Study Guides on Computer Networks and Distributed Systems

UDP and TCP transport protocols

The solutions provided hereon have the purpose of helping students improve their grasp about Distributed Systems; they should not to be taken as a reference for how exam questions should be completed whatsoever.

Motivation

SD study guides contain diverse resources for your preparing this course such as proposed and solved exercises, illustrating examples and recommendations for study. I recommend that you work them in synchrony with the course, that way, you will not be swamped with too much material to prepare as the term exam date approaches. At specific times in the semester, I will request that you submit specific questions contained in these study guides, which entails your keeping a record of your solutions.

I encourage you to produce your own outline of each lesson, particularly by examining sections of the textbook other than those explicitly explained. As to specific concepts that you might find difficult to grasp, I recommend that you consider Cal Newport advice, which he provides in his book *How to become a straight-A student*:

"The most effective way to imprint a concept is to first review it and then try to explain it, unaided, in your own words. If you can close your eyes and articulate an argument from scratch, or stare at a blank sheet of paper and reproduce a solution without a mistake, then you have fully imprinted the concept"

Basic References

This section contains a listing of recommended resources you can consult to solve the questions and exercises. Note that in this course about Distributed Systems we use two essential textbooks: Computer Networks (Peterson & Davie) and Distributed Systems (Coulouris, Dollimore, Kindberg, etc). You can find both textbooks in the School Library, however, I recommend that you get the pdf document to each one, since they will allow you to carry out your searches and, if necessary, you can print them out. In this Study Guide we used only the textbook by P&D alongside with my PPT lecture presentation:

- Chapter 5 Computer Networks by Peterson & Davie, Ed. 5
- PPT Presentations about UDP and TCP protocols:
  - http://paloalto.unileon.es/ds/index.html
- Commented past exams:
  - http://paloalto.unileon.es/ds/index.html

Questionnaire

The following questions 1, 2 and 3 deal with the write operation on a datagram socket, the resulting maximum sizes allowable and the protocol stack behavior. To start off, we recommend that you scan exercise 2 from the following CN exam solution:

http://paloalto.unileon.es/cn/Q/CN-Ex1-2018-RefSOL.pdf

1. The Length field of UDP datagrams represents the total size of the datagram in bytes and including the header, which consumes 8 bytes. Calculate the maximum theoretical number of bytes that can be encapsulated in a single UDP datagram.

2. Regarding the result that you have just obtained in the previous question, can it be considered that it represents the maximum number of bytes a programmer will be authorized to include in a single write operation on a datagram socket? (The operation might be a datagram socket sendto() function call).

3. Continuing with the topic of the previous two questions, consider now IP fragmentation and its influence in the answers to the above questions, if any. Build an overall explanation of the interaction between max write size at the datagram socket interface and the transmitting interface MTU or the path MTU.

4. The Internet is a great communication and collaboration medium, its capabilities are impressive; however, looking at it with a sufficient level of detail, we discover that the services it provides result somewhat imperfect and must be complemented by the intervening hosts so that the application programs running in those hosts offer some guarantees to their users. Explain the IP Service Model, also known as *Best Effort Model*, which constitutes a brief summary of the raw Internet capabilities and shortcomings.

   This is a review question about Computer Networks that is developed in Ch.1 and Ch. 5 of the textbook by Peterson and Davie (P&D)

5. Explain the purpose of Transport Protocols. Transport protocols belong to OSI architecture layer 4 and have specific missions to accomplish, which are completely different than those of the other OSI layers, then, recall those layers and their main functions and highlight the Transport Protocols.

   Review the initial lecture slides included in this presentation and also the OSI architecture in P&D Ch.1.

   `http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf`

6. The UDP protocol is one of the constituent protocols of the TCP/IP architecture

   a. Which TCP/IP layer does UDP reside on?
      Layer 3

   b. Why is UDP known as the *simple* multiplexer? Consequently, might we ask whether there exists a *complex* multiplexer?
      UDP is the simple multiplexer because its essential mission consists of providing a

multiplexing key for identifying running applications in a host to the Internet via their respective UDP socket

c. What UDP fields constitute its multiplexing key?
   Only the datagram's Destination Port

d. What RFC number specifies the UDP protocol?
   RFC 768

7. What is the protocol number assigned to UDP in an IP packet's multiplexing key?

   Consult https://tools.ietf.org/html/rfc790

8. Explain why UDP offers a non-reliable service

   UDP service model is non-reliable because it provides no delivery guarantees. Review the initial lecture slides included in this presentation:

   http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf

9. Briefly comment on the correctness of this statement: *UDP's demux key is comprised of the Source and Destination ports.*

   That is correct, the MUX key to the UDP protocol is comprised of two 16-bit fields: The destination port and the source port.

10. List the fields of a UDP Datagram that constitute its multiplexing key.

    See answer to question no. 6.

11. Briefly discuss the following statement: The DNS protocol (Domain Name System) can be run over UDP or TCP, depending on specific technical circumstances, this is due to the fact that UDP is not reliable, i.e., when reliability is needed, DNS will use TCP.

    Usual DNS query traffic is served by UDP; however, zone transfer operations involving DNS servers are served by TCP, since the large amounts of information exchanged requires a high degree of reliability.

12. For broadcast communications on a single network, which protocol would you use, UDP or TCP?

    Only UDP, since TCP, by definition is unicast

13. What does UDP communicate, computers, networks, internet hosts or processes running in those hosts? Explain your answer.

    UDP communicates Datagram Sockets (Communication endpoints), which are opened and used by host processes

14. Same question as the preceding one, but, this time we want to consider TCP.

    TCP communicates Stream Sockets (Communication endpoints), which are opened and used by host processes. In this case, the communications are connection-oriented, i.e., each socket only communicates with another socket.

15. What is the protocol number assigned to TCP in an IP packet's multiplexing key?

    Consult https://tools.ietf.org/html/rfc790

16. List the three main functions of TCP

    See slides 14 and 15 in this presentation:
    http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf

17. A TCP segment uses a multiplexing key, list its fields and comment whether all those fields span other protocol units (The IP packet that encapsulates the TCP segment). Note the radical differences between demultiplexing TCP segments' payloads and UDP datagrams'.

    See slide 20 in this presentation: http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf

18. What does "stream service" mean when applied to TCP?
    TCP is a connection-oriented and byte-oriented protocol, consequently, the information sent across a connection is organized as a stream of bytes where each byte is assigned a sequence number that will allow the receiver to reliably rebuild the information sent by the sender.

19. List the segments exchanged by a TCP client and a TCP server when they establish a TCP connection (3-way handshake); in each of the segments, highlight the flags that are activated alongside with their meaning. Also, highlight an essential TCP options set in the segments interchanged.

    See slide no. 57 in this presentation:

    http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf

20. Continuing with TCP's 3-way handshake, do the Initial Sequence Numbers set by each of the two parties (The client and the server) have to be the same integer?

    The Initial Sequence Number to each of the directions of a TCP connection is randomly chosen for reliability reasons. No obligation whatsoever applies to the likeness of both ISNs

21. A TCP Client (C) runs in a host located at 8 hops from a Server with which it has established a TCP connection. C is connected to the network by Wi-Fi NIC transmitting at 54 Mbps and the average Rtt is 71 ms. Calculate the 2BD product (Bandwidth x Delay). Later, we'll build upon this example an illustration of the significance of the 2BD product in TCP connections.

    We can reasonably assume that the bottleneck link is the Wi-Fi link itself, therefore the

2BD product = $2 \cdot 54 \cdot 10^6$ bits/s $\cdot 71 \cdot 10^{-3}$ s = $7{,}67 \cdot 10^6$ bits

22. Is TCP a secure communication protocol?

    TCP includes no security-specific features; however, we must emphasize that TCP is a reliable transport protocol

23. Is it true that there are two types of TCP segments: Data Segments and Control Segments?

    No, TCP doesn't make any distinction whatsoever between control and data segments. Data sent in one direction is always piggybacked in each segment with control information relevant to the other direction

24. What's the *technical term* used to refer to the fact that TCP segments contain data sent in a direction (A->B) alongside with acknowledgements to data received in the other direction (B->A). Beyond recalling the term just requested, it's important that you successfully grasp the concept.

    Piggybacking (See previous question)

25. A new TCP segment has been received by TCP, explain how the receiving process is identified so that it can effectively read the segment's payload?

    The process that is to receive the payload contained in the segment is identified by using the TCP multiplexing key which is comprised of the Src IP, Src Port, Dst IP and Dest Port

26. When does a TCP module send a segment that has the SYN flag set?

    The only segments that can have the SYN flag set are the two first segments sent in the 3-way handshake

27. When does a TCP module send a segment that has the ACK flag set?

    All the segments exchanged across a TCP connection must have the ACK bits set, except the first one which has the SYN bit set

28. TCP receives a segment with ACK set and the ACK number field contains 1000. Explain what interpretation should the receiver apply to that segment ACK fields.

    The receiver of this segment implicitly assumes that all the bytes it sent to the other party of the connection, beginning with the ISN and ending and including SN 999 have been successfully received.

29. An IP packet arrives at host A whose protocol field contains 4, when its payload is deencapsulated, which protocol module is it delivered to?

    `Consult` https://tools.ietf.org/html/rfc790

30. Host A's TCP module receives a segment which Src Port field contains 15882 and which Dst Port field

contains 80: explain how TCP is able to make the decision regarding what is the final destination for the payload contained within this segment.

Please, refer to the solution to a similar question above.

31. What can we state that TCP connects: computers, networks, internet hosts or processes running in these hosts?

Please, refer to the solution to a similar question above.

32. "ACKs are a special type of TCP segment totally separated from data segments", explain whether this statement is correct or not.

Please, refer to the solution to a similar question above.

33. A process writes a block of 10 bytes on an open Java Socket, how many of these bytes will be immediately transmitted over the underlying TCP connection in a single segment? We are not seeking an exhaustive answer to this question, your comments will suffice.

TCP offers a stream service, therefore, we cannot ascertain how many bytes will be chosen by TCP to be transmitted in the next segment; that depends on whether the Nagle algorithm is enabled on the socket and on the dynamic TCP state (SS or CA).

34. [Español, Pregunta de examen] El host de un extremo de una conexión TCP envía un segmento con el flag ACK activado y con ACK number = 10000 ¿**Cuál** de las siguientes opciones es la caracterización más apropiada de la conexión TCP?

- El último segmento recibido por el host contenía datos hasta el número de secuencia 9999 inclusive
- El host ha recibido segmentos hasta el número de secuencia 10000 y el siguiente número de secuencia esperado es el 10001
- El ultimo segmento recibido por el host tenía un número de secuencia 9999
- El ultimo segmento recibido por el host contenía datos hasta el número de secuencia 9999 inclusive y el siguiente número de secuencia esperado es el 10000
- *El host ha recibido segmentos que incluyeron hasta el número de secuencia 9999 (inclusive); el siguiente número de secuencia esperado por el receptor es el 10000*
- El ultimo segmento recibido por el host incluía datos hasta el número de secuencia 10000 inclusive

Comentarios: En este tipo de pregunta tendréis que elegir la mejor de las opciones ofrecidas, esto significa que puede haber otras opciones correctas aparte de la mejor, pero, se os pide que decidáis, de acuerdo con vuestros conocimientos y vuestro criterio cuál es la mejor. La pregunta se evalúa con un esquema pasa/no-pasa, es decir, recibís la puntuación completa de la pregunta, cualquiera que ésta sea, o bien, recibís 0 puntos; en ningún caso esta pregunta resta.

35. A TCP transmitter receives 3 duplicate ACKs for some ACK number, what should the transmitter do now?

Retransmit the segment containing bytes starting at the ACK number; only that segment *should be retransmitted.*

36. "*ACKs are a special type of TCP segment totally separated from data segments*", explain whether this statement is correct or not.

    Refer to a similar question solved above

37. Every time TCP receives a new segment it reacts by sending an ACK corresponding to it. Please, comment the preceding statement according to the explanation we provided on the lecture.

    TCP, as specified in the relevant RFC's, reacts by sending an immediate ACK in response to some specific segments, for example, an out-of-order segment. Normally, in bulk transmissions, TCP only ACKs every other received full segment, at least. Sometimes, some versions of TCP may transmit stretch ACKs which span more than two received, in-sequence segments.

38. Use the result obtained in exercise no. 18 above to explain the significance of the 2BD product in TCP connections.

    Both, the Transmit and Receive buffers must have a size larger than the product 2BD if the TCP connection is to achieve the highest possible throughput

39. Estimate the value of the 2BD product (Bandwidth x RTT) of a Internet path which bottleneck link is a Gigabit Ethernet (Gbe) interface and the RTT measured at one of the end hosts to the other host is 0,6 ms?

    If the bottleneck link is the Gbe link itself, the 2BD product $= 2 \cdot 1 \cdot 10^9$ *bits/s* $\cdot 0{,}6 \cdot 10^{-3}$ *s* $= 1{,}2 \cdot 10^6$ *bits. This is the minimum size of the Rcv and Tr buffers.*

40. In a Gbe network, a TCP connection is established between two hosts. Calculate a reasonable value for the TCP's sliding window size. Discuss whether that value for the sliding window size can be represented in TCP's segment WS field and how this can be solved.

    (See question no. 36). The size of the initial sliding window $1{,}2 \cdot 10^6$ *bits* $\cdot$ *1 Byte/8 bits =* $150 \cdot 10^3$*Bytes. This window size cannot be represented by the WS field in the TCP segment, which is only 16-bit wide. Extending the field size entails specifying a TCP option upon the 3-way handshake (Window Size scaling factor), in this case, representing* $150 \cdot 10^3$*Bytes requires* $\mathrm{ceil}(\log_2(150 \cdot 10^3)) = 18$ *bits, therefore the scaling factor would be* $18 = 16 + 2$*, that is,* $2^2 = 4$*.*

41. Use Wireshark to determine the AWS value of a few TCP connections over a Gbe network like that in the preceding question (You may want to use Lab B6's Gbe LAN). Test with different TCP/IP stacks: Linux, BSD, Windows, etc.

    This is a practical exercise, in fact. I provide no solution to it with the intention of leaving it for you to practice.

42. What does a TCP sender do to know what size of the Flow Control window it can use when transmitting segments to the other party of the TCP connection?

A sender discovers the remaining buffer space at the receiver by considering the AWS (Advertised Window Size) field in the segments received. The effective window size is the result of subtracting the number of bytes that are in flight at this time (Transmitted and not yet acknowledged) from the value contained in the AWS field.

43. Read textbook section 5.2.5 *Triggering Transmission,* then, explain what timer is referred to in the closing lines of this section where it speaks about the push operation.

    The dynamic state of TCP establishes that the reception of a segment that ACKs a new block of data contiguous to the data block received so far, i.e., an ACK that advances snd.una (LastByteAcked) is an authorization for the transmitting TCP to send new one or more segments. The advancing ACKs act as a clock that times when TCP can send new data segments into the network.

44. How many simultaneous TCP connections can a web server have? We do not want an exhaustive calculation but a simple estimation and discussion of the involved factors.

    We have to consider the four components of the TCP multiplexing key, by simply observe the huge number of IPs and port numbers, we conclude the result will be huge. The potential number of connections that can be formed by combining different values of the multiplexing key will be much larger than the real size of the memory resources a server can allocate to those TCP connections.

45. A new TCP segment has been received by TCP, explain how the receiving process is identified so that it can actually read the segment's payload?

    Again, consider the TCP multiplexing key. Observe slide no. 20 in the following presentation:

    `http://paloalto.unileon.es/ds/lect/BasicTCP-1.pdf`

46. What's the fundamental difference between Slow Start and Congestion Avoidance?

    While in SS, TCP will send a new TCP segment for each ACK received in each Rtt, whereas in CA, TCP will send at most one additional full MSS in each Rtt.

47. What *network element* informs a TCP sender that there is congestion and that, consequently, the Congestion Window must be closed?

    Considering the basic TCP, without advanced features that we have not covered (ECN, for example), there exists no such element informing TCP about the congestion taking place; it's the transmitting TCP that becomes aware of congestion when the segments ACKing previously sent data, don't arrive or arrive so much in late.

48. What is a network power curve, what does it represent?

See Fig. 6.3 in the textbook. We didn't cover this topic in the lectures, therefore we will include no question about it in the term exam.

49. What situations cause a TCP sender to enter Slow Start? Is Fast Retransmit one of them?

    The behavior of TCP maybe different across its different versions. For example, in the Tahoe version, TCP enters SS when initializing the connection and after the RTO time fires; by contrast, 3-DUP causes TCP to reduce CWND to SSTRSH/2 and proceed linearly thereafter by sending at most one additional full segment in each Rtt.

50. An IP packet arrives at host A whose protocol field contains 4, when its payload is deencapsulated, which protocol module is it delivered to?

    See a similar question above (Multiplexing at the IP layer)

51. Host A's TCP module receives a segment which Src Port field contains 15882 and which Dst Port field contains 80: explain how TCP makes the decision regarding which is the final destination for the payload contained within this segment?

    See a similar question above (Multiplexing at the TCP layer)

52. Two hosts, A and B, have an active TCP connection. Tell in which TCP state is each host if none of the two has closed the connection yet. Consult the TCP state diagram in the textbook.

    See TCP state diagram in the textbook. The state at which C and S must be is "established", if the 3-way handshake was successfully completed and none of the has issued a connection close yet (Segment with the FIN bit set).

53. For each of the following events happening during the evolution of a TCP connection, mark its corresponding TCP state transition on the State Diagram (You can find it on the last slide of the BasicTCP-1.pdf at paloalto.unileon.es or in the Textbook by P&D).

    a. Welcome socket is created *(LISTEN)*
    b. Active open by the client *(SYN_SENT)*
    c. Bidirectional data transfer is taking place, now (Mark a state, not a transition, in this case) *(ESTABLISHED)*
    d. Server affirmatively responds to client's active open *(SYN_RCVD)*
    e. Server starts shutdown*(FIN_WAIT_1)*
    f. Client sends active open to server, which hasn't created a Welcome Socket*(NOT REPRESENTED IN THE STATE DIAGRAM; TCP stack sends a RST and ACK segment to client)*

54. Host A carries out a TCP active open. Explain how an active open is executed in Java and in C with Berkeley Sockets.

    Consult the presentations corresponding to the last two practicals and, if necessary, the C and Java code.

55. Same as preceding question, this time we ask you about a passive open.

Proceed like in the preceding question

56. Is it possible to implement a distributed system based on the C/S model, using the UDP protocol?

It's perfectly possible, however, where reliability might be needed, the programmers of that DS would have to incorporate that reliability implemented on UDP.

57. Explain the meaning of acronyms Rtt and RTO in the context of the TCP protocol.

Rtt is the round-trip time and RTO is the Retransmission TimeOut

58. The present value of SRTT in a TCP connection (In the direction A -> B) is 100ms. What time length will TCP program the RTO with?

RTO = 2 x Rtt = 200ms

59. Explain the Nagle algorithm.

When enabled on a socket, Nagle's algorithm prevents the sender from sending any partial segment if some ACK is pending.

60. The transmission buffer of a TCP connection (A FIFO) at a host has the following *blocks* of bytes to transmit:
First: 10 bytes
Second: 1 MSS
Third: 1 MSS
Comment how TCP deals with this situation and what's going to happen next.

Recall TCP doesn't necessarily honor the write boundaries used by the transmitting application; consequently, the question is misleading because the coalescing action of TCP will form a full MSS with the first 10 bytes and the ensuing bytes in the second segment and the third, therefore, the queue will have the following byte blocks:

Head of queue (1st) = 1 MSS
2nd = 1 MSS
3rd = 10 bytes

Assume the Nagle algorithm is active on the socket. TCP can send the first block of MSS bytes in a new full MSS segment and the next. Once these two segments have been transmitted, TCP will transmit the segment resulting form the last 10 bytes, only if all the pending ACKs have been received.

61. Assume a TCP connection has an SRTT value of 100ms in the direction from host A to host B. The next three samples of of the Rtt are the following: {250, 95, 110}. Calculate the resulting SRTT and the value of the RTO if the Karn-Partridge statistics is used.

We can assume all times (Rtt) are given in units of ms (10-3 s)

SRTT[n+1] = a SRTT[n] + (1 – a) * RttSample[n]*

*SRTT[1] = 0,9 * 100*10-3 + 0,1 * 250*10-3*

Proceed through the end of the calculation by applying the same formula recursively. When you have the final SRTT computed, multiply it by 2 to obtain the resulting RTO.
