Q1. **Circuit switching in the Internet**

In the Internet backbone, packet-switching is employed as the switching mechanism instead of circuit-switching. Circuit switching however was used before the Internet in the early telephone networks.

**Q1.1:** Name at least 2 reasons why employing circuit switching is not a good idea for the Internet in general.

Possible reasons:

- High latency to establish an end-to-end circuit, which is especially harmful to short flows;
- The workload of users within the Internet are bursty, and thus pre-allocating bandwidth would lead to under-utilization as it takes time before the circuit is freed again;
- The Internet routers and switches can fail, and it takes time before these failures are detected and propagated to all ISPs to renegotiate the circuit. In packet switching only the local ISP in which the fault occurs has to reroute packets around the fault;
- Circuit switching forces massive amount of circuit state to be kept in the switches. Secondly, additional complexity and scheduling is needed to enforce the circuit-allocated bandwidth.

**Q1.2:** Despite all the downsides you answered in Q1.1, can you give an example of an application which would greatly benefit from a circuit-switched Internet over the current packet-switched Internet? Why?

Circuit switching does not need large buffers because application rates are fixed. Without buffers the latency of packets is very low. Applications such as gaming,
tele-surgery or Skype calls have a traffic flow which is long-lasting, and their quality of experience (QoE) is heavily dependent on response time. Consistently guaranteed low round trip time would improve their functioning substantially, which is not anywhere near-optimal in the packet-switched Internet.

Q2: How would you implement flow prioritization (i.e., one connection receives more bandwidth (or better quality of service in general) than another) in packet switching? And in circuit switching?

Packet switching requires a priority field in the packet header, such that the switch can queue the packets based on their priority. Circuit switching already provides the ability to assign more bandwidth to particular flows as it has to explicitly reserve its bandwidth when a flow arrives to the network.

Q3: Making use of cloud services instead of own networking infrastructure has become increasingly popular, especially among small- to medium-sized companies not in the IT field (i.e., a company selling items online). From the perspective of network connectivity, explain why such a company will have difficulty with its own infrastructure in terms of (a) handling peak usage, (b) providing global quality of service, and (c) maintenance, and how cloud services ameliorate these issues.

Possible reasons:

• **Lack of statistical multiplexing ability**: A company has certain periods of massive amounts of shoppers visiting their website, e.g. on Black Friday. If the company would have to build a network to handle this peak load (as they would lose revenue if they did not), the network would remain severely underutilized during the remainder of time. In general, in order to facilitate growth, a first expensive initial purchase of equipment has to be made: in cloud-services you can scale on-demand.

• **Locality**: The company’s infrastructure would need good Internet access and manage content distribution on a world-wide scale. Buying small amounts of infrastructure and deploying/growing it globally is infeasible. Cloud services already have presence around the world and offer the rental of small amount of infrastructure.

• **Maintenance**: Maintaining and operating data centers (across the globe) is highly specialized labor that a non-technical company does not possess. Cloud service companies (i.e., Amazon AWS) are specialized in maintaining large network infrastructure, as it’s its core business.

Q4. Provisioning
Assume that $N$ users share the same up-link to their Internet Service Provider. The link capacity is $C$ Mbps. When online, a user will require $U$ Mbps bandwidth. The probability of a user being online at any point in time is $P_{\text{online}}$. 
independent of other users.

**Q4.1:** In closed form, what is the probability $P_{\text{inadequate}}$ that there is not enough available link capacity to fully satisfy the needs of the online users?

Let’s say the amount of discrete users online is $X$. Because the probability of a user being online is independent from other users, $X$ follows a binomial distribution:

$$P(X = x) = \text{Bin}(x; P_{\text{online}}, N)$$

The probability of being inadequate is when $X > \text{floor}(\frac{C}{U})$, i.e. when the demands of the users exceed the available link capacity. From this follows:

$$P_{\text{inadequate}} = P(X > \text{floor}(\frac{C}{U})) = \sum_{i=\text{floor}(\frac{C}{U})+1}^{N} \text{Bin}(i; P_{\text{online}}, N)$$

**Q4.2:** For $N = 5$ users, $C = 20$ Mbps, $U = 7$ Mbps, and $P_{\text{online}} = 0.3$: what is the probability $P_{\text{inadequate}}$ of the link being inadequate?

$$\text{floor}(\frac{C}{U}) = \text{floor}(\frac{20}{7}) = 2$$

$$P_{\text{inadequate}} = \sum_{i=3}^{5} \text{Bin}(i; 0.3, 5) = 0.16308$$

**Q4.3:** Because provisioning for peak usage is too expensive, provisioning is done by keeping risk within a reasonable bound. In this exercise, risk is the probability of the link being inadequate. Given $(N, U, P_{\text{online}}, P_{\text{max.risk}})$ we want to know: what is the minimum $C$ such that $P_{\text{inadequate}} \leq P_{\text{max.risk}}$? Write the algorithm that answers this question.

The problem statement is:

$$P_{\text{inadequate}} = \sum_{i=\text{floor}(\frac{C}{U})+1}^{N} \text{Bin}(i; P_{\text{online}}, N) \leq P_{\text{max.risk}}$$

**Insights:**

- Because of the floor and summation, there is no continuous analytic solution for $C$;
- $\text{Bin}(i, P_{\text{online}})$ is independent of the choice of $C$, and $0 \leq i \leq N$;
- $i = \text{floor}(\frac{C}{U}) + 1$, so we only need to test $C$ as multiples of $U$. Any capacity less than $U$ added to the multiple of $U$ will not allow us to have more users simultaneously online.
Algorithm:

```plaintext
function MIN-LINK-CAPACITY-NEEDED(N, U, P_{\text{online}}, P_{\text{max.risk}})
    sum = 0.0
    for i \in \{0, 1, ..., N\} do
        sum = sum + Bin(i; P_{\text{online}}, N)
        if sum \geq 1 - P_{\text{max.risk}} then
            return i \times U
        end if
    end for
end function
```

Q4.4: Implement the algorithm of Q4.3 in a language of your choosing. For $N = 130$ users, $U = 4$ Mbps, and $P_{\text{online}} = 0.2$: what is the minimum link capacity $C_{\text{min}}$ needed such that $P_{\text{inadequate}} \leq 0.01$?

$C_{\text{min}} = 148$

Algorithm in Java:

```java
package com.example.workspace.xpt.networks2018;
import java.math.BigDecimal;
import java.math.BigInteger;
public class Exercise1 {
    public static void main(String args[]) {
        BigInteger N = BigInteger.valueOf(130);
        BigDecimal U = BigDecimal.valueOf(4);
        BigDecimal POnline = BigDecimal.valueOf(0.2);
        BigDecimal PMaxRisk = BigDecimal.valueOf(0.01);
        System.out.println(calculateCMin(N, U, POnline, PMaxRisk));
    }
    private static BigDecimal calculateCMin(BigInteger N, BigDecimal U,
            BigDecimal POnline, BigDecimal PMaxRisk) {
        BigDecimal sum = BigDecimal.ZERO;
        for (int i = 0; i < N.intValue(); i++) {
            sum = sum.add(binomialPMF(BigInteger.valueOf(i), N, POnline));
            if (sum.compareTo(BigDecimal.ONE.subtract(PMaxRisk)) >= 0) {
                return U.multiply(BigDecimal.valueOf(i));
            }
        }
        return null;
    }
    private static BigDecimal binomialPMF(BigInteger kBI, BigInteger nBI,
            BigDecimal pBD) {
        return new BigDecimal(factorial(nBI)).divide(new BigDecimal(factorial(kBI).multiply(factorial(nBI.subtract(kBI))))).multiply(pBD.pow(kBI.intValue())).multiply((BigDecimal.ONE.subtract(pBD).pow(nBI.subtract(kBI).intValue())));
    }
    private static BigInteger factorial(BigInteger i) {
        if (i.equals(BigInteger.ZERO)) {
            return BigInteger.ONE;
        } else {
            return i.multiply(factorial(i.subtract(BigInteger.ONE)));
        }
    }
}
```

Q5: The `hosts.txt` file still exists on systems (i.e. in `/etc/hosts/`). Why is it only writable with administrator privileges? Similarly, why is it important that we only use trusted DNS servers?
Firstly, many applications rely on DNS to function, so if the DNS records are invalid these applications will not work properly. Secondly, when the DNS records are maliciously edited, an attacker can redirect your requests. For example, you go to bank.com in your browser and are redirected to a phishing copy hosted on an attacker’s server.

**Q6: Throughput in a Network**

In this exercise you will calculate the throughput in a network and the time it takes to send a file over this network. We will neglect any sort of delay and focus solely on the transmission rates of links. We will further assume that the server and the client in the picture below are the only ones communicating on the network and therefore congestion (and therefore packet loss) is also irrelevant.

1. The diagram below shows the capacities for each link in Mbps. Compute the throughput of this network also in Mbps.

2. Assuming this network is much larger and more complex, is there a way to systematically compute the throughput?

3. Given a file F of size 19 million bits. How long does it take to transmit the whole file (again, neglecting any delay except for transmission delay)?

1. The throughput is given by the maximum flow, which is 5Mbps.

2. This corresponds to finding the max flow in a graph and can be systematically solved using for example Ford Fulkerson. See [https://en.wikipedia.org/wiki/Ford%E2%80%93Fulkerson_algorithm](https://en.wikipedia.org/wiki/Ford%E2%80%93Fulkerson_algorithm)
3. The throughput of the network is 5Mbps and the file has $F = 19 \times 10^6$ bits. Dividing $F/R$ yields 3.8s.

**Q7: Symmetric and Asymmetric Connections**
Still today many homes use a DSL connection to connect to the internet. As you have read in the book, this connection is not symmetric, i.e., the upload and download stream have different bandwidths.

1. Does the up- or download connection have higher bandwidth? What is (historically) the reasoning behind introducing this asymmetry?

2. Assume you are running a business that provides a content service to your customers over the internet. Is making use of such an asymmetric connection a good idea? Why or why not?

3. In recent years, fiber optic connections to the home (FTTH) have gained in popularity. FTTH can deliver up to Gbps rates. Name two reasons why this technology has not become the most prevalent yet.

1. The download connection has higher bandwidth, by a multiplicative factor. These home connections are optimized for the type of usage that on average occurs on them. For an average home connection the amount of data that is downloaded is substantially larger than the data that is uploaded. The reason for this is that requests for data such as homepages or even videos are many times smaller than the data that is being requested, at least if we look at the average request and data size. A good example is when a client wants to load a webpage which can nowadays easily be multiple MB large while the request for this webpage is a single one lined request. (The same applies to the original use of cable, which is broadcasting television.)

2. If you are a content provider many requests for data are sent to you, which uses your download speed. However, as a content provider, you will need to send significantly more data to the customer/user as a response, which uses your upload connection. As we have mentioned in previous question, these connection usually have a higher download bandwidth than upload bandwidth and that the data is (usually) larger than the requests. Therefore for a content provider such a asymmetric connection is matched exactly in the incorrect direction, the data has to be sent through the low bandwidth (upload) connection and the request come in through the high bandwidth connection (download). For this reason businesses often use different offerings from their respective ISPs than consumers.

3. (a) FTTH requires, as the name suggests, fiber optic cables, which are not laid in most older homes. (b) New technologies have sped up DSL and cable, offering higher up- and download rates than the original protocols.

**Q8: DNS Statements**
1. Every IP address maps to a host name. (true / false) false

2. A host name can map to multiple IP addresses. (true / false) true

3. No two host names can map to the same IP addresses. (true / false) false

4. There can be one or more physical servers behind an IP address. (true / false) true

5. Your local DNS server doesn’t store any DNS records itself and always performs lookups via the root DNS servers. (true / false) false

Q9: DNS Questions

Q9.1: What is the main difference between an iterative and recursive DNS query from the viewpoint of the host making the query?

Recursive queries directly return the requested record and offload the work to the queried server. With an iterative query the host only receives the name of the next DNS server to query until it reaches the authoritative NS for the requested resource.

Q9.2: In the absence of recursive query support by the DNS servers, a user needs to resort to iterative queries, which can be a lot slower. Which mechanism is in place within the DNS protocol to speed up (iterative) queries?

Caching. Each record is associated with a Time-To-Live (TTL), which is set by the authoritative NS for that record. Hosts and servers (and even your applications) can cache DNS records for TTL seconds after receiving them. This speeds up resolution time because your computer probably already knows about a TLD server for ch. and maybe even ethz.ch. and thus doesn’t have to make those queries.

Q9.3: In order to perform any iterative DNS query from scratch, one needs to contact a root DNS server. Which mechanisms are in place for root DNS servers to deal with the traffic resulting from this and to prevent future unnecessary traffic?

(1) There are 13 root servers that have knowledge of all TLD servers. (2) Each of these root servers is replicated multiple times across the globe using the same IP address and leveraging BGP anycast. (3) If the root server to query is chosen at random from the 13 servers, traffic should be balanced between root servers. (4) Due to caching, most queries don’t need to start at the root server since computers or your local DNS server often have a lot of TLD servers already cached.
Q9.4: Why would someone associate multiple IP addresses with the same host name?

(1) Load balancing: the order of the A records can be randomized and thus different queries will return different IP addresses. This can be used to balance requests across multiple IP addresses. (2) Redundancy: if the first IP cannot be reached, an alternative is provided.