Name Ni, Tianshu Date: 2023.9.21

(last name, first name)

NYU ID: N16371283

Course Section: 001

**Assignment 1**

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

**Professor’s Comments:**

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

**(Sign here)**

* 1. Problem 1 – Modeling delay in a communications network:
     1. If the distances between all nodes in a network were constant (D) and each node (end systems and routers are nodes) could transmit R bits per second, what would the end-to-end delay equation look like for both circuit and packet switching? (The only constant that you may assume is that the speed of light in the given transmission medium is 2\*108 m/sec)

Let N be the number of routers, L be the packet length, R be the link bandwidth (bits/sec), D be the length of physical link, dproc be the processing delay, dqueue be queuing delay, then the end-to-end delay equation for circuit switching is: N(dproc + D/2\*108) + 1/R; the end-to-end delay for packet switching is: N(dproc+dqueue+1/R+D/2\*108).

* + 1. Give four examples of network variables (e.g., number of ISPs) that you expect to remain constant in the Internet and explain why in each case.
* The layers in the Internet protocol stack will remain consistent. Modifying the stack structure would mean significant alterations to both the endpoint operating systems and core network devices, which would incur prohibitive costs.
* The count of levels in ISPs is expected to stay consistent, even though new ISPs might join existing tiers like International/National ISPs, Regional ISPs, and Local ISPs.
* The separation between a satellite and a ground station in a geosynchronous orbit is fixed. To maintain this specific orbit, the distance is always set at 42,164 km.
* The number of global Internet exchange points is likely to remain relatively stable, indicating that the quantity of global ISPs won't rapidly fluctuate.
  1. Problem 2 – IoT Networks
     1. Catalog the main IoT protocols in use today.

MQTT, XMPP, CoAP, AMQP, DDS

* + 1. Why is the proliferation of IoT protocols difficult to handle withing the current Internet network stack and is there a solution to this issue?

It is difficult to handle scalability and maintain network security. Standardization of IoT protocols is the only solution to this problem.

* 1. Problem 3 – Network Models:
* Origins
  + TCP/IP: Developed for ARPANET by U.S. Department of Defense.
  + OSI: Formulated by the International Organization for Standardization (ISO).
* Number of Layers
  + TCP/IP: Consists of four layers.
  + OSI: Comprises seven distinct layers.
* Layer Breakdown
  + TCP/IP:
    - Application (encompasses OSI's application, presentation, session)
    - Transport (corresponds to OSI's transport)
    - Internet (akin to OSI's network)
    - Link (merges OSI's data link and physical)
  + OSI:
    - Physical
    - Data Link
    - Network
    - Transport
    - Session
    - Presentation
    - Application
* Purpose
  + TCP/IP: Conceived for pragmatic application in ARPANET and the Internet's establishment.
  + OSI: Theoretically structured as a guideline for the formulation of network protocols.
* Rate of Adoption
  + TCP/IP: Serves as the backbone of the contemporary Internet.
  + OSI: Extensively taught academically; however, its specific protocol suite saw limited adoption.
* Interoperability
  + TCP/IP: Prioritizes wide-ranging interoperability.
  + OSI: Stresses on stringent modularity and layering.
* Intrinsic Nature
  + TCP/IP: Grounded in practical, real-world protocol requirements.
  + OSI: An idealistic and abstract blueprint for networking.
    1. Explain the main architecture heuristics that are used to design network models and elaborate on their advantages and drawbacks.

Network models, particularly in the realm of communications and software design, often employ a layered architectural approach. This design choice is rooted in the principles of high cohesion and low coupling, ensuring the model's modularity, clarity, and maintainability. A layered architecture's core philosophy is to ensure that each layer focuses on a highly specific function or service, embodying high cohesion. This means that all the components within a given layer are dedicated to achieving a single, well-defined task. On the other hand, low coupling ensures that these layers interact with each other in a minimal and well-defined manner. Specifically, each layer is designed to communicate primarily with its immediate neighbors, particularly the layer just below it. This hierarchical interaction minimizes dependencies between disparate layers.

One significant advantage of this design paradigm is its adaptability. Since each layer is somewhat insulated from the others, changes or upgrades can often be made to a single layer without necessitating a redesign of the entire system. This modular approach promotes flexibility, scalability, and ease of maintenance.

However, this design is not without its challenges. The layered approach can introduce processing overhead. As data traverses through the layers, it may undergo multiple processing steps, sometimes even for similar functionalities across different layers. For instance, both the transport layer and the application layer might have security protocols that need to be separately invoked. This redundancy can slow down data transmission and processing.

* 1. Problem 4 – Segmentation:
     1. Why was segmentation introduced in IPv4?  
        Hence packets may be created in a way that allows them to traverse a link with a smaller MTU compared to the original datagram's size.
     2. How does segmentation relate to encapsulation and where/when does it take place in the network?

Segmentation involves breaking down large data packets into smaller chunks to fit the constraints of network paths, while encapsulation adds headers or trailers to these packets, providing routing and control information. Segmentation primarily occurs at the transport layer to ensure data packets meet the size requirements of the underlying network, whereas encapsulation happens at various layers of the OSI model, adding specific headers as data moves down the layers before transmission.

* + 1. Give examples of negative side effects of segmentation

For instance, if one segment is lost or corrupted during transmission, the entire data packet may need to be resent, leading to increased latency and reduced overall throughput. Additionally, the overhead associated with managing multiple segments can strain network resources, and in some environments, segments may arrive out of order, necessitating complex reordering processes.

* 1. Problem 5 – IETF Standards ([www.ietf.org/rfc.html):](http://www.ietf.org/rfc.html))
     1. What is RFC 2026 about?
* It defines the procedures and criteria used in the Internet standards process.
* It outlines the different stages of standardization and the requirements for progression between these stages.
* It discusses the roles and responsibilities of various entities involved in the standards-making process, such as the Internet Engineering Steering Group (IESG) and the Internet Architecture Board (IAB).
* It provides an essential framework for how Internet standards are developed, reviewed, and approved within the Internet Engineering Task Force (IETF) community.
  + 1. Was HTML standardized by IETF? Why or why not?

HTML was initially standardized by the IETF. The original specification for HTML, RFC 1866, was published by the IETF in 1995. However, as the web evolved, the responsibility for HTML standardization shifted to the World Wide Web Consortium (W3C). The W3C took over because it was better suited to handle the broader aspects of web technologies, whereas the IETF primarily focused on lower-level network protocols. As a result, most of the subsequent advancements and versions of HTML have been overseen by the W3C.

* + 1. How many RFCs are there for TCP and UDP and why is there multiple RFCs for each of these protocols?

TCP – 10 Standardized RFCs

UDP – 3 Standardized RFCs.

There are multiple RFCs for many protocols because the internet and its associated technologies are continually evolving. Multiple RFCs for a single protocol often represent updates, enhancements, or clarifications to previous versions. As issues are discovered, or as technological advancements occur, protocols may need to be refined or expanded. Additionally, a single protocol might have various features or components, and sometimes it's more manageable to address each in separate RFCs. Over time, this iterative process leads to the creation of multiple RFCs associated with a single protocol to capture its complete evolution and breadth.

* 1. Problem 6 – Packet vs. Circuit-Switched Communications:
     1. Is it preferable to use circuit-switched or packet-switched communication when operating in a wireless context? Please explain you answer as well as the service guarantee you would expect.  
        In a wireless context, the choice between circuit-switched and packet-switched communication hinges on the nature of data transmission. For long sessions with steady rates and predictable bandwidth, circuit-switching is more apt. However, for bursty data transmissions, packet-switching is preferable, efficiently managing the shared wireless bandwidth. This distinction applies to both wireless and wired scenarios. Quality of Service (QoS) mechanisms in packet-switched networks optimize user experience, but absolute guarantees can be challenging due to the inherent variability in wireless conditions.
     2. Give two examples of applications for which circuit switching and packet switching would be more suitable respectively.

Circuit switching is more suitable for applications like traditional voice calls and dedicated video conferencing, where a steady and continuous connection is crucial. Packet switching is preferable for applications like web browsing and email, where data is transmitted in bursts and a continuous connection isn't necessary.

* + 1. Describe a model that could be implemented to compute a utilization percentage at all times for overall network paths in circuit-switched communications.

100[Transmission time / (Transmission Time + Propagation Time)]

* 1. Problem 7 - Protocols:
     1. Is there a general protocol that may be used and specialized (i.e., by you and Luke and by the two blue armies respectively) to avoid defeat in cases 1 and 2 below? If there is such a protocol, please use an Event-Condition-Action (ECA) architecture (i.e., name the corresponding events, conditions, and actions associated to the protocol)

Case 1: You and Luke Sky Walker are commanders in the rebel army preparing to attack Darth Vader and his Death Star. By yourselves, neither you nor Luke possesses enough fire power to defeat Darth Vader, but together you can destroy the Death Star. However, you must come to agreement on the precise moment to attack, but you cannot communicate using the normal communications else your presence and location will be detected by Darth Vader.

But you each possess an unlimited number of R2D2 messenger droids that you can use to send messages to each other. But, the droids may be destroyed by the enemy’s PatrolBots, so you have no way of knowing if your message gets through unless Luke sends a droid back to you with a confirmation message.

Suppose your droid gets through to Luke and Luke sends a droid back to you with a confirmation message agreeing to the time to attack, but it is destroyed by a PatrolBot? Should he attack? You haven’t received a confirmation, so what will you do?

Case 2: Two blue armies are each poised on opposite hills preparing to attack a single red army in the valley. The red army can defeat either of the blue armies separately but will fail to defeat both blue armies if they attach simultaneously. The blue armies communicate via an unreliable communications system (i.e., a foot soldier). The commander of one of the blue armies would like to attack at noon. However, if he sends a message to the other blue army ordering the attack, he cannot be sure that the message will get through. He could ask for acknowledgement but that might not get through either.

TCP might be useful, which is designed to provide reliable, ordered, and error-checked delivery of a stream of packets between applications running on hosts on an IP network. It has mechanisms like acknowledgments (ACKs), retransmissions, and timeouts to handle unreliable communication channels.

* **Initial Transmission**:
  + **Event**: Decision to attack.
  + **Condition**: You decide a time for the attack.
  + **Action**: Send an R2D2 droid to Luke with the proposed attack time.
* **Acknowledgment**:
  + **Event**: Luke receives the attack time proposal.
  + **Condition**: The proposed time is acceptable.
  + **Action**: Luke sends a confirmation R2D2 droid (akin to an ACK in TCP) back with the agreed time.
* **Re-transmission (if needed)**:
  + **Event**: You don't receive Luke's confirmation within a certain timeframe.
  + **Condition**: Presumed message loss (similar to TCP timeout).
  + **Action**: Resend the initial message or a new proposed time to Luke.
    1. What are the main differences between human and network protocols? Is comparing the two a viable metaphor/analogy, why or why not?
* Nature: Human protocols are informal, adaptive, and often context-dependent (e.g., waving hello, cultural etiquettes), while network protocols are formal, standardized, and must be strictly followed for successful communication (e.g., TCP, HTTP).
* Flexibility: Humans can infer meaning, recognize non-verbal cues, and adjust their communication dynamically. In contrast, network protocols lack such adaptability and rely on predefined rules and formats.
* Error Recovery: Humans can ask for clarification if something is misunderstood, while network protocols have specific error-handling mechanisms like retransmissions and acknowledgments.

Comparing human and network protocols can be a viable analogy to a certain extent. There are several shared patterns in how both humans and network nodes communicate and share data. In both scenarios, interactions often adhere to an Event-Condition-Action (ECA) framework.