ComputerNetworks

Homework 2

Student Name : ABID ALI
Student ID : 2019380141
Student Email : abiduu354@gmail.com / **3616990795@qq.com**
Lecture : Professor ZHANG Ying
Submission : 10/07/2022
Submitted Email : 1282208386@qq.com

1. What is the difference between **TCP** and **UDP services**?

Solution:

TCP

Transmission Control Protocol (TCP) is connection-oriented, meaning once a connection has been established, data can be transmitted in two directions. TCP has built-in systems to check for errors and to guarantee data will be delivered in the order it was sent, making it the perfect protocol for transferring information like still images, data files, and web pages.

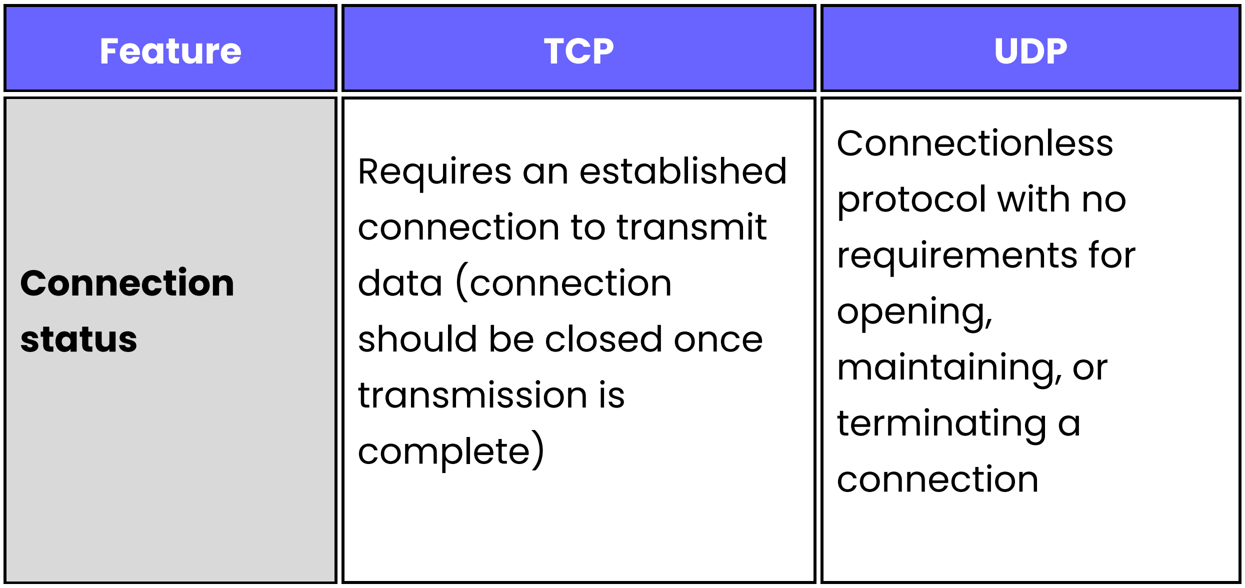
But while TCP is instinctively reliable, its feedback mechanisms also result in a larger overhead, translating to greater use of the available bandwidth on your network.

UDP

User Datagram Protocol (UDP) is a simpler, connectionless Internet protocol wherein error-checking and recovery services are not required. With UDP, there is no overhead for opening a connection, maintaining a connection, or terminating a connection; data is continuously sent to the recipient, whether or not they receive it.

Although UDP isn't ideal for sending an email, viewing a webpage, or downloading a file, it is largely preferred for real-time communications like broadcast or multitask network transmission.

The difference between TCP and UDP



Feature	TCP	UDP
Connection status	Requires an established connection to transmit data (connection should be closed once transmission is complete)	Connectionless protocol with no requirements for opening, maintaining, or terminating a connection

Data sequencing	Able to sequence	Unable to sequence
Guaranteed delivery	Can guarantee delivery of data to the destination router	Cannot guarantee delivery of data to the destination
Retransmission of data	Retransmission of lost packets is possible	No retransmission of lost packets
Error checking	Extensive error checking and acknowledgment of data	Basic error checking mechanism using checksums
Method of transfer	Data is read as a byte stream; messages are transmitted to segment boundaries	UDP packets with defined boundaries; sent individually and checked for integrity on arrival
Speed	Slower than UDP	Faster than TCP
Broadcasting	Does not support Broadcasting	Does support Broadcasting

Optimal use	Used by HTTPS, HTTP, SMTP, POP, FTP, etc	Video conferencing, streaming, DNS, VoIP, etc

2. Please write some **HTTP response status codes**.

Solution:

An HTTP status code is a server response to a browser's request. When you visit a website, your browser sends a request to the site's server, and the server then responds to the browser's request with a three-digit code: the HTTP status code.

These status codes are the Internet equivalent of a conversation between your browser and the server. They communicate whether things between the two are A-okay, touch-and-go, or whether something is wrong. Understanding status codes and how to use them will help you to diagnose site errors quickly to minimize downtime on your site. You can even use some of these status codes to help search engines and people access your site; a 301 redirect, for example, will tell bots and people that a page that has moved somewhere else permanently.

The first digit of each three-digit status code begins with one of five numbers, 1 through 5; you may see this expressed as 1xx or 5xx to indicate status codes in that range. Each of those ranges encompasses a different class of server response.

Common HTTP status code classes:

1xxs – Informational responses: The server is thinking through the request.

2xxs – Success! The request was successfully completed and the server gave the browser the expected response.

3xxs – Redirection: You got redirected somewhere else. The request was received, but there's a redirect of some kind.

4xxs – Client errors: Page not found. The site or page couldn't be reached. (The request was made, but the page isn't valid — this is an error on the website's side of the conversation and often appears when a page doesn't exist on the site.)

5xxs – Server errors: Failure. A valid request was made by the client but the server failed to complete the request.

The most important status codes for SEOs

It's important for every professional SEO and website owner to understand the status codes that have the biggest impact on SEO.

Imagine you're working on a site that's showing a lot of 5xx errors; you'll want to know off the top of your head that this is a server issue. 4xx errors affect visitor experience, so right away you can start thinking about any changes you may have made to your URLs, or whether you've any deleted pages. Once you understand the cause of the issue, you can look at implementing a custom 404 page, or look into using the all-powerful 301 redirect to send visitors to the right place.

It's worth learning — and committing to memory — the most impactful status codes every SEO should know:

HTTP Status Code 200 - OK

This is your ideal status code for your normal, everyday, properly functioning page. Visitors, bots, and link equity pass through linked pages like a dream. You don't need to do anything and you can happily go about your day secure in the knowledge that everything is just as it should be.

HTTP Status Code 301 - Permanent Redirect

A 301 redirect should be utilized any time one URL needs to be redirected to another *permanently*. A 301 redirect means that visitors and bots that land on that page will be passed to the new URL. In addition, link equity — the power transmitted by all those hard-earned links to your content — is also passed to the new URL through a 301 redirect. Despite talk from Google that all 3xx redirects are treated equally, tests have shown this is not completely true. A 301 redirect remains the preferred method of choice for permanent page redirects.

HTTP Status Code 302 - Temporary Redirect

A 302 redirect is similar to a 301 in that visitors and bots are passed to the new page, but link equity may not be passed along. We do not recommend using 302 redirects for permanent changes. Using 302s will cause search engine crawlers to treat the redirect as temporary, meaning that it may not pass along the link equity that the magical 301 does.

HTTP Status Code 404 - Not Found

This means the file or page that the browser is requesting wasn’t found by the server. 404s don’t indicate whether the missing page or resource is missing permanently or only temporarily. You can see what this looks like on your site by typing in a URL that doesn't exist. It’s like hitting a brick wall. Just as you’ve experienced, your visitors will hit a page that has a 404 error and either try again (if you’re lucky) or wander away to another site that has the information they’re seeking.

Every site will have some pages that return 404 status codes. These pages don’t always have to be redirected; there are other options. One common misconception is that it’s an SEO best practice to simply 301 redirect pages that return a 404 status code to the homepage of the given domain. This is actually a bad idea for the majority of cases, because it can confuse users who may not realize that the webpage they were trying to access doesn't exist.

If the pages returning 404 codes are high-authority pages with lots of traffic or have an obvious URL that visitors or links are intended to reach, you should employ 301 redirects to the most relevant page possible. For example, if your page on sugar-free cupcakes no longer exists, you may want to redirect this URL with a 301 to your sugar-free recipe category page.

Outside of these instances, it may be necessary for a URL return a 404 on purpose — this will keep them from getting indexed and repeatedly crawled by search engines. Give your visitors the best experience possible with a custom 404 page, as suggested by this Google Search Console guide. For example, e-commerce sites often produce 404 pages when products go out of stock, so these sites are great candidates for creating a custom e-commerce 404 page.

HTTP Status Code 410 - Gone

A 410 is more permanent than a 404; it means that the page is gone. The page is no longer available from the server and no forwarding address has been set up. Any links you have on your site that are pointing to a 410 page are sending bots and visitors to a dead resource, so if you see them, remove any references or links to them from your content.

HTTP Status Code 500 - Internal Server Error

Instead of the problem being with pages missing or not found, this status code indicates a problem with the server. A 500 is a classic server error and will affect access to your site. Human visitors and bots alike will be lost, and your link equity will go nowhere fast. Search engines prefer sites that are well maintained, so you'll want to investigate these status codes and get these fixed as soon as you encounter them.

HTTP Status Code 503 - Service Unavailable

Another variety of the 500, a 503 response means that the server is unavailable. Everyone (human or otherwise) is asked to come back later. This could be due to temporarily overloading the server or maintenance of the server. A 503 status code ensures that the search engines know to come back soon because the page or site is only going to be down for a short time.

3. How cookies work and what **cookies** can be used for?

Solution:

Cookies may be delicious as baked goods but what about the other kind of cookie? Digital cookies are an essential part of any internet browsing experience. A computer cookie may also be called an HTTP cookie, web cookie, internet cookie, and browser cookie, but they're all the referring to the same thing: a way to track your website activity.

When you visit a website, your web server transfers a small packet of data to your device's browser: a computer cookie.

This cookie is designed to remember information about you, including a record of your website visits and activity. It's important to understand how they work and when you need to step in to manage how your browser stores these packets of information.

How Do Cookies Work?

Computer cookies are small files, often including unique identifiers that web servers send to browsers. These cookies then can be sent back to the server each time your browser requests a new page. It's a way for a website to remember you, your preferences, and your habits online.

- **Suggested products or content**: If you're on a shopping site that has a "related searches" feature, you'll see results based on your cookies that are cross-referenced with other users who have similar shopping habits and preferences.

- **Customization**: Cookies store valuable information like your location and currency preference so you don't have to select these options every time you enter a site or start a session.
- **Form submission**: If you've ever had to repeatedly fill out your information on a web form, you know how annoying it can be to type in your entire mailing and billing address over and over again. Cookies can save this information so it auto-fills the next time you're on the site.
- **Online shopping**: When you place a product in your online shopping cart but leave the site, you can typically come back to the cart and find everything still there. That's thanks to cookies.

What Are Cookies Used For?

Websites use HTTP cookies to streamline your web experiences. Without cookies, you'd have to login again after you leave a site or rebuild your shopping cart if you accidentally close the page. Making cookies an important a part of the internet experience.

Based on this, you'll want to understand why they're worth keeping — and when they're not.

Here's how cookie are intended to be used:

1. **Session management.** For example, cookies let websites recognize users and recall their individual login information and preferences, such as sports news versus politics.
2. **Personalization.** Customized advertising is the main way cookies are used to personalize your sessions. You may view certain items or parts of a site, and cookies use this data to help build targeted ads that you might enjoy.
3. **Tracking.** Shopping sites use cookies to track items users previously viewed, allowing the sites to suggest other goods they might like and keep items in shopping carts while they continue shopping.

While this is mostly for your benefit, web developers get a lot out of this set-up as well.

Cookies are stored on your device locally to free up storage space on a website's servers. In turn, websites can personalize while saving money on server maintenance and storage costs.

References:

1) https://www.hp.com/us-en/shop/tech-takes/what-are-computer-cookies
2) https://www.kaspersky.com/resource-center/definitions/cookies

4. Please describe an example of **DNS name resolution**, you can draw the process and write how it works.

Solution:

DNS name resolution

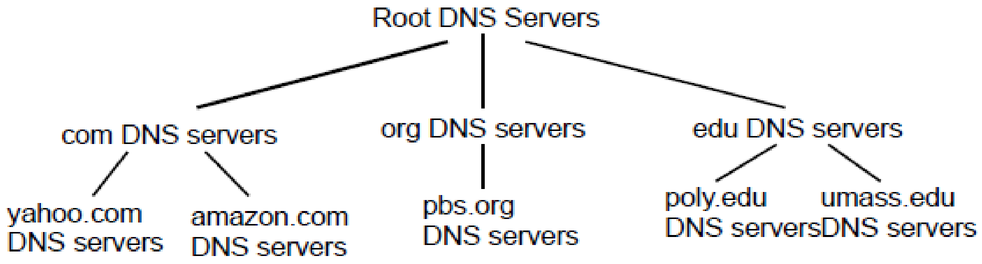
The **Domain Name Systems** (DNS) is the phonebook of the Internet. Humans access information online through domain names, like **quora.com** or **google.com.** Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

How does DNS work?

The process of DNS resolution involves converting a **hostname** (**such as example.com**) into a computer-friendly **IP address** (**such as 192.168.1.1)**. An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - *like a street address is used to find a particular home.*

When a user wants to load a webpage, a translation must occur between what a user types into their web browser (**example.com**) and the machine-friendly address necessary to locate the **example.com** webpage.

DNS: a distributed, hierarchical database



client wants IP for www.amazon.com;

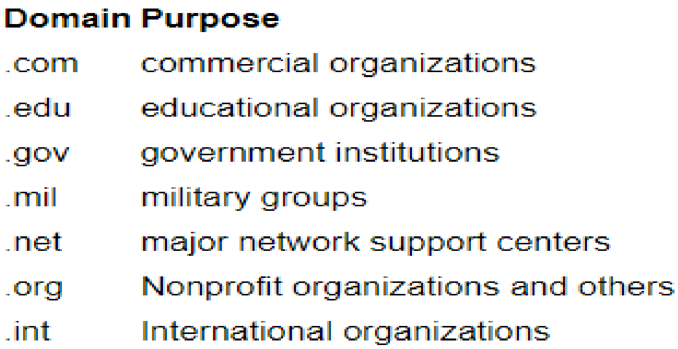
- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Local DNS name server

- Does not strictly belong to hierarchy

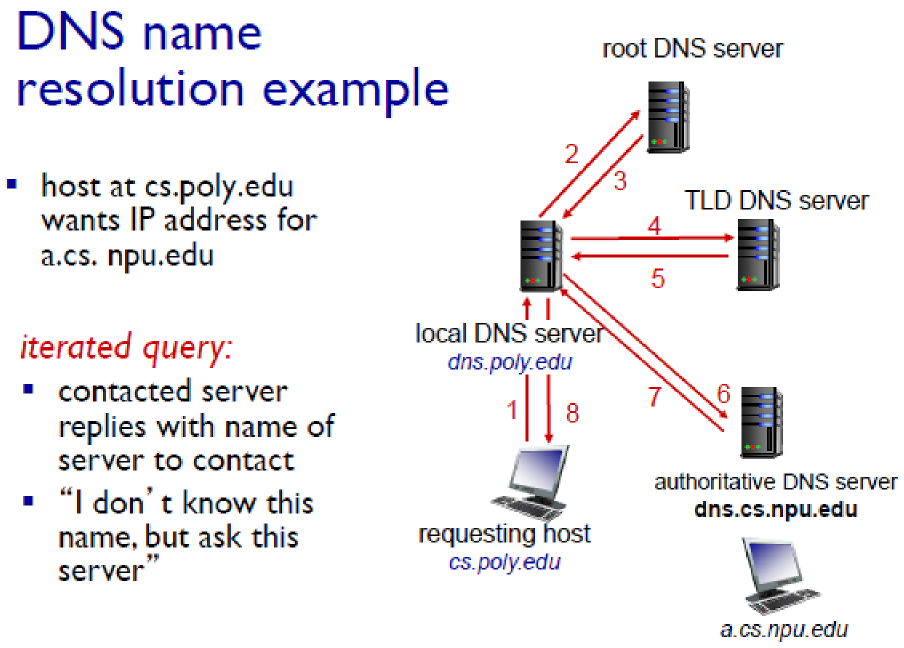
- Each ISP (residential ISP, company, university) has one.
    1. Also called “default name server”
- When host makes DNS query, query is sent to its local DNS server
    1. Has local cache of recent name-to-address translation pairs (but may be out of date!)
    2. Acts as proxy, forwards query into hierarchy

Organizational Hierarchy



Domain	Purpose
.com	commercial organizations
.edu	educational organizations
.gov	government institutions
.mil	military groups
.net	major network support centers
.org	Nonprofit organizations and others
.int	International organizations

DNS name resolution, follow the given below process and working procedure.

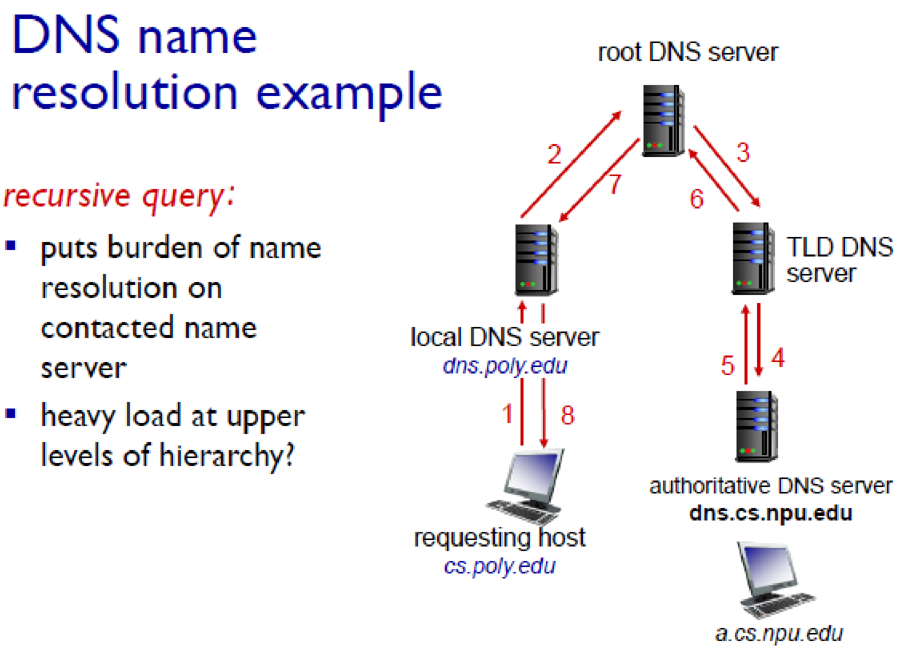


DNS name resolution example

- host at cs.poly.edu wants IP address for a.cs. npu.edu

iterated query:

- contacted server replies with name of server to contact
- “I don’ t know this name, but ask this server”



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?

5. Please try to identify the categories of the following **DNS records** (Based on 2-13 pages of **7-Chapter2**):

（1） （foo.com, dns.foo.com）

（2）(relay1.bar.foo.com, 145.37.93.126)

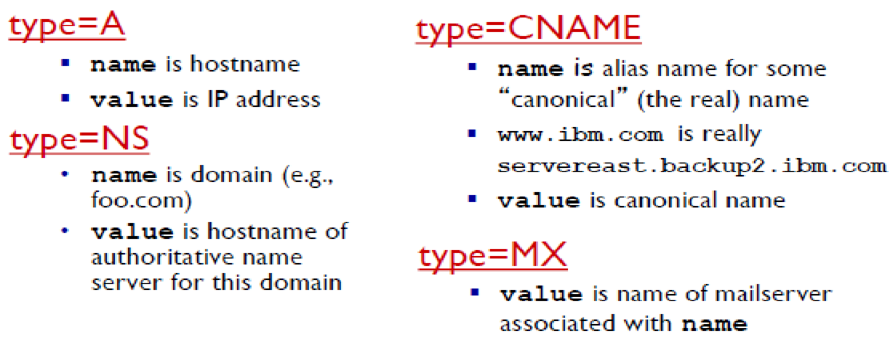
（3） （foo.com, relay1.bar.foo.com）

（4） （umass.edu， dns.umass.edu）

Solution:

DNS: Distributed database storing resource records (RR)

RR format:**(name, value, type, ttl)**



type=A

- **name** is hostname
- **value** is IP address

type=NS

- **name** is domain (e.g., foo.com)
- **value** is hostname of authoritative name server for this domain

type=CNAME

- **name** is alias name for some "canonical" (the real) name
- `www.ibm.com` is really `servereast.backup2.ibm.com`
- **value** is canonical name

type=MX

- **value** is name of mailserver associated with **name**

Type=A; host name to host IP

（2）(relay1.bar.foo.com, 145.37.93.126)

Type=NS; domain to an authoritative name server for domain

（1）　（foo.com, dns.foo.com）

Type=CNAME; provides canonical hostname for alias hostname

（3）　（foo.com, relay1.bar.foo.com）

Type=MX; canonical name of a mailserver
（4）　（umass.edu， dns.umass.edu）

Reference: http://www2.ic.uff.br/~michael/kr1999/2-application/2_050-dns.htm

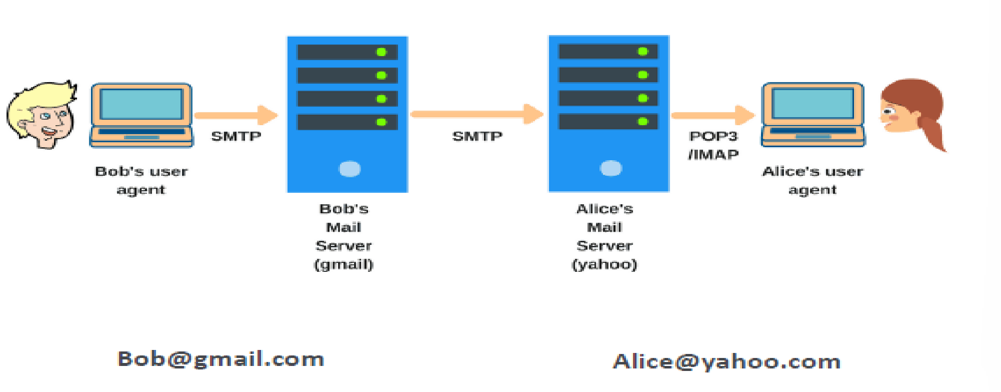
6. How **email** works? And what do the **user agent and mail servers** do?

Solution:

Electronic mail (email or e-mail)

- Method of exchanging messages ("mail") between people using electronic devices.
- synchronous
- Inexpensive
- Rich contents

How email works –example



Major components:

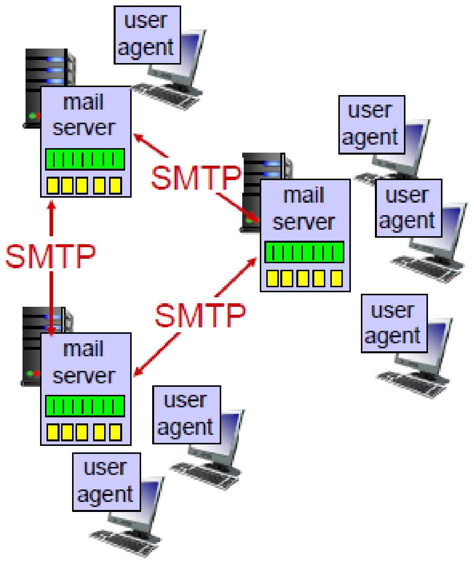
- User agents
- Mail servers
- Simple mail transfer protocol: SMTP

User Agent

- a.k.a.“mail reader”
- composing, editing, reading mail messages.
- e.g., Outlook, iPhone mail client.
- outgoing, incoming messages stored on server.

Mail servers:

- **Mailbox** contains incoming messages for user.
- **Message queue** of outgoing (to be sent) mail messages.



A **mail user agent** is just an email client. Examples of mail user agents include Microsoft Outlook and Mozilla Thunderbird. **Webmail** (web interface to email) is another type of email client where you access email stored on the server without downloading it locally.

A **mail server** is an always-on service software running on a server. It listens & responds to email access requests from email clients via POP3/IMAP, and it sends out emails received from email clients via SMTP relay mechanism. Examples of mail servers include Microsoft Exchange, Apache James, etc.

Gmail from a UI perspective is just a Webmail client. However, Gmail as a service is a combination of Webmail and mail server.