SECTION 1.1

R1. What is the difference between a host and an end system? List several different types of end systems. Is a Web server an end system?

Explanation:

1. **Core Concept:** This question tests your understanding of the most basic terminology in networking: what constitutes a device on the "edge" of the network.
2. **Step-by-Step Breakdown:**
   - **Host vs. End System:** In the context of the Internet architecture, these terms are **synonymous**. There is no technical difference. A host *is* an end system and vice versa. They both refer to any device that is connected to the Internet and is either a source or a final destination of a message.
   - **Types of End Systems:** End systems are diverse. They are not just traditional computers. Think of any device that can run a network application (client or server software).
     - Examples: Desktop PCs, laptops, smartphones, tablets, IoT devices (like smart TVs, security cameras, smart appliances), and **servers** (e.g., web servers, email servers, game servers).
   - **Web Server as an End System:** Absolutely yes. A web server is a perfect example of an end system. It's the destination for HTTP requests and the source for HTTP responses.

Hint: Don't overcomplicate the first part. The terms are used interchangeably in this context. For the list, think of every device you personally use to get online, and also the powerful computers you connect to when you use an online service.

R2. The word protocol is often used to describe diplomatic relations. How does Wikipedia describe diplomatic protocol?

Explanation:

1. **Core Concept:** This question uses an analogy to help you grasp the abstract but critical concept of a "protocol." A network protocol is like a human protocol—a set of rules that govern communication.
2. **Step-by-Step Breakdown:**
   - The goal here is to look up the definition and see the parallels.
   - **Diplomatic Protocol:** According to Wikipedia, diplomatic protocol is "a set of rules, procedures, and conventions that govern the official interactions between representatives of different states." This includes rules for ceremonies, forms of address, and order of precedence.
   - **Connection to Networking:** A network protocol (e.g., TCP, HTTP) is similarly a set of rules and conventions that govern how network entities (software, hardware) interact. It defines the **syntax** (format of messages), **semantics** (meaning of each section of the message), and **timing** (order and speed of message exchange).

Hint: The key takeaway is that both definitions revolve around **rules** and **conventions** that ensure smooth, predictable, and successful communication between independent parties, be they nations or computers.

R3. Why are standards important for protocols?

Explanation:

1. **Core Concept:** This addresses the "why" behind the entire standards ecosystem (e.g., IETF, RFCs, IEEE). Without standards, the Internet as we know it would not exist.
2. **Step-by-Step Breakdown:** Imagine a world without standards:
   - **No Interoperability:** A device from Company A would not be able to communicate with a device from Company B. We'd have isolated islands of technology, not a global network.
   - **Stifled Innovation:** Developers would have to write different software for every different type of network hardware, which is incredibly inefficient. Standards create a stable, common platform upon which everyone can innovate.
   - **Market Fragmentation:** It would be much harder for new companies to enter the market if their products couldn't work with the existing ones.

Hint: Think of standards like a common language. If everyone in the world spoke a different language, global collaboration would be impossible. Protocols standards are the common language of the Internet.

SECTION 1.2

R4. List four access technologies. Classify each one as home access, enterprise access, or wide-area wireless access.

Explanation:

1. **Core Concept:** This question is about the "last mile" technologies that connect end systems (homes, businesses, mobile users) to their Internet Service Provider (ISP) and thus to the global Internet.
2. **Step-by-Step Breakdown:**
   - You need to recall different physical media and their typical use cases.
   - **Example Answer:**
     1. **DSL (Digital Subscriber Line):** Uses telephone lines. **Home access.**
     2. **Ethernet:** Uses twisted-pair copper cables in a local area. **Enterprise access** (typical in office/buildings).
     3. **Cellular Networks (e.g., 4G/5G):** Uses radio spectrum. **Wide-area wireless access.**
     4. **Satellite Internet:** Uses communication satellites. Can be for **home access** (in rural areas) or mobile.

Hint: Categorize them based on their physical scope and typical customer. "Home" is individual residences, "Enterprise" is organizations/businesses, and "Wide-area wireless" is for users on the move over large geographic areas.

R5. Is HFC transmission rate dedicated or shared among users? Are collisions possible in a downstream HFC channel? Why or why not?

Explanation:

1. **Core Concept:** This tests your understanding of Hybrid Fiber Coax (HFC) architecture, which is used by many cable Internet providers. It combines fiber optic and coaxial cable.
2. **Step-by-Step Breakdown:**
   - **Shared vs. Dedicated:** The key characteristic of cable Internet is that users in a neighborhood **share** the access network's bandwidth to the ISP. It is not a dedicated point-to-point link like DSL. Your speed can be affected by your neighbors' activity.
   - **Downstream Collisions:** The downstream channel is a broadcast channel from the Cable Modem Termination System (CMTS) at the ISP to all the homes. The CMTS is the only transmitter in this direction.
     - Since only one entity (the CMTS) is sending, and all homes are just listening, there is **no possibility for collisions** on the downstream channel. A collision requires two or more devices transmitting on the same medium at the same time.

Hint: For the collision part, draw a simple diagram. One central transmitter (ISP) and many receivers (homes). Can two receivers cause a collision if they are only listening?

R6. List the available residential access technologies in your city. For each type of access, provide the advertised downstream rate, upstream rate, and monthly price.

Explanation:

1. **Core Concept:** This is an applied question designed to make you research the real-world market for Internet access. The answer will vary drastically by location.
2. **Step-by-Step Breakdown:**
   - **Action:** Go to the websites of major ISPs in your area (e.g., Comcast/Xfinity, Charter/Spectrum, AT&T, Verizon FIOS, local providers).
   - **Find:** Look for their residential Internet plans. They will always advertise the **downstream** (download) rate (e.g., "200 Mbps"), the **upstream** (upload) rate (often much smaller, e.g., "10 Mbps"), and the **monthly price** (often promotional for the first 1-2 years).
   - **Technologies:** Common ones you'll find are Cable (HFC), DSL, Fiber-to-the-Home (FTTH), and sometimes fixed wireless.

Hint: This is not a theoretical question. You must perform a quick web search. The values are publicly advertised by ISPs.

R7. What is the transmission rate of Ethernet LANs?

Explanation:

1. **Core Concept:** Ethernet is the dominant LAN technology, and it has evolved over decades with vastly different speeds.
2. **Step-by-Step Breakdown:**
   - This is a bit of a trick question. There is no *single* transmission rate.
   - Ethernet has many standards. You should be familiar with the common ones:
     - Traditional Ethernet: **10 Mbps**
     - Fast Ethernet: **100 Mbps**

        - Gigabit Ethernet: **1 Gbps**
        - 10-Gigabit Ethernet: **10 Gbps**
        - ... and even faster standards (40 Gbps, 100 Gbps) exist for data centers.

Hint: The correct answer is to list the common variants and their rates, not to give one number.

R8. What are some of the physical media that Ethernet can run over?

Explanation:

1. **Core Concept:** Ethernet is a link-layer protocol that is independent of the physical medium. This is a key to its longevity and success.
2. **Step-by-Step Breakdown:**
    - Think about the cables you've seen connecting computers.
    - **Common Media:**
        - **Twisted-pair copper cable:** (e.g., Category 5e, 6, 6a). This is the most common for connecting PCs to switches.
        - **Coaxial cable:** Used in early Ethernet ("thinnet", "thicknet").
        - **Fiber optic cable:** Used for high-speed backbone links and connections between buildings. Common standards use multimode or single-mode fiber.

Hint: Look at the cable connecting your laptop to a wall jack (copper) and imagine the cables running between floors in a building (fiber).

R9. HFC, DSL, and FTTH are all used for residential access. For each of these access technologies, provide a range of transmission rates and comment on whether the transmission rate is shared or dedicated.

Explanation:

1. **Core Concept:** Compare and contrast the key residential access technologies.
2. **Step-by-Step Breakdown:**
    - Create a small table in your mind or on paper.
    - **HFC (Cable Internet):**
        - **Rates:** Typical downstream rates range from **10-1000+ Mbps**. Upstream is often 5-50 Mbps.
        - **Shared/Dedicated: Shared** among users in a neighborhood.
    - **DSL (e.g., ADSL):**
        - **Rates:** Rates vary with distance from the telco central office. Downstream is typically **1-100 Mbps**. Upstream is lower.
        - **Shared/Dedicated: Dedicated** point-to-point connection from the home to the telco office. Not shared with neighbors.
    - **FTTH (Fiber-to-the-Home):**
        - **Rates:** Can be very high and symmetric. Common plans from **100 Mbps to 1 Gbps** and beyond (e.g., 2 Gbps, 5 Gbps).

        - **Shared/Dedicated:** Architectures can vary, but it is often effectively **dedicated** or shared among a very small number of users, providing consistent performance.

Hint: The "shared vs. dedicated" property is a fundamental differentiator between DSL and Cable Internet, with FTTH often being superior to both.

R10. Describe the most popular wireless Internet access technologies today. Compare and contrast them.

Explanation:

1. **Core Concept:** Understand the landscape of wireless connectivity, which is dominated by two families: Wi-Fi for local access and Cellular for wide-area access.
2. **Step-by-Step Breakdown:**
    - **Wi-Fi (IEEE 802.11):**
        - **Description:** A wireless *local area network* (WLAN) technology. Provides access within a home, office, or coffee shop (a "hotspot"). The wireless router is typically connected to a wired Internet connection (like DSL or Cable).
        - **Range:** Short (~100 feet indoors).
        - **Use Case:** Fixed and nomadic indoor access.
    - **Cellular (4G LTE, 5G):**
        - **Description:** A wireless *wide area network* (WWAN) technology. Provides access over a large geographic area via a network of cell towers.
        - **Range:** Long (miles from a tower).
        - **Use Case:** Mobile access anywhere within coverage.
    - **Bluetooth:** Mention it as a very short-range personal area network (PAN) technology for connecting peripherals, but it is not a primary *Internet access* technology.
    - **Comparison:** Contrast them primarily on **range** and **mobility**. Wi-Fi for high-speed local access, Cellular for ubiquitous mobile access.

Hint: Think about how you get online with your phone. You use Wi-Fi at home and cellular data when you're driving. These are the two most popular technologies.

SECTION 1.3

R11. ...what is the total end-to-end delay to send a packet of length L?

Concept: Store-and-forward packet switching delay calculation.
Hint: The switch must receive the *entire* packet before it can begin forwarding it. The total delay is the sum of the transmission delays on each link. Formula: `Delay = L/R1 + L/R2`.

R12. What advantage does a circuit-switched network have over a packet-switched network? What advantages does TDM have over FDM...?

Concept: Trade-offs between circuit-switching and packet-switching.
Hint: Circuit-switching's advantage is **performance guarantees** (constant rate, no delay/jitter). For TDM vs. FDM,

think about the technology: TDM (digital time slices) is simpler and cheaper to implement with digital systems than FDM (analog frequency filters).

R13. Suppose users share a 2 Mbps link...

Concept: Statistical multiplexing and queuing theory fundamentals.
Hint:
a. Circuit switching reserves a dedicated slice. How many 1 Mbps slices fit in a 2 Mbps pie?
b. If the sum of transmitting users' rates is <= link rate, no queue builds. If it's greater, a queue forms.
c. Simple probability given: 20% or 0.2.
d. Use the binomial probability formula or logic for "all three transmitting": `0.2 * 0.2 * 0.2`.

R14. Why will two ISPs at the same level often peer?... How does an IXP earn money?

Concept: Internet structure and business relationships.
Hint: Peering saves money (no transit fees) and improves performance (direct path). An IXP earns money by charging its member ISPs a fee (usually a monthly port fee) to connect to the exchange switch.

R15. ...Describe Google's network. What motivates content providers to create these networks?

Concept: Content provider networks and the shift in Internet hierarchy.
Hint: Google's network is a massive **private global network** interconnecting its data centers with its own fiber and switches. Motivations: **1. Cost:** Avoid paying transit fees to Tier-1 ISPs. **2. Performance:** Have greater control over routing, reducing latency for their services. **3. Reliability.**

SECTION 1.4

R16. List the delay components... Which are constant and which are variable?

Concept: The four types of packet delay.
Hint: Remember the four: **1. Nodal Processing:** (Variable, but often small and constant). **2. Queuing Delay:** (Highly Variable, depends on traffic). **3. Transmission Delay:** `L/R` (Constant for a fixed packet size and link rate). **4. Propagation Delay:** `d/s` (Constant for a fixed path).

R17. Visit the Transmission Vs Propagation Delay interactive animation...

Concept: The crucial difference between transmission delay (time to put bits on the wire) and propagation delay (time for a bit to travel across the wire).
Hint: To have the sender finish first: Use a **huge packet (large L)** and a **slow link (small R)**, making `d_trans` large. Use a short distance and fast propagation speed, making `d_prop` tiny. For the reverse: Use a **tiny packet (small L)** on a **fast link (large R)**, making `d_trans` tiny. Use a long distance, making `d_prop` huge.

R18. How long does it take a packet... to propagate... Does this delay depend on packet length? Transmission rate?

Concept: Isolating and understanding propagation delay.
Hint: Propagation delay is purely a function of physics: distance and speed. The formula is `d_prop = distance / propagation_speed`. It does **not** depend on the packet length (`L`) or the link's transmission rate

(R). A bit travels at the speed of light in the medium, regardless of how many bits are in front of it or how fast they were put on the link.

R19. Suppose Host A wants to send a large file to Host B...

Concept: Throughput, which is the effective transfer rate, is limited by the **bottleneck link**.
Hint:
a. Throughput = min($R1$, $R2$, $R3$) = 500 kbps.
b. Time = File Size / Throughput. Convert bytes to bits (multiply by 8).
c. The new bottleneck is min(500k, 100k, 1000k) = 100 kbps. Recalculate.

R20. ...how does end system A create packets?... what information... does the router use?... Why analogous to asking directions?

Concept: Packetization and the role of the network layer.
Hint:

- **Creating packets:** Break file into chunks. Add a header to each chunk with addressing info (source/destination IP address) to form a packet.
- **Router's decision:** The router looks at the **destination IP address** in the packet's header and consults its **forwarding table** to decide the outgoing link.
- **Analogy:** You (the packet) don't know the entire route to the destination city. You ask at each intersection (router) for the next step towards your final address (IP address).

R21. Visit the Queuing and Loss interactive animation...

Concept: Relationship between traffic intensity and queuing delay/loss.
Hint: The maximum emission rate is the rate at which packets arrive at the queue ($a * L$). The minimum transmission rate is the link rate (R). Traffic intensity = $(a * L)/R$. If this is > 1, the queue will *always* grow infinitely and packets will be lost. The time to loss depends on the *initial* size of the queue, which is set randomly in the animation, so two runs will likely give different results.

SECTION 1.5

R22. List five tasks that a layer can perform...

Concept: The purpose of layering.
Hint: Tasks include: error control, flow control, segmentation and reassembly, multiplexing, connection setup, reliable data transfer, routing, addressing, message generation. Yes, it's possible (and common) for the same task (e.g., error detection) to be performed at more than one layer for different reasons.

R23. What are the five layers in the Internet protocol stack?...

Concept: The Internet (TCP/IP) protocol stack.
Hint: Remember "Please Do Not Throw Sausage Pizza Away":

1. **Physical:** Bits on the wire.
2. **Link:** Frame transfer between adjacent nodes (e.g., Ethernet).
3. **Network:** End-to-end datagram delivery across multiple links (IP, routing).
4. **Transport:** Process-to-process data transfer (TCP, UDP).

5. **Application:** Network applications (HTTP, SMTP, DNS).

R24. What is an application-layer message?...

Concept: Protocol Data Units (PDUs) at different layers.
Hint:

- **Application-layer message:** The payload created by the application (e.g., an HTTP request).
- **Transport-layer segment:** A message encapsulated with a transport layer header (TCP/UDP header).
- **Network-layer datagram:** A segment encapsulated with an IP header.
- **Link-layer frame:** A datagram encapsulated with a link-layer header and trailer (e.g., Ethernet header/trailer).

R25. Which layers does a router process?... a link-layer switch?... a host?

Concept: The functionality of different network devices.
Hint:

- **Router:** Is an IP device. It processes layers **1, 2, and 3**. It looks at IP addresses.
- **Link-layer switch:** Is a frame-forwarding device. It processes layers **1 and 2**. It looks at MAC addresses.
- **Host:** The source and destination of all communication. It processes **all five layers**.

SECTION 1.6

R26. What is self-replicating malware?

Concept: Malware classification.
Hint: It's malware that can copy itself and spread to other systems without human intervention. A **virus** is the classic example, as it requires attaching itself to a host program. A **worm** is a standalone self-replicating malware.

R27. Describe how a botnet can be created and how it can be used for a DDoS attack.

Concept: Botnets and DDoS attacks.
Hint:

1. **Create:** Infect many computers (drones/zombies) with malware that allows remote control.
2. **Use for DDoS:** The attacker (botmaster) commands all these drones to simultaneously send a massive volume of traffic to a target server. This floods the server's resources (bandwidth, CPU), making it unable to respond to legitimate requests.

R28. ...List some of the malicious things Trudy can do...

Concept: The "man-in-the-middle" attack.
Hint: Trudy can: **1. Eavesdrop** on the entire conversation. **2. Modify messages** being sent in either direction. **3. Inject her own fake messages** pretending to be Alice or Bob. **4. Hijack the session** and take over the communication. **5. Launch a replay attack** by recording and re-sending messages later.

PROBLEMS (P1 onwards)

(The problems are more complex and often require mathematical derivation or design. The hints will focus on the approach.)

P1. Design and describe an application-level protocol...

Hint: This is a protocol design question. Define the **message types** (e.g., `Auth_Request`, `Auth_Response`, `Balance_Query`, `Withdrawal_Request`, `Success`, `Error`). For each, specify the **fields** they contain (e.g., card number, PIN, amount, response code). Define the **sequence** of messages for a successful and failed withdrawal. State your assumption about the transport service (e.g., it is reliable, like TCP).

P2. Generalize [the delay] formula for sending P packets...

Hint: The formula for one packet over N links is `N * (L/R)`. If you send P packets back-to-back, the first packet takes `N*(L/R)` time. The *last* packet will leave the source at time `(P-1)*(L/R)` after the first. Its delay will be `(P-1)*(L/R) + N*(L/R)`. So the total time to receive all P packets is `(N + P - 1) * (L/R)`.

P3. [Steady rate application]

Hint:
a. A **circuit-switched** network is more appropriate because it provides a **constant, guaranteed rate**, which matches the application's steady, long-lived traffic pattern. Packet switching could introduce variable queuing delay and jitter.
b. Even though the sum of data rates is less than link capacities, **congestion control is still needed**. Why? Because the applications are bursty (they generate an N-bit unit *every k time units*). If many applications happen to generate their burst at the same time, the *instantaneous* arrival rate at a switch could exceed its output link capacity, causing queues to build and packets to be lost.

P4. [Circuit-switched network with 4 circuits per link]

Hint: Draw the 4-switch ring. Remember, each *link* can support 4 simultaneous calls.
a. The maximum is limited by the links. The network can support up to 8 calls (4 on the top links, 4 on the bottom links?).
b. All connections between A and C. They can go A-B-C or A-D-C. The limiting factor is the links from A and the links to C. The maximum is 8? (4 via each path? But check the links involved).
c. You need to check if the required paths (A-C and B-D) can be mapped onto the 4 available circuits on each link without exceeding the capacity of any single link.

P5. [Car-caravan analogy]

Hint: This analogy models **transmission delay** (time for the caravan to pass a tollbooth) and **propagation delay** (time for the caravan to travel the road).
a. End-to-end delay = time for first car to reach first booth + time for whole caravan to pass all three booths + propagation time for first car to go from first to last booth? Carefully calculate the time when the last car clears the final tollbooth.
b. With fewer cars, the *transmission delay* at each tollbooth decreases, but the *propagation delay* remains the same.

P6. [Explore propagation and transmission delay]

Hint: This is a fundamental exercise.
a. `d_prop = m / s` (distance / speed).
b. `d_trans = L / R` (bits / rate).
c. `d_end-to-end = d_prop + d_trans`.
d. At `t = d_trans`, the last bit is *just leaving* Host A.
e. If `d_prop > d_trans`, at `t = d_trans`, the first bit is still propagating along the link. It has not yet reached Host B.
f. If `d_prop < d_trans`, at `t = d_trans`, the first bit has already arrived at Host B and the last bit is just leaving Host A.
g. Set `d_prop = d_trans` and solve for `m`: `m / s = L / R` -> `m = (L * s) / R`. Convert units consistently (bytes to bits!).

P7. [VoIP delay calculation]

Hint: Calculate the cumulative delay for a single bit. It experiences: 1. **Processing delay** at A to be packetized (wait until its packet is filled). 2. **Transmission delay** of its packet onto the link. 3. **Propagation delay** across the link. 4. **Processing delay** at B to be depacketized (wait until the entire packet is received before decoding can begin for the first bit). The key is that a bit in the middle of the packet has to wait for the whole packet to be created and received.

P8. [Packet switching with 120 users]

Hint:
a. Same as R13a. `Number of users = Total Link Rate / User Rate = 10 Mbps / 0.2 Mbps = 50`.
b. `p = 0.1` (10% of the time).
c. This is a **binomial probability** problem. `P(n users transmitting) = C(120, n) * p^n * (1-p)^(120-n)`.
d. `P(>=51) = 1 - P(n <= 50)`. You would typically use a binomial calculator or table for this.

P9. [1 Gbps link]

Hint:
a. `N = New Link Rate / User Rate = 1 Gbps / 100 kbps = 1,000,000 / 100,000 = 10,000` users.
b. The probability that more than N users are active is given by the **binomial distribution**: `P(>N users) = Sum from k=N+1 to M of [ C(M, k) * p^k * (1-p)^(M-k) ]`.

P10. [Total end-to-end delay over three links]

Hint: The packet must be transmitted and propagated over each link. It also suffers processing delay at each switch (but not at the end systems). The total delay is: `(L/R1 + d1/s1) + d_proc + (L/R2 + d2/s2) + d_proc + (L/R3 + d3/s3)`. Plug in the given values. Watch units! (bytes vs bits, km vs m, msec vs sec).

P11. [Cut-through switching]

Hint: If the switch does cut-through (instead of store-and-forward), it eliminates the per-packet transmission delay at the switch. It only incurs a tiny processing delay per bit. The total delay becomes essentially the transmission delay on the first link plus the propagation delay on all links: $L/R + (d1+d2+d3)/s$. The transmission delays on links 2 and 3 are overlapped and hidden.

P12. [Queuing delay calculation]

Hint: When your packet arrives, it sees:

- A packet that is halfway done being transmitted. The remaining time for this packet is $(L - x) / R$.
- Four full packets ahead of it in the queue. The time to transmit these is $4 * (L / R)$.
  So, your queuing delay is $((L - x)/R + 4*(L/R))$.
  The general formula is $((L - x)/R + n*(L/R))$.

P13. [Average queuing delay for N packets]

Hint:
a. The first packet has 0 queuing delay. The second waits L/R time. The third waits $2L/R$, ..., the last waits $(N-1)L/R$. The average is the sum of these waits divided by N: $(0 + 1 + 2 + ... + (N-1)) * (L/R) / N = [(N-1)N/2] * (L/R) / N = (N-1)(L/(2R))$.
b. If packets arrive every LN/R seconds, the system is in a steady state. The average queuing delay should be the same as in (a), which is $(N-1)(L/(2R))$.

P14. [Total delay and plotting]

Hint:
a. Total Delay = Queuing Delay + Transmission Delay = $(IL/R(1-I)) + (L/R)$. Simplify this expression.
b. Plot Total Delay (y-axis) as a function of L/R (x-axis). You will see that as L/R increases (or as traffic intensity I approaches 1), the total delay shoots up towards infinity.

P15. [Delay in terms of a and μ]

Hint: The link's transmission rate in packets/sec is $\mu = R / L$. The traffic intensity is $I = La/R = a / \mu$. Substitute $I = a/\mu$ and $L/R = 1/\mu$ into the total delay formula from P14.

P16. [Little's formula]

Hint: Little's Formula is $N = a * d$. You are given:

- N = Average number in buffer + being transmitted = 100 packets + 1 packet = 101 packets.
- d = average total delay = queuing delay + transmission delay = 20 msec + (1 packet / 100 packets/sec) = 0.02 sec + 0.01 sec = 0.03 sec.
  Now solve for a: $a = N / d = 101 \text{ packets} / 0.03 \text{ sec} \approx 3366.67 \text{ packets/sec}$.

P17. [Generalize end-to-end delay]

Hint:
a. Generalize Equation 1.2 for N links with different rates `R_i`, prop delays `d_prop_i`, and proc delays `d_proc_i`. The total delay is the sum of `d_trans_i`, `d_prop_i`, and `d_proc_i` for all links and switches.
b. Now add a queuing delay `d_queue_i` at each node (switch). The total delay becomes the sum of all components.

P18. [Traceroute experiment]

Hint: This is a lab exercise. Use the `traceroute` (or `tracert` on Windows) command. The key is to run it at different times to see the variability of Internet paths and delays. You are analyzing real-world data.

P19. [Metcalfe's law]

Hint: If there are `n` users, and each sends one message to every other user, how many unique pairs are there? The number of messages sent is the number of edges in a complete graph with n nodes: `n*(n-1)/2`. This is proportional to `n^2`, which supports Metcalfe's law (value ∝ n^2).

P20. [Throughput for M client-server pairs]

Hint: The throughput for *one* flow is limited by the bottleneck: `min(R_s, R_c, R)`. But if there are `M` flows sharing the network link `R`, the total throughput cannot exceed `R`. However, the server and client links might also be bottlenecks. The general per-flow throughput is `min(R_s, R_c, R/M)`.

P21. [Throughput over multiple paths]

Hint:

- **One path:** The throughput is the bottleneck rate on that path: `min(R_1^k, R_2^k, ..., R_N^k)`.
- **All M paths:** The server can achieve a total throughput equal to the *sum* of the throughputs achievable on each individual path.

P22. [Packet loss probability]

Hint:

- Probability of success on one link is `(1-p)`. For a path with `h` links, the probability of success is `(1-p)^h` (if losses are independent).
- The number of transmissions needed is a **geometric random variable**. The average number of transmissions required for success is `1 / (probability of success) = 1 / ((1-p)^h)`.

P23. [Bottleneck link and packet spacing]

Hint:
a. The inter-arrival time at the destination is determined by the **bottleneck link**. The second packet gets queued behind the first at the slow link. The inter-arrival time is `L / R_c`.
b. To avoid queuing, the time between sending packets `T` must be large enough so that the second packet arrives at the bottleneck *after* the first packet has finished transmitting on it. `T` must be at least `L/R_c - L/R_s`.

P24. [FedEx vs. 100 Mbps link]

Hint: Calculate the time to transfer 50 TB over the 100 Mbps link. `Time = (50 * 10^12 * 8 bits) / (100 * 10^6 bits/sec)`. Convert this time to hours or days. Compare it to 24 hours (overnight). You will find that FedEx's "sneakernet" is vastly faster for this enormous amount of data. This highlights the difference between **bandwidth** and **latency**.

P25. [Bandwidth-delay product]

Hint:
a. `R * d_prop`. First calculate `d_prop = distance / speed`.
b. The max number of bits in the link is the bandwidth-delay product. It's like the volume of the pipe.
c. It is the maximum amount of data that can be "in flight" on the link at any given time.
d. Width of a bit = (Length of the link) / (Number of bits in the link) = `m / (R * d_prop)`.
e. Substitute `d_prop = m/s` into the formula from (d).

P26. [Bit length equals link length]

Hint: Set the width of a bit (from P25e) equal to the link length `m`. Then solve for `R`. You will get `R = s` (the propagation speed)! This is a fun result.

P27. [Repeat P25 with R=500 Mbps]

Hint: Repeat the calculations in P25 with the new, much larger `R` value. Notice how the number of bits in the link (BDP) increases and the physical length of a bit decreases.

P28. [Stop-and-wait protocol]

Hint:
a. Simple transmission: `Time = (File Size) / R + d_prop` (but since it's one big message, `d_prop` is negligible compared to the huge `d_trans`).
b. With stop-and-wait, the sender sends one packet and waits for an ACK. The time per packet is `d_trans_packet + d_prop + d_ACK_trans + d_prop ≈ (L_packet/R) + 2*d_prop` (since ACK is small). Multiply by the number of packets.
c. You will find that breaking it up and waiting for ACKs adds a huge overhead due to the propagation delay. This is why stop-and-wait is inefficient for long-delay paths.

P29. [Satellite link]

Hint:
a. `d_prop = distance / speed`. The distance for a geostationary satellite is ~36,000 km.
b. `R * d_prop`.
c. To be continuously transmitting, the time to transmit the photo must be *at least* as long as the time to take the next photo (60 seconds). So, `x / R >= 60 sec`. Solve for the minimum `x`.

P30. [Air travel analogy and headers]

Hint: Yes, there is an equivalent. The "headers" added at each layer of the airline stack are the tickets, boarding passes, baggage tags, and routing information attached to you and your luggage. This information is used by the "protocols" at each stage (check-in, security, gate, baggage handling) to route you to your final destination.

P31. [Message segmentation]

Hint: This is a crucial problem.
a. Without segmentation, the entire message must be received by the first switch before any bit is forwarded. Total time = 3 * (`L_message / R`).
b. With segmentation, the first packet is forwarded as soon as it is received. The second packet starts transmission immediately after the first. This leads to **pipelining**.
c. The total time with segmentation will be significantly less. Calculate it as the time for the first packet to reach the destination *plus* the time for the remaining 99 packets to be transmitted by the source (as they are pipelined through the network).
d. Reasons: Reduces delay (as shown), reduces packet error rate (a single bit error only corrupts one small packet, not the whole message), and is fairer to other flows (don't monopolize a link for a long time).
e. Drawbacks: Overhead from adding headers to every packet, and increased processing complexity at the receiver to reassemble the original message.

P32. [Message Segmentation interactive animation]

Hint: This is a lab exercise. The animation should visually confirm the calculations from P31. Adding propagation delay adds a constant offset to the timing but does not change the fundamental benefit of pipelining with segmentation.

P33. [Optimal segment size S]

Hint: The total delay has two components: 1. The time to transmit the extra header bits (`80 * number_of_packets / R`). 2. The time to transmit the actual data bits (`F / R`). The number of packets is `F / S`. Write the total delay as a function of `S`: `Delay(S) = (F/S * (80+S)/R) + (F/R)`. Simplify and find the value of `S` that minimizes this function. (Note: The `F/R` term is constant and can be ignored for the minimization. You will find the optimal `S` is independent of `F`).

P34. [Skype from PC to phone]

Hint: This requires a **gateway**. The Skype client on your PC uses Internet protocols (likely over UDP) to send digitized voice to a Skype server/gateway that is also connected to the traditional telephone network (PSTN). This gateway translates the Internet voice packets into a format understood by the PSTN and initiates a call to the destination phone number.