DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master Practical Course
Computer Network Simulation

27.11.2022

Assignment 1
Part 1 - Theoretical Questions

Group 2

Zafer Tan Cankiri
Erdem Ege Marasli

Question-1: What metrics could/should be used to assess network conditions in a wired network? Describe at least 5 metrics.

1. Latency: It is the amount of time it takes for a packet to travel from its origin to its destination in a network. It is measured as round trip time (RTT) as a common practice, which is the time spent delivering a request to the destination and getting a response from it. It is important to know the RTT of a network since it is used to determine how long a sender should wait until it gets the acknowledgment of the packet it sent. This metric is also used in congestion control mechanisms like TCP BBR.

2. Throughput: It refers to the volume of data that can be sent in a specific amount of time in a network from one point to another. Usually, it is measured in bits per second (bit/s or bps). Therefore, it is used as "Internet Speed" in common sense. The speed tests that can be conducted from "speedtest.net" or "fast.com" give the throughput.

3. Packet Loss: It is the number of packets that were lost during the travel in a network, which means the packets were successfully sent out from one node towards other nodes but lost in between the ends due to a data corruption, full buffers of the routers, etc. It can be used to determine the network performance. Usually it is expressed as a percentage of the total number of sent packets. A packet loss more than 3% indicates a poor network performance.

4. Jitter: It is the variation in latency. Normally, it is expected to have the same RTT value for a communication between a point A and point B. If there is a high jitter in the network, the RTT values change significantly during the communication. If there is a low jitter in the network, then the RTT values change not that much insignificantly during the communication. These changes in the communication affect the RTT expectations of the applications and cause unordered packet deliveries. Jitter mostly affect the real-time applications such as VoIP applications since they are based on the forwarding of packets at a constant rate and sequentially. The problems like speeding up and slowing down of the voice during a VoIP call are the results of jitter in the network.

5. Packet Delivery Ratio (PDR): It is calculated by dividing the number of packets received by the number of packets sent. A high PDR means high performance in the network since it states the delivery performance of the packets.

Ref: https://medium.com/obkio/how-to-measure-network-performance-5-performance-metrics-c034e5dedf28

Question-2: What affects data traffic within an application?
Briefly describe at least 5 factors.

1. Use Case of your Application: The requirements of your application affect the data traffic. For example, if you have a real-time video conference application, your application will send small-sized packets frequently to reflect immediate changes. However, if you have a file transfer application, your application will send large-sized packets with less frequency. Therefore, traffic pattern is affected.

2. Security Concerns: Mostly mobile network operators, use proxy caches to reduce traffic. However, this approach requires application data to be unencrypted. For example, if the application uses HTTP as the underlying protocol then its data can be cached by the network operators to reduce the traffic but if the application uses HTTPS as the underlying protocol, then its data cannot be cached since the same content will be seen different for each different sessions.

3. Reliability Concerns: If your application requires reliability, it will use TCP instead of UDP. Since TCP has a lot of overheads such as the handshake mechanism. It will create much more packets to ensure reliability. This increases the traffic when compared with a UDP-based application.

4. Number of Users: If a large number of users use your application, there will be much more traffic.

5. Starvation Due to Congestion Control: The transport layer protocol that the application uses also affect the data traffic. For example, if the application runs on a TCP implementation that uses TCP Vegas and there is another flow that uses TCP Cubic in the same network, then our application would be starved in terms of bandwidth since TCP Vegas starves against TCP Cubic.

Ref : https://www.rfc-editor.org/rfc/rfc793
Ref : https://www.rfc-editor.org/rfc/rfc768
Ref : https://www.rfc-editor.org/rfc/rfc3143
Ref : https://acn.net.in.tum.de/slides/04_TCP_UDP_QUIC.pdf

Question-3: Describe at least 6 differences between TCP and UDP.
Provide two examples of applications that each of the transport protocols would be best suited for, one for each protocol. Briefly describe your choices and why the protocols are suitable.

1. Reliability:
   - TCP is a reliable protocol that ensures the delivery of the data from sender to receiver.
   - UDP is an unreliable protocol that does not guarantee the delivery of the data from sender to receiver.
2. Type of Service:
   - TCP is a connection-oriented protocol that has handshake and tear-down phases to establish and close connections.
   - UDP is a datagram-oriented protocol that is stateless and does not require an established (agreed beforehand) connection to send data.
3. Multicasting and Broadcasting:
   - TCP supports only unicasting since it requires an established connection (connection-oriented) to deliver data.
   - UDP supports unicasting, multicasting and even broadcasting since it does not require an established connection. It can send the same data to the multiple targets at once.
4. Overhead:
   - TCP has a 20 Bytes header due to the additional functionalities (like congestion control, flow control) it has. In addition, it requires more resources for these additional functionalities and it is slower since it needs to maintain a reliable transfer (additional waiting for acknowledgement messages or adjustments for congestion windows).
   - UDP has an 8 Bytes header since it only carries the mandatory fields for a delivery (Source and destination ports, length, checksum) and nothing extra. Since UDP does not have a reliability concern, it has less overhead and works a lot faster than TCP.
5. Sequencing:
   - TCP has sequencing feature. The packets are delivered in order to the receiver application (layer).
   - UDP does not support sequencing. The order of the packets should be handled by the receiver application (layer).
6. Error Checking and Correction:
   - TCP has an enhanced error checking mechanism since it has flow control and provides acknowledgements in addition to checksums. Since TCP is a reliable protocol, it does not deliver the corrupted packets and corrects them via retransmissions.
   - UDP has only error checking with checksum it carries in the header but it does not provide any correction mechanism. It simply drops the problematic packets instead of delivering them to the application layer.

Example Applications:

- File transfer applications that uses "File Transfer Protocol (FTP)" in the application layer uses TCP since FTP based on TCP because file transfer operations requires reliable transfer of the data. A corruption or a lost packet during the delivery causes corruption in the whole file. In addition, the files should be delivered as they are so the sequential delivery is also necessary which is also supported by TCP.
- Real-time communication applications such as VoIP use UDP as the underlying protocol since it is much faster than TCP. In real-time communication, speed is more important than reliability because little errors can be ignored. For example, if there is a delay in the transmission of the speakers' voices during a conversation, the people cannot communicate fluently and cannot understand each other. Therefore, data transfer should be fast. On the other hand, even if some words are lost during this conversation, these words can be understood from the overall context of the talk and communication can be maintained. In this case, these losses are negligible. Therefore, UDP is a better choice since it is faster although it is unreliable.

Ref: https://www.geeksforgeeks.org/differences-between-tcp-and-udp

Question-4: Imagine you are a network engineer working for TUM. You are tasked with design of a campus wide network that must support the needs of 50000 students at once. To verify the functionality of your network you decide to model it first with a simulation in OMNeT++. Describe your approach. What will be the limitations of your approach and which simplifications you will have to make to get a result in realistic time frame? Will you run into any scalability issues with the selected approach? If so, how would you circumvent them.

First of all, it is necessary to determine how users will connect to the network and to analyze the situation on campus. Questions such as how many buildings are there, and which buildings are the users in the most, should be answered. In addition to these, the way of connecting the users is decided. It is necessary to decide whether the users will be offered the possibility of a wired connection or only the possibility of a wireless connection.

Accordingly, we would start to sketch the network in OMNeT++. We would add switches and access points (APs) to the buildings to provide wired and wireless connectivity to the users accordingly. To be able to finish the design as soon as possible, we would go with a setup that requires fewer configurations. Therefore, we would use plug-and-play access points and switches at the ends. Then, we would designate a central point to place the actual router which will be connected to these switches and the outside world. All the connection requests, traffic management, and routing would be handled in this main router.

Then we would implement this setup in OMNet++. Even to make it more realistic, we would record some real word application data with Wireshark and implement these patterns in OMNet++. Maybe a middle-ware tool can be built to feed the OMNet++ with PCAP files.

This simulation would be very hard to run with 50000 students at once. Even if we could run this simulation somehow, this design would end up with a single point of failure problem. In this design, all the traffic goes through the main router which is very difficult to handle for a single device. In addition, all the packets would be forwarded to a single point which would create congestion in the data medium toward the main router. There is a necessity for scaling. Therefore, to reduce the load, we would add additional routers to the buildings on the campus and we would connect them to create networks of networks this would also allow us to create different sub-nets.

In addition, we would replace this one main router with several routers, each having access to the outside world (Border routers). We would design an autonomous system like this on the campus and let the routing algorithms like RIP, OSPF, and BGP take place and handle the traffic and distribute it. If we could not run this last scenario with all these devices in the simulator, we would design simulations of these subnets separately. Then we would prepare a separate simulation of the main network that connects the subnets. We would run all these simulation setups separately but evaluate the results altogether.