Internet Protocol HW2

P4.

a)
The document being requested is : /cs453/index.html at the Host : gai a.cs.umass.edu .
So the URL of the document being requested by the Browser is :
gai a.cs.umass.edu/cs453/index.html

b)
The version of HTTP being run by the Browser is HTTP 1.1

c)
The browser is requesting a persistent connection as shown by the Connection : “Keep-Alive” Header

d)
The IP address of the host on which the browser is running cannot be seen in this GET request.
However, IP address of the Host on which request is send will be IP address of “gai a.cs.umass.edu”

e)
User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec ko/20040804 Netscape/7.2
The browser that initiates the request is seen in the User Agent field of the GET request.
Different types of browsers have different requirements and specifications on how a page should be displayed. Also, different types of Browsers are required to interpret different pages in a different manner. HTTP just defines the protocols and the headers and the data that is to be sent and received. However, how the webpage should be handled and displayed is determined by the type of Web Browser.

P5.

a)
The server was able to find the document successfully as it can be seen from the HTTP/1.1 200 OK response code.
The document reply was provided on : Date: Tue, 07 Mar 2008 12:39:45GMT

b)
The document was last modified on Last-Modified: Sat, 10 Dec2005 18:27:46 GMT

c)
The number of bytes in the document being returned can be shown by the Content Length field:
Content-Length: 3874

d)
The server did agree to persistent connections as shown by the Keep-Alive Header and also the timeout field that shows that the connection will time out after max=100 seconds.
The first 5 bytes of the document that are returned are :
<!doc

P6.

a)
Having gone through the RFC 2616 regarding the regarding the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed.
Persistent connections have a mechanism using which a client and a server can signal towards each other telling that they want the TCP connection to be closed. This can be done using a connection header that can be sent by any one either the server or the client. Once, a connection close header has been sent by the server to the client, the client should ensure that it does not send any more requests on that particular TCP connection.
Further,
HTTP 1.1 has a connection header that can be defined as close. This ensures that the connection will be closed after a particular response. It also indicates that the connection should not be considered persistent after that particular request or response is complete.

Also, this connection close can be done by any one - either the client or the server or both as well.

b)
There is no encryption mechanism that is provided by HTTP itself.
The users should be very careful to prevent unintentional leakage of the data. There should be a particular interface to prevent the dissemination of the information. Also, people can make use of HTTPS for encrypting the data transfer.

c)
In general, the clients can open any number of connections with the server simultaneously. However, it is mentioned in the RFC that clients should limit the number of simultaneous connections that they maintain to the given server. A client must not have more than 2 connections with any proxy or server.

d)
It is totally possible that the server or client may close a transport connection between then if either one detects the connection has been idle for some time. It is also possible that one side starts closing the connection while the other side is transmitting data via this connection. Let’s say for example there can be a situation when the server sits idle and the client is processing a response. The server will not be aware of the client so it might close the connection.
In this case the client software should open the connection again, and retransmit the aborted sequence of requests without user interaction so long as the request sequence is idempotent.

P7.

The time taken to do the DNS lookup = RTT1 + RTT2 + RTT3 + RTT4 + RTT5 + RTT6 ---- RTTN

Once the DNS lookup is complete the the time taken to form a TCP connection is RTT0 and time taken to fetch the file is RTT0.

So the total time taken is RTT0* 2 + RTT1 + RTT2-------RTTN

P8.
a)
Non persistent with no parallel TCP connections:
RTT1+RTT2+RTT3---RTTN + 2*8*RTT0 + 2RTT0 = RTT1+RTT2+RTT3---RTTN + 18RTT0

This time each of the 8 objects will take 2*RTT0 time each
Time taken to set up connection will be 2*RTT0
Time taken for DNS lookup RTT1+RTT2+RTT3---RTTN

b)
Non persistent HTTP with the browser configured for 5 parallel connections:

Time taken for DNS lookup + Time taken for connection setup + Time taken for 5 concurrent files + Time taken to again setup the connection + Time taken to get 3 files concurrently

Time taken for DNS lookup : RTT1+RTT2+RTT3---RTTN
Time taken for connection setup : 2* RTT0
Time taken to get 5 files concurrently : 2*RTT0
Time taken to get 3 files concurrently : 2*RTT0
So total is, RTT1+RTT2+RTT3---RTTN + 6RTT0

c)
When there will be persistent HTTP connection
Time taken for DNS lookup : RTT1+RTT2+RTT3---RTTN
Time taken to setup connection : 2* RTT0
Time taken to request the object : RTT0 (Please note the connection need not be set up again)
So, total = RTT1+RTT2+RTT3---RTTN + 3RTT0

P18.
a)
When we register a domain name, the CANN (Internet Corporation for assigned names and numbers) requires your domain name registrar to submit your personal contact details to the WHOIS database. Once our details are there in the WHOIS database they become publicly accessible.
So, WHOIS is a protocol that is used for querying the information associated with Domain Names, IP addresses etc.

b)https://www.mydomain.com I used this for finding the details for google.com
https://www.whois.com/whois/ I used this for finding the details for geeksforgeeks.org
My local DNS server IP is in the /etc/resolv.conf file on my MAC operating system

The name server for when i queried google.com is NS1.GOOGLE.COM
The name server when I queried geeksforgeeks.org was NS-1520.AWSDNS-62.ORG

c) I found my local DNS server using the following command:
cat /etc/resolv.conf

nameserver 152.1.14.14
nameserver 152.1.14.12
nameserver 152.1.14.69

Then I performed the nslookup command on “google.com” querying various type of records:
A record, NS record, MX record

Please refer to the below screenshot:

```
Rajshrees-MacBook-Pro:~ rajshreejain$ nslookup
> server
Default server: 152.1.14.14
Address: 152.1.14.14#53
Default server: 152.1.14.12
Address: 152.1.14.12#53
Default server: 152.1.14.69
Address: 152.1.14.69#53
> 152.1.14.14
Server:         152.1.14.14
Address:        152.1.14.14#53

14.14.1.152.in-addr.arpa        name = vrd1.ddi.ncsu.edu.
> set q=A
> google.com
Server:         152.1.14.14
Address:        152.1.14.14#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.164.142
> set q=MX
> google.com
Server:         152.1.14.14
Address:        152.1.14.14#53

Non-authoritative answer:
google.com      mail exchanger = 20 alt1.aspmx.l.google.com.
google.com      mail exchanger = 40 alt3.aspmx.l.google.com.
google.com      mail exchanger = 10 aspmx.l.google.com.
google.com      mail exchanger = 50 alt4.aspmx.l.google.com.
google.com      mail exchanger = 30 alt2.aspmx.l.google.com.

Authoritative answers can be found from:
> set q=NS
> google.com
Server:         152.1.14.14
Address:        152.1.14.14#53

Non-authoritative answer:
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns1.google.com.

Authoritative answers can be found from:
>
```

Then I performed nslookup on “ncsu.edu” and found out the A record, MX record and the NS record.
Please find the screenshot below:

```
Rajshrees-MacBook-Pro:~ rajshreejain$ nslookup
> server
Default server: 152.1.14.14
Address: 152.1.14.14#53
Default server: 152.1.14.12
Address: 152.1.14.12#53
Default server: 152.1.14.69
Address: 152.1.14.69#53
> 152.1.14.14
Server:         152.1.14.14
Address:        152.1.14.14#53

14.14.1.152.in-addr.arpa        name = vrd1.ddi.ncsu.edu.
> set q=A
> ncsu.edu
Server:         152.1.14.14
Address:        152.1.14.14#53

Name:   ncsu.edu
Address: 152.1.27.202
> set q=MX
> ncsu.edu
Server:         152.1.14.14
Address:        152.1.14.14#53

ncsu.edu        mail exchanger = 5 alt2.aspmx.l.google.com.
ncsu.edu        mail exchanger = 10 aspmx2.googlemail.com.
ncsu.edu        mail exchanger = 5 alt1.aspmx.l.google.com.
ncsu.edu        mail exchanger = 10 aspmx3.googlemail.com.
ncsu.edu        mail exchanger = 1 aspmx.l.google.com.
> set q=NS
> ncsu.edu
Server:         152.1.14.14
Address:        152.1.14.14#53

ncsu.edu        nameserver = rdd42.ddi.ncsu.edu.
ncsu.edu        nameserver = rdd31.ddi.ncsu.edu.
ncsu.edu        nameserver = rdd41.ddi.ncsu.edu.
ncsu.edu        nameserver = rdd32.ddi.ncsu.edu.
>
```

d)
It can be seen in the above screen shot that NCSU uses a lot of webservers. These web servers are all used to handle the DNS traffic incoming.

e)
The traffic range used by my university is as:152.1.0.0 - 152.1.255.255

Network: NET-152-1-0-0-1

Source Registry	ARIN
Net Range	152.1.0.0 - 152.1.255.255
CIDR	152.1.0.0/16
Name	NCSU
Handle	NET-152-1-0-0-1
Parent	NET-152-0-0-0-0
Net Type	DIRECT ASSIGNMENT
Origin AS	*not provided*
Registration	Fri, 07 Jun 1991 03:00:00 GMT (Thu Jun 06 1991 local time)
Last Changed	Wed, 02 Sep 1998 12:46:05 GMT (Wed Sep 02 1998 local time)
Self	https://rdap.arin.net/registry/ip/152.1.0.0
Alternate	https://whois.arin.net/rest/net/NET-152-1-0-0-1
Port 43 Whois	whois.arin.net

f) whois and nslookup expose a lot of information to the internet.
The attacker would be able to determine the details of the registrant the domain and also some information like contact etc.
They can also provide false authentication for the institution's verification system. By masquerading as the end point for an institution, they can introduce malware and virus into the institution's servers, compromising its digital integrity since information is power.

g)They help in determining the registrant and the domain servers and the information about the domain.
It helps in the verification of the owner of the domain and to ensure that the domain is authentic.

P22.
Consider distributing a file of 15 Gbits into N peers.
Server upload rate of u_s = 30 Mbps
Each peer download rate of d_i=2Mbps
Upload rate of each peer is u
N = 10, 100, 1000
u=300 Kbps, 700Kbps, 2Mbps

Client-Server Architecture

The minimum distribution time for client-server distribution, the formula used will be: Dcs = max {N*F/u_s, F/dmin}

dmin=d_i=2Mbps

u_s=30Mbps
F=15Gbits=15 Gbits * 1024 = 15,360 Mbits

300 Kbps = 300/1000 Mbps
700 Kbps = 700/1000 Mbps
2 Mbps

N*F/u_s=10*15360/30 for N=10
N*F/u_s=100*15360/30 for N=100
N*F/u_s=1000*15360/30 for N=1000
F/dmin=15360/2=7680

		N	N	N
		10	100	1000
u	300 Kbps	7680	51200	512000
u	700 Kbps	7680	51200	512000
u	2 Mbps	7680	51200	512000

Peer to Peer Architecture

The minimum distribution time for peer to peer distribution, the formula used will be:
D max{F/u_s , F/d_min , NF/(u_s+∑u_i)} ---> u_i is the upload speed for all the peers

When u_i = 300/1000 Mbps
u_s+∑u_i = 30 + 10 * 300/1000, N=10 = 33 Mbps | NF= 10*15360 | 4654.54 seconds
u_s+∑u_i = 30 + 100 * 300/1000, N=100 = 60 Mbps | NF= 100*15360 | 25600 seconds
u_s+∑u_i = 30 + 10000 * 300/1000, N=1000 = 330 Mbps | NF= 1000*15360 | 46545.45 seconds

When u_i = 700/1000 Mbps
u_s+∑u_i = 30 + 10 * 700/1000, N=10 = 37 Mbps | NF= 10*15360 | 4251.35 Seconds
u_s+∑u_i = 30 + 100 * 700/1000, N=100 = 100 Mbps | NF= 100*15360 | 15360 Seconds
u_s+∑u_i = 30 + 1000 * 700/1000, N=1000 = 730 Mbps | NF= 1000*15360 | 21041.09 Seconds

When u_i = 2 Mbps
u_s+∑u_i = 30 + 10 * 2, N=10 = 50 Mbps | NF= 10*15360 | 3072 Seconds
u_s+∑u_i = 30 + 100 * 2, N=100 = 230 Mbps | NF= 100*15360| 6678.26 Seconds
u_s+∑u_i = 30 + 1000 * 2, N=1000 = 2030 Mbps | NF= 1000*15360 | 7566 Seconds

F/u_s = 15,360/30 = 512 seconds
F/dmin = 15360/2 = 7680 seconds
NF/(u_s+∑u_i)

N*F =

		N	N	N
		10	100	1000
u	300 Kbps	7680	25600	46545.45
u	700 Kbps	7680	15360	21041.09
u	2 Mbps	7680	7680	7680

P23.
A file of F bits into N peers
a)
We are saying that u_s/N < d_min
Consider that the server is sending the file to each client at a speed of u_s/N
So the time should be calculated according to the speed of u_s/N.
Now the time that will be taken by a client for F bits is
F/(u_s/N) = F*N/u_s
Hence this should be the overall time as well.

b)
In this part, we say that u_s/N > dmin
In this case, since dmin is less then we can consider the server to send the file at a speed of dmin to each of the clients.
The total N*dmin will be less than u_s according to u_s/N > dmin
The time taken by the clients to get the data F bits will be F/dmin
Since all the clients will have an overall time F/dmin

c)
We can see in part a of this question that when u_s/N < d_min then overall time is F*N/u_s
Also, we can see in part b of this question when u_s/N > d_min then overall time is F/d_min
These are the only 2 cases that can be considered while calculating the time for the overall distribution of the file.
If we combine both the parts a and b we get the simple equation :
Dcs = max {N*F/u_s, F/dmin}

26.
Bob Joins BitTorrent but he does not want to upload any data to his peers. He just wants to receive the file.
a)
Bob claims that he can receive a complete copy of the file that is shared by the swarm.
Yes, this is possible.
Ideally, the peers choose to send data to peers who are actively sending them data at the highest rate. This is how the particular peer unchocks the peers. Apart from this the sender of file also randomly chooses a

particular neighbor and sends it data. This is called optimistically unchocking.
So, if at all the peers stay for a long time in the network, they can optimistically unchock the particular peer and send them chunks.
This is how even bob can receive the full file, without sharing even a single chunk by itself.

b)
Bob also claims that he can make this free-riding more efficient. He wants to make use of the multiple computers that are present in his lab in the computer department.
Yes, he can achieve that as well.
He can parallelly run a BitTorrent client on all of his lab computers and request the same file. He can ask for particular parts of the file on separate computers and run them parallelly or even schedule them to run one by one. All these parts of the file will be received by optimistic unchocking. Once all the parts are received on all the computers, he can manually merge and combine the file.

Wireshark Lab is done in a separate file

Answers to the Research Paper Questions:
Paper 1
Internet Indirection Infrastructure

1
Briefly describe the rendezvous based model of i3, its communication primitives, and ID stack generalization.
Ans.
Rendezvous based model of i3
-The purpose of the I3 model is to make the act of sending and receiving separate.
-This model is called the rendezvous based model of i3, the packets are pairs where (id, data) in which the id is an m bit identifier and payload is the data that is to be sent.
-The receivers make use of trigger points to show their interest in the packets and receive them
-There is (id, addr) where id represents the trigger identifier and the addr represents the node address consisting of the address and the port.
-In case we talk about the communication in the sender and the receiver, where R wants to receive packets sent to id. The receiver inserts a trigger (id, R) in the network. This way the (id, data) packet is sent to the receiver R.
-Hence both the sender and receiver are unaware of each other
-Further the point noting is an identifier id_t matched identifier id iff and only iff id_t is the longest prefix match and the longest prefix match is as long as at least K. K can be chosen such that 2 randomly chosen identifiers match is negligible

Communication primitives of I3.
I3 can be used by applications for purposes like mobility, unicast and anycast.
Mobility - The host maintains mobility when a host is assigned a new address when it moves from one location to another. The host has to change the address from and to as a result of changing subnets so that end to end connectivity is preserved by just updating the triggers.
In I3 to keep up the efficiency 2 major steps are taken -
*The address of the server storing the trigger is cached at the sender and subsequent packets are directly sent to the server via IP addresses

*end hosts use off-line heuristics to choose triggers that are stored at the i3 servers close to them

Multicast -

Creating a multicast group is equivalent to having all members of the group register triggers with the same identifier . As a result, any packet that matches is forwarded to all members of the group .

Anycast -

Anycast ensures that a packet is delivered to exactly one receiver in a group, if any. Anycast enables server selection, a basic building block for many of today's applications. To achieve this with , all hosts in an anycast group maintain triggers which are identical in the most significant bits. These bits play the role of the anycast group identifier.

Stack Generalization

In this generalization of the model, the identifiers are replaced by stack identifiers. The stack identifiers are of form : (id_1,id_2,id_3,--id_k**)** where id_i is an identifier or an address.

The packets and triggers take the from :

Packet p = (id_stack, data)

Trigger t = (id, id_stack)

The generalized form of packets allows a source to send a packet to a series of identifiers, much as in source routing. The general- ized form of triggers allows a trigger to send a packet to another identifier rather than to an address.

The code for sending and receiving the packets by I3 server is written accordingly.

2.

What applications of i3 are described in the paper? Can you think of other applications that could be implemented using i3?

Ans.

The applications of I3 described in the paper are:

Service Composition

The stack identifier functionality of I3 can be used to help in the applications where the applications might require the third parties to process the data before it reaches the destination. This is done when the data needs to be transformed by a number of third party servers before it reaches the destination.

The sender associates with each data packet the stack (id_html-wml,id) , where id represents the flow identifier. As a result, the data packet is routed first to the server which performs the transcoding. Next, the server inserts packet (id, data) into i3, which delivers it to the receiver.

Heterogeneous Multicast

Let's say we need a video to be played in MPEG by one receiver and H.263 by the other receiver.

To provide this functionality, we use the ability of the *receiver*, instead of the sender.

Using this functionality receivers with different display capabilities can subscribe to the same multicast group.

Server Selection

Use of the last bits of the identifiers to encode application preferences for server selection.

Let us say our goal is to balance the requests in the servers.

This goal should be served by setting the least significant bits of both the trigger and packet identifiers to random values

Let us say there is a goal of selecting a server that is close to the client in terms of latency.

The goal should be served if each server can use the last bits of its trigger identifiers to encode its location, and the client can use the last bits in the packets' identifier to encode its own location.

Large Scale Multicast
In case of multicast, it is assumed that all the members of the multicast group insert triggers with the same identifiers. And now since the triggers with identical identifiers are stored at the same server the server should be responsible for performing multicasting to all the members of the group. However, this solution is not viable for large scale groups so we look into an approach to address this issue by building hierarchy of triggers.

Some other applications that can be built using the I3 architecture are as follows:
Let us say there is a live telecast of some big event we can use the I3 model in that case helping in the multicast of the packets. Because in this case, the senders and the receivers are completely decoupled. The receivers can accept the packets based on their requirements. The receivers can have triggers to accept the packets based on their requirements.

Another application that it can be used in is the Peer to peer communication model. Where the peers communicate with each other based on their requirements. The triggers and identifiers can be made and adjusted in such a way that the I3 model works as a perfect communication channel.

3.
What is the difference between public and private triggers? How can the latter be used to improve routing and security?
The differentiation in the public and private triggers is made at the application level. I3 really does not differentiate between public and private triggers.

*Public triggers allow in communication between 2 end-hosts. In case of public triggers the identifier is known to all the end hosts in the system.
*An example that can be considered for the same is a Web Server which has a public trigger and this can allow any client to contact it.
*Public triggers are long lived and can be existent for like months or days
*Further now coming to private triggers, these triggers are used by a relatively smaller number of clients.
*Private triggers are short lived unlike the public triggers and these private triggers just exist during the duration of the flow.
*As soon as the communication ends and the private trigger is used by the server and the client, the private trigger is destroyed.

It is mentioned that the Private triggers can be used to increase the Routing efficiency and also increase the Security.

Routing Efficiency
In order to efficiently manage routing in the I3 model. The IP address of the I3 server is cached at the sender. However, While caching the server storing the receiver's trigger reduces the number of hops, we still need to deal with the triangle rout- ing problem. That is, if the sender and the receiver are close by, but the server storing the trigger is far away, the routing can be in- efficient. Solution to this issue is that receivers choose

their private triggers such that they are located on the nearby servers. That would ensure that the packets would not take a long detour before reaching their destination.

Security

While we introduce flexibility for the users, we might also create new opportunities for the malicious users.

-Eavesdropping

With , a public trigger is known by all users in the system, and thus anyone can eavesdrop the traffic to such a trigger. To alleviate this problem, end-hosts can use the public triggers to choose a pair of private triggers, and then use these private triggers to exchange the actual data. To keep the private triggers secret, one can use public key cryptography to exchange the private triggers.

-Trigger hijacking

One possibility to guard against this attack is to add another level of indirection.

-DoS Attacks

To alleviate these attacks, uses three techniques:

- -Challenges
- -Resource allocation
- -Loop detection:

Paper 2
The Akamai Network: A Platform for High-Performance Internet Applications

1.
What are the main challenges in delivering content directly over the Internet?
Ans.
There are a number of factors that make inter-network data communication not reliable and inefficient. The major factors are:

*__Peer point congestio__n-There are points where the networks are forced to cooperate with competing entities. These peering points become the cause of packet loss and latency.
*I__nefficient Routing Protocols__ - The protocol BGP that is used, was never designed for performance. BGP bases its route calculations primarily on AS hop count but it does not know anything about topologies, latencies, or real-time congestion of the underlying networks. Also, it is well known that BGP is vulnerable to human error as well as foul play; misconfigured or hijacked routes can quickly propagate throughout the Internet, causing route flapping, bloated paths, and even broad connectivity outages.
*__Unreliable networks-__ There can be several times when outages can happen due to a variety of reasons. Some of the reasons are cable cuts, misconfigured routers, DDoS attacks, power outages, even earthquakes and other natural disasters.
*__Inefficient communications protocols-__ TCP was designed for a reliable transfer of packets and prevent congestion in the network. Some other major issues related with TCP are as follows:
- significant overhead and can have suboptimal performance for links with high latency or packet loss
-Middle mile congestion exacerbates the problem, as packet loss triggers TCP retransmissions, further slowing down communications
-for interactive applications, the multiple round trips required for HTTP requests can quickly add up, affecting application performance
-Web browser also limits the number of parallel connections they make for a given host name, further limiting performance over long distances for sites that consist of many objects.
-TCP also becomes a serious performance bottleneck for video and other large files.
*__Scalability-__Scaling is required when web applications have a lot of traffic incoming. It is difficult to plan the number of resources to be allocated. Scaling and mirroring origin infrastructure is costly and time-consuming, and it is difficult to predict capacity needs in advance. There can be instances where people might over provision leading to extra expenditure and wastage of money. Or underprovisioning leading to failure of the web application. Further, it also needs to be ensured that the scalability is end to end with adequate bandwidth at all the points.
*__Application limitations and slow rate of change adoption__ - Although there might be some problems based on the protocols networks etc. But there is dependency on application also.

2.
What are the main components of Akamai's delivery network?
Ans.
The main components of the Akamai Delivery Network are the following:
-Mapping System
-Edge Server
These come into the picture when the user types the URL in the browser. The domain name of the URL is translated by the **mapping system** into the IP address of an **edge server** to serve the content. This helps to assign the user to a server, the mapping system bases its answers on large amounts of historical and current data that have been collected and processed regarding global network and server conditions. This data is used to choose an edge server that is located close to the end-user.

- Edge server platform
It is a large global deployment of servers located in thousands of sites around the world. These servers are responsible for processing requests from nearby users and serving the requested content.

-Origin server
-Transport system

The dynamic content on the Web Pages can be very customized according to the particular user and it cannot be saved in the cache. Hence, it is to be fetched from the **Origin Server.** The **Transport System** is used to transfer this data from the origin in a reliable and efficient manner over the long-haul Internet.

-Communications and Control System
It is used to spread system information, control messages, status information, and if there is any fault in the configuration and similar updates.

-Data Collection and Analysis System
Collects and processes various information like Server Logs, Client Logs, Server Information. This collected data can be used for monitoring, alerting, analytics, reporting, and billing etc.

-Management Portal
*Configuration level platform that provides an Enterprise level customer fine control on how content and applications are served to the users.
*Visibility to the enterprise customers through the Management Portal in showing how the clients and users are interacting with the applications and content.

3.
What are the main architectural solutions for streaming and content delivery?
Architectural Solutions for streaming and content delivery -

* In order to help performance, reliability and scalability for content and stream delivery it can be attempted to reduce the long haul communication through the middle mile bottleneck of Interest. This was made possible by making the servers really close to the users.

* The paper points to Akamai's approach for increasing the performance of the streaming and content delivery.

Akamai's approach is to reach out to the true edge of the Internet, deploying not only Tier1 and Tier 2 data centers but also in large numbers of end user ISPs. Akamai has deployed several clusters in various locations instead of massive servers in one location.

* Being highly distributed also increases platform availability, as an outage across an entire data center (or even multiple data centers) does not need to affect delivery network performance.

* Peer to Peer technologies provide a very good solution for serving static Web Content. But there is some lack of implementation and functionality in the current P2P algorithms. However, Akamai provides a hybrid option for Client - side delivery. Akamai ensures the freshness of the data being served to the users, how different content is handled, ability to handle real time traffic reports and analytics.

4.
How does application delivery differ from content delivery? What are the main techniques Akamai uses to optimize the performance of applications?

Content delivery networks are networks with data centers all over the globe. These are used to deliver Internet content such as web pages, graphics, media files, software downloads, and streaming video in an efficient manner. Content delivery network providers create these private networks, optimize them for

content delivery, and sell their services to customers who want a fast, efficient means of delivering content to end users.

An application delivery network is similar to one designed to deliver content. However, with an application delivery network, the network is optimized to handle the demands of interactive applications.

Perhaps the best way to understand the difference between content and application delivery networks is to understand the difference between static content and dynamic applications. Static content, such as a music download, remains the same. Users around the world can download that music file. With a content delivery network, copies of that file are hosted around the world and the closest server is used to deliver it to a requesting user. The benefits of using a content delivery network are numerous with one of the biggest benefits being an increase in delivery speeds and reliability.

Dynamic content, such as application-related content, changes constantly. An Office 365 user might open and edit a Word document, creating a transaction that is completely unique from that of other users. While an application delivery network also has servers around the world, it creates a network environment where dynamic content can flow back and forth in the most efficient manner. The benefits of using an application delivery network are also numerous with application speed, performance, and reliability improvements being among the top benefits.

Source for this answer also from the website :
https://mlytics.com/blog/whats-the-difference-between-adn-and-cdn/

Main techniques used by Akamai to optimize the performance of Applications:
Approach 1 :
To speed up long haul Internet Connections by using Akamai platform as a high performance overlay network
>Transport system for Akamai‘s application delivery network relies on the use of Akamai‘s highly distributed edge servers as a high-performance overlay network that makes wide-area Internet communications faster and more reliable.
>Communications between any two Akamai servers can be optimized to overcome the inefficiencies we discussed in Section 3 through a number of techniques including path optimization and protocol enhancements.
> This transport system is applicable to many types of situations: accelerating non-cacheable customer content and applications, retrieving content (or performing freshness checks)
>Additional application logic can also be implemented by edge servers, such as authentication or serving different versions of a page based on attributes of the client.

Approach 2:
Push application logic from the origin server to the edge of the Internet.
>Akamai introduced such capabilities on its platform nearly a decade ago with its EdgeComputingTM services, which include the capability for companies to deploy and execute request-driven Java J2EE applications or application components onto Akamai‘s edge servers.
>Akamai EdgeComputing takes cloud computing to a level where application resources are allocated not only on-demand but also near the end user.

5.
What are the main implications of the discussion in Section 6 on Cloud computing?
Akamai has built on Cloud Computing itself in such a way that it has a resilient architecture and also allows edge computing. It allows the companies to deploy and execute request driven Java J2EE applications or application components on Akamai Edge servers.
The users and enterprise companies can take application resources not only on demand but also near the end user.
The allowance of cloud computing by edge to allow the architecture to be hybrid is also beneficial. For example, there are applications that might be transactional. Hence they might have to communicate with the origin servers a lot. Some example of these are : Content aggregation/transformation, Static databases, Data collection, Complex applications.

6.
Describe the main parts of Akamai’s mapping system.
Ans.
The main parts of the Akamai mapping system are :
Scoring System
Real-time mapping
Scoring System
-Creates a current, topological map capturing the state of connectivity across the entire Internet.
-Map divides the Internet into equivalence classes of IP addresses and represents how (and how well) they connect to each other. This requires collecting and processing tremendous amounts of historic and real-time data—including pings, traceroutes, BGP data, logs, and IP data, collected cumulatively over the years and refreshed on a continual basis.
-Network latency, loss, and connectivity are monitored at a high frequency, enabling immediate response to Internet faults and changes in performance

Real-time mapping
-Creates the actual maps used by the Akamai platform to direct end users (identified by the IP addresses of end users and their name servers) to the best Akamai edge servers to respond to their requests
-Intermediates for tiered distribution and the overlay network

\