is 192.168.1.4 and B's IP is 10.1.1.5, both are private IPs and are not directly in same LAN so cannot be connected.

Even if A and B know their NAT public IPs, connection setup is not possible.
Consider A connecting to B.

A's TCP SYN will need to be received by the other endpoint B and in turn the other endpoint's SYN + ACK should be received back by A for the TCP connection to be setup.

But in this case, A's TCP Syn sent to B's public IP will go out via A's NAT router – TCP hole punching, but when it gets routed and reaches B's NAT router, it will be discarded as there will be no NAT entry.

**The only static solution for this is the NAT router to have a Port forwarding rule setup, but again it is not a dynamic solution and will fail if the IP of machine or TCP ports change.**

**The other cumbersome mechanism is to have Forwarding or RELAY server at the Application layer level.**

13. I launched Wireshark on my Windows XP - Dell Vostro machine with Dell BIOS and started a packet capture. My machine is in a LAN with multiple subnets. Many of the Ethernet packets were tagged by Wireshark as coming from a MAC address 00:8E:F2:0A:F5:77 which it identified as Netgear_0a:f5:77 Interestingly, our LAN router is Netgear brand and these packets were indeed coming from that router to my machine. Explain how Wireshark identified the router's brand.

**ANSWER :**

Well, the fact is that Ethernet MAC addresses are FLAT and not hierarchical as IP addresses. What this means is that my machine may have a MAC address - 00:8E:F2:0A:F5:77 – and the next MAC address 00:8E:F2:0A:F5:78 may be for a machine in Sri Lanka ! So technically there is no ordering required for MAC addresses. But, to avoid duplicate MAC addresses for devices, they are assigned

to different vendors/device manufactures in blocks – in some ways similar to Public IP address assignment to companies/enterprises.

So looking at the initial set of bytes in a MAC address, it is possible to detect the vendor of the device. Of course, this database continues to be updated as more blocks are assigned, so Wireshark may detect known MAC address blocks as in this case – it detected that the MAC address matches Netgear – a popular router vendor. Wireshark may have an internal static database/table/map with which it matches. So it may also fail to detect the vendor if there is no match found in its internal map/table.

14. I would like to setup my personal website available at these 4 domain names xyz.com, gotoxyz.com, cometoxyz.com, atxyz.com Propose a scheme so that duplication of website data is avoided and I am able to track which user came via which domain. Note that all 4 websites should show the same content at all times.

**ANSWER :**

1. All the four domains have to be Registered with a Domain REGISTRAR separately as these are no subdomains but main domains under the TLD .com
2. To avoid Duplication of website data, we can run one server machine with the website contents and then assign it a Public IP and setup the DNS A record for all the 4 websites to point to this one and only IP. That way even if users are typing different URLs/Domains in the browser, it will resolve to the same IP and will get connected to the same machine/server.
3. One other way is to do load balancing on the four domains, such that 4 different servers with individual IPs will be published in the DNS. But all 4 machines/servers will access the website data from a network-mounted file system.

15. Typically, we are used to login to websites - be it ecommerce, email, or some other service website. Given that HTTP the underlying protocol is said to be stateless, how is this made possible?

**ANSWER:**

HTTP is stateless which means each request is independent and no ordering is required between requests. Eg. If a website has 4 images under <IMG > tag in HTML, it is completely up to the browser (or any other logic in that page) to fire 4 independent HTTP GET requests in any particular order for those 4 images.

Interestingly, servers need to know whether to honor a request or not by assessing/detecting whether a user is logged in or not. For example, if you try to request for an image URL when not logged in, server might return a response showing the login page. So as requests are independent the question arises, how the server would know. There needs to be 'something' in the GET Request for the image URL that indicates to the server that the request comes from a logged in client. This 'something' is termed as 'HTTP Cookie'. It is an extra header in the HTTP GET Request that is sent to the server. This cookie is a string that is received from the server typically as part of the login response to the client, which the client is expected to send in all subsequent requests so as to indicate to the server that the client is logged in. Cookies are domain specific and stored internally by the browser.

**** Answers with No mention of the word 'HTTP Cookie' will receive ZERO credit.**

16. WinSCP is a software to transfer files over SCP/SFTP to remote machines. I have a WinSCP connected to a remote server from machine A. Is it programmatically possible for me to run another program in the same machine A to send a file over the connection used by WinSCP? Yes/No. Validate your claim.

**ANSWER :**

**NO. Not possible.**
**A program cannot inject data into another program's connected TCP socket.**

SCP/SFTP is an application layer protocol running on top of TCP. So a WinSCP session uniquely translates to a underlying TCP connection from machine A to server – which is identified by the quadrapule – source TCP Port, source IP Address, destination TCP Port and destination TCP Address.

Only the connected socket will be able to send data in the connection. So this socket will be typically created by WinSCP or the software and will be available only to it. So it is not programmatically possible for me to inject data into WinSCP's TCP socket or connection.

Interestingly, it is plain technical common sense to understand if at this is possible it would mean any rogue software will be able to misuse this feature and send spurious data to another program and effectively make the other program fail. So, this is not possible, not required.

16. My Linux RHEL6 machine has MAC Address for 'eth0' as 68:5B:35:89:88:46 and IP address as 192.168.1.40. The default gateway(router) is set to 192.168.10.1 and the LAN subnetmask is 255.255.0.0 Interestingly, the DNS nameserver on the machine has been set to 192.168.10.1

    and Internet browsing is working fine in the machine. Now list out the steps that happens when I launch the Firefox browser to login to my bank account and type in the URL https://www.iobnet.co.in Feel free to assume other points of interest but do list them and talk about their relevance.

**ANSWER :**

1. **Browser constructs to send out a HTTP GET packet to www.iobnet.co.in – so it needs the destination IP of iob server so that it can issue a TCP connect to that machine.**
2. **To get the IP address of iob server, DNS A Record Lookup for** www.iobnet.co.in needs to be sent out first.
3. **Note the fact the DNS server also happens to be the Default gateway 192.168.10.1 in this setup. So to send the DNS packet to 192.168.10.1, my machine needs to know the MAC address of the default gateway 192.168.10.1 – So it sends out an ARP request with ARPing IP as 192.168.10.1**
4. **Hopefully, my system receives an ARP response from the router and gets to send out the DNS A record lookup packet which is typically carried in UDP.**
5. **Hopefully, a DNS answer is received by my system shortly where it may contain www.iobnet.co.in maps to an IP eg. 157.7.7.7**
6. **So the TCP Connection is setup – SYN packet sent out to 157.7.7.7**
7. **Note the URL used here is https – so typically even before the HTTP GET packet can be sent out – SSL/TLS negotiation between my system/browser and the iob webserver happens. It may also require out-of-band SSL certificate verification**
8. **Once the SSL/TLS channel is set up, the first HTTP GET packet can be sent out to the iob webserver.**
9. **After that, depending on the webpage sent back in the 200 OK HTTP response by the iob webserver,**

18. A router is a contraption to forward packets from one LAN to another. Consider the case of a route on the edge of a LAN connecting to the Internet/WAN. List out the activities/steps involved in the router's mechanism when an IP Packet is received via the i) LAN Port ii) WAN Port Enumerate the significant differences.

**ANSWER :**

**Steps (not an exhaustive list) – Receiving an IP Packet on the LAN side**

1. First look at IP Header Version field and process based on IPv4 or IPv6 packet rules
2. IP Checksum verification on the incoming Packet
3. Check if the destination IP matches the router's IP interfaces. If so it is a packet destined for the router. Eg. Trying to open the mgmt. interface in the router via telent/http.
   Or a simple ping request (ICMP) of the router IP.
4. If the destination IP is to an external IP/Public IP NAT translation mechanism kicks in.
5. IP Header TTL checks are carried out and in case of failure ICMP TTL expired in Transit messages sent back to the sender IP.
6. IP Header length is checked to determine the IP Packet length and accordingly IP options processing happens.
7. If the packet needs to be forwarded then TTL is decremented and a new IP packet is generated and header checksum is recalculated.
8. If the packet is coming from the LAN machines – source IP check – and the destination IP is a public IP, and the Protocol field is checked to be TCP/UDP and then NAT forwarding is done by setting the source IP, as the NAT IP on the router which is a public IP. The source Port being set depends on the the type of NAT translation in the router firmware and it might be Symmetric NAT or cone NAT or port restricted nat, etc.

**Steps (not an exhaustive list) – Receiving an IP Packet on the WAN side**

1. IP Header Version check
2. destination IP is looked into and the decision to route the packet is taken
3. If the destination IP matches the NAT IP of the router, NAT table is consulted and if a matching entry based on the source port, destination port, source ip, destination ip and protocol (TCP/UDP) is found the packet needs to be sent to that internal LAN IP. But before doing so, a new packet is generated changing the destination IP/Port accordingly and header checksum is recalculated.
4. If no match in NAT table, then such packets are dropped silently.
5. If unable to send the packet to the internal LAN IP, ICMP errors Destination Unreachable are generated and sent back to the source IP.
6. If packet needs to be forwarded externally, the route table is consulted based on the dest.IP and if the packet is bigger than the path MTU, fragmentation kicks in.

** **Mention of NAT is important.**

** **Mention of DHCP or Broadcast/Multicast packets will need to be awarded positive credit.**

** **Mention of IP Header level details is a MUST – answers without that and in general English stories like Router processes the IP packets by destination address, etc. will need to be credited only ZERO marks.**

19. Router-R's routing table is as below

| **Address/mask** | **Next Hop** |
|---|---|
| 135.46.56.0/22 | Interface 1 |
| 135.46.32.0/19 | Interface 2 |
| 135.46.60.0/22 | Interface 3 |
| 192.53.40.0/23 | Router 1 |
| Default | Router 2 |

If incoming packet came with these following destination IPs, how would the packet be forwarded?
(a) 135.46.63.10 (b) 135.46.52.2 (c) 192.53.56.7

Show clear Bit Arithmetic.

**ANSWER :**

x.y.z.0/t in CIDR form means the prefix has t bits. A route with t + 1 bits is a more specific route and has to be selected as the ***to-be-forwarded-route*** - longest prefix match, when multiple matches.

So writing in binary, in the case of Entry1,
135.46.56.0/22 10000111 00101110 00111000

Entry2,
135.46.32.0/19 10000111 00101110 00100000

Entry3,
135.46.60.0/22 10000111 00101110 00111100

a) 135.46.63.10 in binary 3 octet form is 10000111 00101110 00111111
The first 22 bits have to match for entry1, but it is matching only till 21 bits, so a) does not match entry1 But Matches with entry2 which is 19 bits prefix.but also matches with Entry3 which is a longer prefix 22 bits. So the match for a) is entry3 and routed via Interface 3

b) 135.46.52.2 In binary 3 octet form is, 10000111 00101110 00110100
Matches only with Entry2. And is routed via Interface 2.

(c) 192.53.56.72 In binary 3 octet form is, 11000000 00110101 00111000
Routing table entry4 is 192.53.40.0/23 translates in binary to,
11000000 00110101 0010100 0

So it does not match Entry4 and will take the DEFAULT route via Router2.

20. I took a full-sized photo at a studio and the person mailed me the photo which was in HD - High Definition - jpg format - approx. 24 MB. If the studio was using an Internet connection of 512 Kbps approximately how much time was spent sending that mail? Note, this studio person was sending the mail from Google Chrome browser running on Windows XP using his gmail account to my mail user@somecompany.com which is located in Netherlands. Do you think it would have taken less time if my company's mailserver/domain was located in India? How about if I had asked the person to mail it to my gmail email account instead of my company mail. Does compression affect things here? Yes/No. Reason out your stand.

**ANSWER :**

As the mail is being sent via a web interface – in this case google's gmail interface via the browser, this becomes the Mail User Agent (MUA). The big time factor in the equation is the time it takes to upload the photo from the browser to the gmail server, which is limited by the speed of the Internet connection.

Assuming, best connectivity at the time of upload/email, 64 KB/sec. we arrive at time taken as below,

Time taken approx. = 24 x 1024 KBytes / 64 KB = 384 secondsroughly 6 minutes.

From there, it is the google mail server's Mail Transfer Agent (MTA) responsibility to send the mail to user@somecompany.com which being Netherlands does not directly affect the mail uploading process as seen in the web interface of gmail.

- No. company mail server being in Netherlands does not affect the time equation (in this case)
- The second part is that, even if I had asked the studio person to mail it to my gmail id, it might not have helped things. Though being in the same domain, there is a good likelihood that I received the mail much faster than the 1st case where the mail was sent to another domain.
- Yes. Compression definitely helps in these cases and before uploading the photo as an attachment compressing the photo (zip, rar) may have reduced the attachment size and so could bring down the time spent in sending the mail. But to be noted is that the photo is already in jpg format, which means it is already compressed. So in this case once again zip or rar compressing the file may not have helped matters much.
- The other big point is that mail attachments are MIME based and typically base64 encoded which means an increase in size of order 1.33. So if the mail attachment was client size base 64 encoded before uploading it might have been 24 x 1.33 = 32 MB to upload which boils down to 512 seconds instead of 384 seconds – though it is most likely not done that way in gmail's web interface rather the attachment would be encoded on the server side.

21. Describe all possible different scenarios when an IP router would drop packets that arrive on one of its interfaces. Does it need to indicate that a packet was dropped ? If so, to whom and how ?

**ANSWER :**

1) Any policy that instructs the router to drop it. This is a "rule" configured intentionally by the router's administrator to control traffic.
2) If the IP Packet version was 6 and IPv6 was not supported or disabled in the router.
3) The packet fails an RPF (Reverse Path Forwarding) check to avoid circuitous routing or flooding. When a packet arrives with a source address that doesn't make sense, and it appears like an attempt at malicious activity.
4) The input queue/buffer is full. This is basically a symptom of the router being overloaded.
5) Expired TTL (Time to Live)
6) IP Header Checksum verification fails which indicates a data-corrupted IP Packet.
7) Failed CRC (Cyclic Redundancy Check), although that is actually a Layer 2 function, not IP.
8) If the packet has Donot Fragment Bit set and if the path MTU on the to-be-forwarded interface is much smaller than the IP Packet size.
9) The packet has a requirment that the router is unable to accomodate. Examples of this would be a non-UDP encapsulated IPSec packet whose destination address requires translation, or a multicast packet arriving at a router that is not configured for multicast routing.
10) Packet that comes with a destination IP in the private IP address range eg. 192.168.0.0/16, etc.

Not always an indication is required, but in many cases the router generates an ICMP error message back to the sender – source IP of the IP packet that had to be dropped. Eg. ICMP Message TTL expired in transit. Etc.

22. A HTTP Request contains the 'Host' header containing typically the domain of the connecting server eg. sub.abcd.com. This looks redundant as the server typically knows its own domain. What useful purpose this field serves ? Or do you think it is a legacy field? How does HTTP caching work in the browser and what is the effect when server and clients are not time synchronized? Do you see any issues there, if so how are they addressed ? Illustrate with an example.

**ANSWER :**

No. It is not a legacy header and is very important and mandatory header in the HTTP GET Request.

If a single machine has been setup in IP 15.15.15.15 and is running both websites abcd.com and efgh.com, it is very important that the server is able to differentiate from the incoming HTTP GET request whether the request came for domain abcd.com or efgh.com

Interestingly, with Virtual instances and Web hosting, this has interestingly become a very important field for things to work, though in the past when a single machine served only one website, this was not all that important except to weed away spurious/erroneous GET Requests.

HTTP Caching is based on Cache-Control header in the HTTP Response where usually the server tells the client how long – if at all – a resource can be cached in the max-age field. Interestingly, the other mechanism is by having a ID for the resource generated by the server. This ID is sent via the Etag field in the server response.

So clients when requesting again the same resource, send the 'If-Modified-Since' header,
Eg. If-Modified-Since: Wed, 20 Jan 2014 00:56:48 GMT

If the resource is modified it is sent back by server, if not server replies with a
HTTP/1.1 304 Not Modified

This way the amount of data transferred specially redownloading the same images/javascript files again and again by the client is avoided.

**No.** time synch is not an issue. All times are server-time and not client need not be in the same timezone or time synched with the server as caching is all about relative time – how long to cache.

** **This is a two part question. If only one part is correctly answered, award maximum of 2 marks.**

** **If no mention of 'HTTP Header' fields and generally talking about Caching is storing the data, etc. will deserve only 0 mark credit.**

23. Why is that the UDP header has a length field while the TCP header does not? Is this for any performance benefit ? Illustrate with an example. What about ICMP messages?

**ANSWER :**

It is indeed quite interesting that TCP header does not have a length field. So the question arises how the TCP endpoint know where the incoming segment ends. IP has a Total length field which is 16 bits, it is the total size of the IP packet including the IP header. Let us take an example, incoming frame had a total length field 388 bytes – IP header 20 bytes and TCP header 20 bytes without any header options. From this, it can be derived the TCP data that came in was 388 – 20 – 20 = 348 bytes.

Note, as TCP is a stream based protocol, it is not length based, but as data comes in, it is received/buffered and sent to the upper layer which is the Application layer protocol running on top-of TCP. Well it may be said, not having a redundant field in every message that is sent out is definitely a performance benefit, but not sure if Vint Cerf , et.al and/or DARPA team which worked on the initial draft of TCP had this in mind or some other reason as well.

In the case of UDP though, it is a datagram based protocol and most likely, entire UDP datagrams make enough sense and hold enough information unlike a TCP segment which may be just 20-30 bytes – So UDP datagrams are given to the upper layer –application layer protocols as an entirety – though not mandatory. Yes, here too, the length can be detected given the fact that UDP header is only 8 bytes and has no Optional headers as well. May be one thought is to have symmetricity and help with memory processing to have even-sized headers. Instead of going with 6 bytes, UDP Source Port, UDP Dest Port and UDP Checksum, it might have seen making sense to make the UDP header 8 bytes with the UDP Length for 2 bytes as well.

Interestingly, ICMP does not have a Length field and once again as ICMP runs on top of IP, the size of an ICMP message can be detected from the IP Total Length header as well. Moreover the size of ICMP data is implementation dependent – eg. Ping in windows adds a 32 byte data as ICMP Payload but is not mandatory.