Name Ni, Tianshu Date: 2023.10.5

(last name, first name)

NYU ID: N16371283

Course Section: 001

**Assignment 2**

**Total in points** (100 points total):

**Professor’s Comments:**

**Affirmation of my Independent Effort:** Tianshu Ni

**(Sign here)**

* 1. Problem 1 – Application Level Protocols:

Design and describe an application level protocol for a soda vending machine?

Draw the operation of your protocol similar to Figure 1.2 from course textbook shown below.

A book with a diagram of a network

Description automatically generated with medium confidence

**Soda Vending Machine Protocol**

1. **User Selection**: The user selects a soda type.
2. **Price Query**: The machine displays the price of the selected soda.
3. **Payment**: The user inserts the money.
4. **Validation**: The machine validates the money and checks if it's enough for the selected soda.
5. **Dispense**: If the money is sufficient, the machine dispenses the soda.
6. **Change Return**: If there's any change to return, the machine returns it.
7. **Finish**: The transaction ends, and the machine waits for the next user.

A diagram of soda vending machine

Description automatically generated

* 1. Problem 2 – Networking Terminology:
     1. Explain the following terms briefly:
        1. Network edge
        2. Network core
        3. Store and forward packet switching
        4. Bandwidth
        5. Thoughput

1. The network edge refers to the components of a network that are closest to the end-user devices. This includes devices like routers, switches, and modems that are at the boundary or "edge" of an interconnected network. The edge often encompasses user devices like computers, mobile devices, and IoT devices.
2. The network core, often just called the "core," is the central part of a telecommunications network. It is responsible for transporting data from one edge of the network to another. The core typically has high-speed routers and switches that forward data as quickly and efficiently as possible.
3. In store and forward packet switching, a node (typically a router or switch) receives a complete packet of data, stores it temporarily, and then forwards it to the next node or its final destination. This method ensures that the entire packet is received before it's forwarded, reducing the chances of data corruption.
4. Bandwidth refers to the maximum rate at which data can be transmitted over a network or network segment. It's typically measured in bits per second (bps) or similar units. Bandwidth indicates the capacity of a communication link.
5. Throughput is the actual rate at which data is successfully transmitted over a network, taking into account factors such as network congestion, protocol overhead, and signal interference. It's often less than the theoretical maximum (bandwidth) because of these factors.
   * 1. Explain the advantage of both circuit-switched and packet-switched networks.

Circuit-Switched Networks:

1. Consistent Connection: Once established, the connection offers a consistent and predictable communication channel.
2. Quality: Provides a uniform quality of service since resources are dedicated, making it ideal for voice calls.
3. Less latency: Because a dedicated path is established, there is typically less latency or delay in data transmission.
4. Simplicity: Data flows in a consistent stream, which can simplify certain applications.

Packet-Switched Networks:

1. Efficiency: Resources are used on-demand, allowing for more efficient utilization of network capacity.
2. Flexibility: Multiple communication sessions can share the same network resources, allowing for dynamic and adaptable communication.
3. Scalability: Can handle a large number of users and devices without needing a dedicated channel for each one.
4. Fault Tolerance: If one path fails, data can be rerouted through another path.
5. Supports Complex Data: Designed to handle complex data streams, such as multimedia content, with varying levels of quality of service.
   1. Problem 3 - Packets Transmission and Delays
      1. Consider the following scenario. A link with capacity of 3Mbps is shared by multiple users.

A black text on a white background

Description automatically generated

* + - 1. How many users can use the link at the same time when circuit switching is used, and each user transmits at 15Kbps?

Number of users = (Total link capacity) / (Transmission rate of each user) = (3 Mbps) / (15 Kbps) = 3,000 Kbps / 15 Kbps = 200 users

* + - 1. How many users can use the link at the same time when circuit switching is used, and each user transmits at 1.5Mbps?

Number of users = (Total link capacity) / (Transmission rate of each user) = (3 Mbps) / (1.5 Mbps) = 2 users

* + - 1. If packet switching is used, with 4 users and each user transmits only 10% of the time, what will be the probability that the 4 users are transmitting at the same time?

Probability that all 4 users are transmitting at the same time = (Probability that User 1 is transmitting) \* (Probability that User 2 is transmitting) \* (Probability that User 3 is transmitting) \* (Probability that User 4 is transmitting) = 0.1 \* 0.1 \* 0.1 \* 0.1 = 0.0001

* + - 1. If packet switching is used, with 4 users and each user transmits only 10% of the time, what will be the probability that the 2 users are transmitting at the same time?

C(4,2) = 4! / (2! \* (4-2)!) = 6

Total probability = 6 \* (0.1 \* 0.1 \* 0.9 \* 0.9) = 6 × 0.0081 = 0.0486

* + 1. Suppose a packet with size 5M bits will be transmitted over a link of transmission rate 500Kbs, with distance 7,500km, with propagation speed of 2.5 x 10 km/s.
       1. What will be the propagation delay?

Propagation delay = Distance / Propagation speed = 7,500 km / 2.5 \* 10^5 km/s = 0.03 s = 30 ms

* + - 1. What will be the transmission delay?

Transmission delay = Packet size / Transmission rate = 5M bits / 500K

bits/s = 5 x 10^6 bits / 5 x 10^5 bits/s = 10 s

* + 1. Suppose a router with infinite buffer capacity. Answer the following questions with yes/no:
       1. Can packets be dropped? No
       2. Would there ever be a delay sending the packets to outgoing links? Yes
    2. Consider a scenario where 10 packets arrive to a router where currently there is no queue. Given each packet has a length of 100Kbits and transmission rate of 10Kbs. What will be the average queuing delay?

Transmission time for one packet = Packet length / Transmission rate = 100 Kbits / 10 Kbps = 10 seconds

Total queuing delay = 0 + 10 + 20 + ... + 90 = 10 (0 + 10 + 20 + ... + 90) / 10 = 10 (0 + 9\*10) / 2 = 10 \* 45 = 450 seconds

Average queuing delay = Total queuing delay / Number of packets = 450 seconds / 10 = 45 seconds

* 1. Problem 4 – Traceroute:
     1. The traceroute program relies on a protocol X to calculate the delay between source and the different routers in the path of destination.
        1. What is the name of protocol X? Internet Control Message Protocol
        2. Which field in the IP header is utilized by traceroute to send packets to particular router in the path?Time-to-Live (TTL)
     2. Use the traceroute program for 3 hostnames. Provide screenshots and answer the following question.

A screenshot of a computer

Description automatically generated

A white text with black numbers

Description automatically generated

A white paper with black numbers and numbers

Description automatically generated

* + - 1. Can you identify the Internet Service Provider (ISP) from the hostnames?

verizon-gni.net -> Verizon

* + - 1. Why does the delay for the same router fluctuates (remember traceroute send 3 packets)?

The network traffic on the route can vary, even in short intervals. If the router or the links near it are experiencing higher traffic for one of the packets, the delay might increase for that packet. In addition, some routers might route the packets through different paths for load balancing, leading to variations in the delay. Each of the three packets might take slightly different routes through the network, especially if the network uses multiple paths for redundancy and load distribution.

* 1. Problem 5 – Web Application Architecture

Explain how Web architectures were developed and refined to increasingly support applications with informational, interactive, transactional, and delivery requirements? Please relate to specific architectures, their corresponding protocols, and describe the improvements that were made over time.

1. **Informational:**
   * **Architecture:** Early web architectures were primarily client-server models where web browsers requested static pages from web servers.
   * **Protocol:** HTTP (Hypertext Transfer Protocol) was the primary protocol. Initial versions of HTTP were stateless and request-response oriented.
   * **Improvements:** HTML (Hypertext Markup Language) evolved to incorporate more elements and attributes, allowing for richer content presentation.
2. **Interactive:**
   * **Architecture:** The need for user interaction led to the CGI (Common Gateway Interface) model, where server-side scripts generate dynamic content.
   * **Protocols & Technologies:** Alongside HTTP, languages and platforms like PHP, ASP, and JSP became popular. Databases like MySQL and PostgreSQL were integrated to store and retrieve data.
   * **Improvements:** Web application frameworks like Django, Ruby on Rails, and Express.js emerged to streamline development. AJAX (Asynchronous JavaScript and XML) enabled parts of web pages to be updated without refreshing the entire page, enhancing user experience.
3. **Transactional:**
   * **Architecture:** E-commerce sites and online services required architectures that could handle transactions. This led to multi-tier architectures, separating presentation, business logic, and data layers.
   * **Protocols & Technologies:** HTTPS (HTTP Secure) became essential for encrypting data and ensuring secure transactions. Session management protocols were developed to maintain user states across requests.
   * **Improvements:** Payment gateways and APIs (Application Programming Interfaces) were integrated. Scalability and high availability became crucial, leading to load balancers, distributed databases, and cloud computing platforms.
4. **Delivery:**
   * **Architecture:** With the rise of multimedia content, CDNs (Content Delivery Networks) emerged. CDNs replicate content in multiple locations globally, delivering it from the nearest location to the user.
   * **Protocols:** RTMP (Real-Time Messaging Protocol) became popular for streaming media. Later, more adaptive protocols like HLS (HTTP Live Streaming) and MPEG-DASH arose, adjusting quality based on the user's bandwidth.
   * **Improvements:** Edge computing brought computation closer to the data source, reducing latency. Progressive web apps (PWAs) allowed web applications to function more like native apps, with offline capabilities and push notifications.
   1. Problem 6 – Real-Time Messaging Applications

List at least four mainstream real-time messaging applications. Document the protocols they use (along with references to corresponding IETF RFCs) and explain in detail how they differ. Please provide references and/or links to all documentation sources used to answer this question.

1. **WhatsApp**:
   * Protocol: Signal Protocol (previously known as the Axolotl Protocol)
   * Reference: Signal Protocol doesn't have a direct IETF RFC, but its design and implementations are open-source and can be found on the Signal website.
   * [Signal Protocol Documentation](https://signal.org/docs/)

WhatsApp's protocol provides end-to-end encryption. The Signal Protocol merges the Double Ratchet algorithm, pre-keys, and a triple Diffie-Hellman (3DH) handshake.

1. **Telegram**:
   * Protocol: MTProto (Mobile Transport Protocol)
   * Reference: Like Signal, MTProto does not have an IETF RFC. Details can be found in the Telegram API documentation.
   * [Telegram API Documentation](https://core.telegram.org/mtproto)

MTProto is designed for access to a server API from applications running on mobile devices. It's fast, simple, and layered.

1. **Facebook Messenger**:
   * Protocol: MQTT (Message Queuing Telemetry Transport) for part of its messaging protocol, especially for mobile notifications. However, for its end-to-end encrypted "secret conversations," it uses the Signal Protocol.
   * Reference for MQTT: [IETF RFC 6206](https://tools.ietf.org/html/rfc6206)
   * [Facebook Messenger MQTT Details](https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920/)

Messenger uses MQTT due to its low bandwidth usage and efficient delivery to large numbers of consumers. It's particularly beneficial for mobile where network conditions can be problematic.

1. **Slack**:
   * Protocol: WebSocket for its Real-Time Messaging API.
   * Reference for WebSocket: [IETF RFC 6455](https://tools.ietf.org/html/rfc6455)
   * [Slack API Documentation](https://api.slack.com/rtm)

Slack's use of WebSockets allows for bi-directional communication between a client and server, which keeps data fresh without needing constant polling.

Both WhatsApp and Facebook Messenger's "secret conversations" use the Signal Protocol for end-to-end encryption. Telegram only uses end-to-end encryption for its "secret chats", while Slack does not have end-to-end encryption. In terms of architecture, While WhatsApp and Facebook Messenger focus on encryption and security, Slack emphasizes real-time collaboration and quick information exchange in professional settings. Telegram strikes a balance between speed, encryption, and cloud-based accessibility.

* 1. Problem 7 – Email Applications

List at least five mainstream email applications. Document the protocols they use (along with references to corresponding IETF RFCs) to send and receive emails and explain in detail how they differ. Please provide references and/or links to all documentation sources used to answer this question.

1. Microsoft Outlook
2. Mozilla Thunderbird
3. Apple Mail
4. Google Gmail (Web-based)
5. eM Client
6. **SMTP (Simple Mail Transfer Protocol)**: For sending emails.
   * IETF RFC: [RFC 5321](https://tools.ietf.org/html/rfc5321)
7. **IMAP (Internet Message Access Protocol)**: For receiving emails and syncing them across devices.
   * IETF RFC: [RFC 3501](https://tools.ietf.org/html/rfc3501)
8. **POP3 (Post Office Protocol version 3)**: For receiving emails without the need to sync them across devices.
   * IETF RFC: [RFC 1939](https://tools.ietf.org/html/rfc1939)

**SMTP (Simple Mail Transfer Protocol)**

* **Function**: SMTP is used for sending emails from a client to a server or between servers.
* **Port**: It typically operates over TCP port 25 for clear text and TCP port 587 for secure (encrypted) communications.
* **Operation**: SMTP servers have a store-and-forward mechanism. They receive outgoing email messages and route them to their appropriate destinations. The destinations could be another SMTP server (relay) or the final receiving mail server.

**IMAP (Internet Message Access Protocol)**

* **Function**: IMAP is used to retrieve messages from a mail server. The emails remain on the server, allowing users to organize, delete, or mark their messages as read/unread. This is particularly useful when accessing the email from multiple devices.
* **Port**: It typically operates over TCP port 143 for clear text and TCP port 993 for SSL/TLS secured communications.
* **Operation**: IMAP maintains the email on the server, allowing multiple clients/devices to access the same mailbox, which makes it suitable for users who need to check their email from different locations. With IMAP, folders are often stored on the server, allowing for consistent syncing across devices.

**POP3 (Post Office Protocol version 3)**

* **Function**: Like IMAP, POP3 is used to retrieve messages from a mail server, but with a significant difference: once emails are downloaded to the client, they're often removed from the server (though there are settings in most email clients to keep copies on the server).
* **Port**: It typically operates over TCP port 110 for clear text and TCP port 995 for SSL/TLS secured communications.
* **Operation**: POP3 is suitable for users who only access their email from a single device. Since emails are downloaded and then often removed from the server, it's not ideal for users with multiple devices accessing the same email account. With POP3, the state information, like whether an email has been read, isn't typically synced between devices.
  1. Problem 8 – Network Management Utilities

1. Explain what the following utilities are used for: traceroute, ping, nslookup, ipconfig, dig?

traceroute: This utility helps in mapping the path data packets take to get from one device to another. It displays the sequence of hops the packet takes and the time it takes for each hop.

ping: It's used to check the connectivity between your device and another device on a network (usually the internet). It sends ICMP (Internet Control Message Protocol) packets to the target host and awaits a response.

nslookup: Stands for 'name server lookup'. It's used to query the Domain Name System (DNS) to obtain domain name or IP address mapping or other DNS records.

ipconfig: Used on Windows-based devices to display the current configuration of the installed IP stack on a networked device.

dig: Similar to nslookup, the dig (domain information groper) command is used for querying DNS nameservers for information about host addresses, mail exchanges, nameservers, and other DNS records.

1. Identify at least three more utilities and explain what they are used for.

netstat: This utility displays network connections, routing tables, interface statistics, masquerade connections, etc.

whois: This is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system.

telnet: It's a network protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility.

1. For each one of the utilities introduced in 8.a. and 8.b., provide a detailed usage scenario along with corresponding screenshots as needed to fully document your example.

traceroute:

Scenario: If you're experiencing delays accessing a website, you can use traceroute to see where the potential bottleneck or failure is happening.

A white text with black numbers and numbers

Description automatically generated

ping:

Scenario: You can't access a website, and you want to see if it's down for everyone or just you.

A close-up of a code

Description automatically generated

nslookup:

Scenario: You're setting up an email server and want to verify the MX (Mail Exchange) records of a domain.

A screenshot of a computer

Description automatically generated

ipconfig:

Scenario: You want to know your computer's IP address within the local network.

A screen shot of a computer code

Description automatically generated

dig:

Scenario: You want to check the A (address) record for a domain to determine its IP address.

A black background with white text

Description automatically generated

netstat:

Scenario: You want to see which programs on your computer are making network connections and to where.

A screenshot of a computer

Description automatically generated

whois:

Scenario: You want to find out who owns a particular domain.

A screenshot of a computer

Description automatically generated

telnet:

Scenario: You want to test if you can connect to a server on a particular port, e.g., testing an SMTP mail server connection.

A black background with white text

Description automatically generated

* 1. Problem 9 – Overlay Networks

1. How is peer churn managed in P2P applications such as file-sharing, conferencing, and content distribution?

Redundancy: Store multiple copies of the same data across different nodes. This

ensures that even if some nodes leave, the data is still available in the network.

Replication: Similar to redundancy, it means creating multiple replicas of the same

content. If a node goes offline, the content can be sourced from another node.

Regular Updates: Nodes regularly update their routing tables or lists of neighbors,

which helps in quickly identifying offline nodes and finding alternative routes or

sources.

Time-to-Live (TTL) Values: Assigning a TTL to data or requests can prevent them

from endlessly circulating in the network in case of failures.

Proactive Rejoining: Nodes that got disconnected due to temporary issues (e.g.,

network glitches) can proactively try to rejoin the network.

Stabilization Protocols: Used to ensure that the network's data structure remains intact despite nodes coming and going.

1. Provide specific examples of P2P applications, explain how they specifically handle churn, and estimate the performance improvements achieved in each case. Please provide references and/or links to all documentation sources used to answer this question.

**BitTorrent:**

* Churn Handling: BitTorrent breaks files into pieces and fetches them from multiple sources. If one source goes offline, it can continue downloading from another.
* Performance Improvements: By sourcing content from multiple peers simultaneously, download speeds can be faster compared to downloading from a single source. Additionally, the network efficiently handles high churn rates.
* References: [BitTorrent Protocol Specification](https://www.bittorrent.org/beps/bep_0003.html)

**Skype:**

* Churn Handling: Used a hybrid P2P system where supernodes (more reliable nodes) were crucial. These supernodes helped in maintaining network stability despite regular nodes experiencing churn.
* Performance Improvements: The supernode architecture allowed for scalable, high-quality VoIP calls even with many nodes frequently joining and leaving.
* References: ["An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol"](https://arxiv.org/abs/cs/0412017)