THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Computing

This is an open-book examination.

(COMP5311)

Internet Infrastructure and Protocols

12 December, 2006 3.5 hours

[Answer at most 7 questions in section A and both questions in section B.]

Section A: Please answer AT MOST SEVEN questions in this section [8 marks each, making up a total of 56 marks out of 100.]

1. (IP addresses) In the classless interdomain routing, each block of addresses is expressed as `n.n.n.n/p`, where $0 \leq n \leq 255$ and $0 \leq p \leq 32$.

   (a) (2 marks) What is the value of `p` for a host-specific route?

   (b) (2 marks) What should be the value of `p` for a point-to-point link (i.e., there are only two interfaces on this link)?

   (c) (2 marks) When `p` = 0, what should be the value of `n.n.n.n`?

   (d) (2 marks) An ISP is allocated with addresses beginning with `10.24.0.0` and ending with `10.31.255.255`. What is the route (i.e., IP address / prefix length) advertised by this ISP?

2. (Measuring delay jitter) This question concerns the measurement of delay jitter between a source node S and a destination node D. Let's first define the delay jitter. From Figure 1, two packets are sent out from S to D, and they are received in the same order. We define the forward delay jitter ($S \rightarrow D$) as $|t_1 - t_2|$.

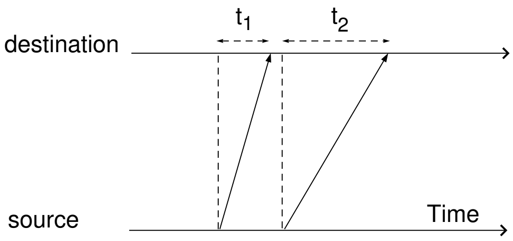

Figure 1: The definition of delay jitter for two packets.

Now consider Bob who is at S attempts to measure the forward delay jitter ($S \rightarrow D$) as well as the backward delay jitter ($D \rightarrow S$). To do so, he sends out two TCP packets at times s_1 and s_2, where $s_1 < s_2$, respectively. Each TCP packet will trigger a TCP packet back to S **immediately**, and each returned TCP packet also includes a timestamp that indicates the time of sending the packet. Let the two timestamp values be d_1 and d_2, respectively. Moreover, let the returned TCP packets are received in order at S at times r_1 and r_2, respectively.

   (a) (4 marks) Based on the times of sending out the TCP packets and the returned TCP packets' timestamps, how does Bob compute the forward delay jitter for the two probing packets?

   (b) (4 marks) Based on the times of receiving the returned TCP packets and the returned TCP packets' timestamps, how does Bob compute the backward delay jitter for the two returned TCP packets?

3. (IP forwarding) Consider the IP network in Figure 2 which is a class B network with a subnet mask of `255.255.255.0`. Therefore, the router R and all the hosts in the four subnets are aware of the subnets (i.e., their forwarding entries are configured with `255.255.255.0`). Moreover, there are no other routers inside the four subnets (the oval). That is, any layer-two broadcast will be heard by everyone on the four subnets. Besides the route to their own subnet, each host uses R as its default router.

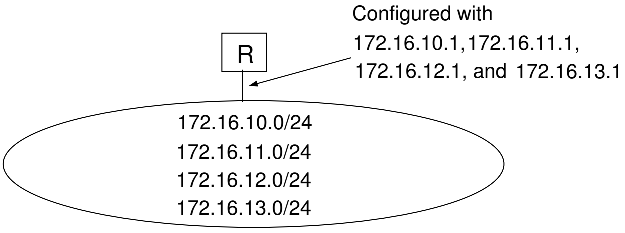

Figure 2: A class B network with subnet mask `255.255.255.0`.

   (a) (4 marks) Under the current configurations, how can a host on subnet 1 send an IP packet to another host on subnet 2?

   (b) (4 marks) Repeat part (a) if we "make" all the hosts unaware of the subnets (i.e., configure their subnet masks as `255.255.0.0`).

4. (IP fragmentation) Recall the tunnel fragmentation example in Figure 3, where the path MTU between LAN A and LAN D is given by $\min\{MTU1, MTU2-20, MTU3-20, PMTU_{2,3}-40, MTU4\}$. To make it simple, assume the MTUs for all the network links are equal to 1,500 bytes; therefore, $MTU1 = MTU2 = MTU3 = MTU4 = PMTU_{2,3} = 1,500$ bytes.

   (a) (3 marks) If a host on LAN A sends out an IP datagram with a total length of 1,470 bytes, where will this datagram be fragmented and why?

   (b) (5 marks) If a host on LAN A sends out an IP datagram with a total length of 1,500 bytes, where will this datagram be fragmented and why?

5. (TCP data transmissions) Recall the definitions for `snd_una`, `snd_nxt`, and `snd_wnd`:

   - `snd_una`: oldest unacknowledged sequence number
   - `snd_nxt`: sequence number of the next segment to be sent
   - `snd_wnd`: size of the send window

   (a) (2 marks) Given `snd_una`, `snd_nxt`, and `snd_wnd`, how many bytes of TCP data have been sent but unacknowledged?

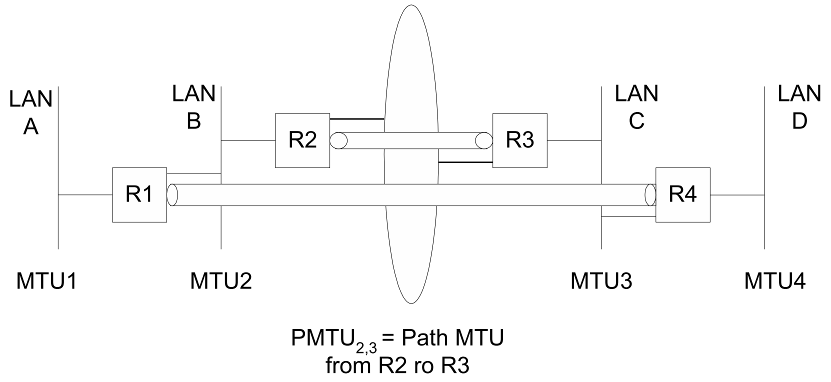

Figure 3: A nested IP tunnel scenario.

(b) (2 marks) Given `snd_una`, `snd_nxt`, and `snd_wnd`, how much of the send window is free, in terms of bytes?

(c) (4 marks) In fact it is possible for the answer in part (b) be negative. Explain why this is so.

6. (TCP congestion control) In Figure 4, we show a TCP data-ACK plot. You may ignore the receiver-side traces (i.e., Rcv_Data and Rcv_Ack). The two sender-side traces are defined as:

   - Snd_Data: TCP data segments sent by the TCP sender
   - Snd_Ack: TCP ACKs received by the TCP sender

   To simulate a packet reordering event, the packet that was supposed to be sent at around 38s (marked as + in the figure) is queued while the succeeding six packets were let through. Then the delayed packet was sent out (indicated by the arrow) with the next packet. There are no packet loss events in this set of traces.

   (a) (2 marks) Around what time was the reordered packet "recovered" (from the sender point-of-view) and why?

   (b) (3 marks) How was the reordered packet "recovered"?

   (c) (3 marks) What happened to the sender's `cwnd` when the reordered packet was "recovered" and why?

7. (TCP and Web) Consider the following HTTP session and we ignore the connection termination phase. T_S is the time that the server spends on setting up the connection and sending the requested document to the client. The requested document consists of only 1 data packet. T_C is the time that the client spends on setting up the connection and receiving the requested document from the server. From the diagram, T_S = T_C = 8 time slots (a time slot is equal the length between two adjacent, horizontal dotted lines).

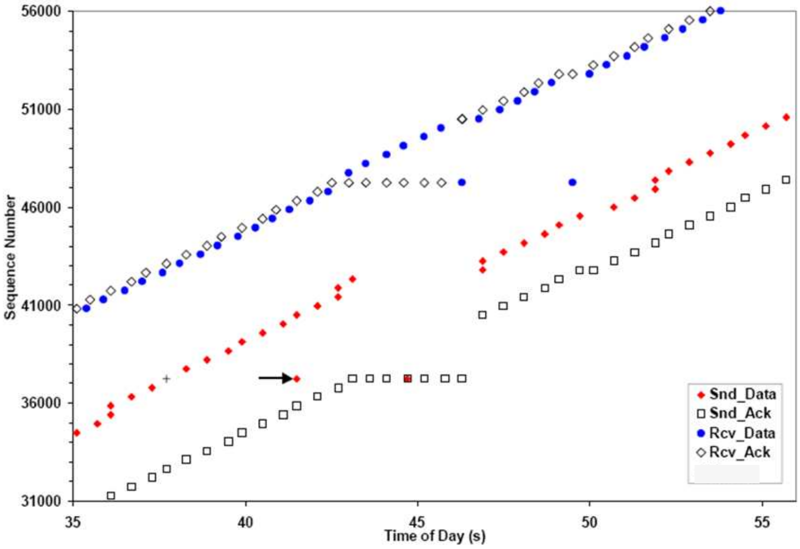

Figure 4: A TCP packet reordering scenario.

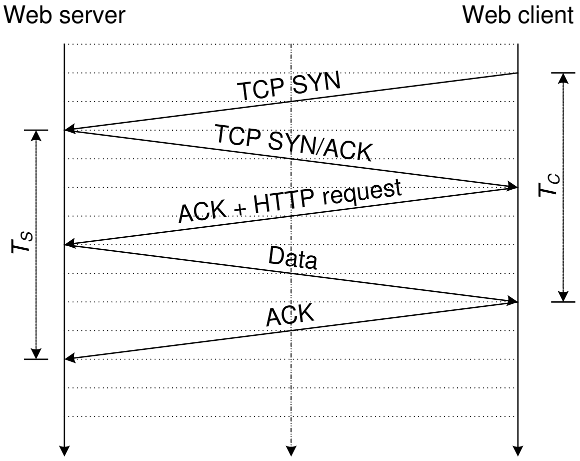

Figure 5: The TCP packet exchange between a Web server and a Web client.

Now, we insert a Web proxy **exactly middle** (at the vertical dotted line) between the server and client; the TCP connection therefore becomes two separate TCP connections (one between the server and proxy, and another between the proxy and client). What are the new values of T_S and T_C in terms of the number of time slots? Assume that the processing time at the proxy is negligible. Moreover, the proxy sends an ACK to the server immediately after receiving the data from the server.

8. (RIP) Consider the network segment in Figure 6 where the routers are running RIP, a distance-vector routing protocol, among themselves. Assume that R_2 and R_3 incur equal costs to reach network A, and that R_1 picks R_2 to be its next hop for network A. Moreover, the split horizon mechanism (with or without poisonous reverse) is implemented by all routers. If the link $L1$ breaks, will there be a routing loop? Clearly explain each step.

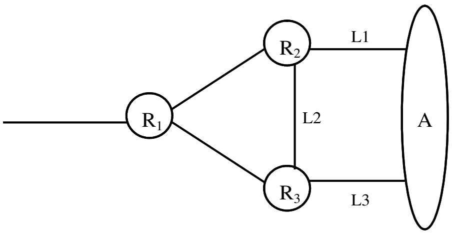

Figure 6: A RIP network

9. (OSPF) A routing protocol developer has proposed a new way of performing "load balancing" on the network links in OSPF networks. Briefly, for each destination, an OSPF router comes up two best routes with different next-hop routers, which do not necessarily have equal costs, and it forwards the traffic onto these two routes proportionally. For example, if one route has a cost of $C1$ (route 1) and the other has a cost of $C2$ (route 2), it will forward $C2/(C1+C2)\%$ of the traffic using route 1 and the remaining traffic using route 2.

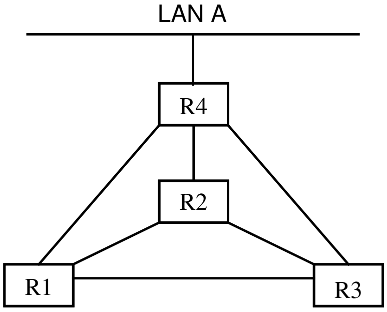

Figure 7: An OSPF network.

However, another developer has pointed out a serious flaw in this mechanism. What is it? Use the network in Figure 7 to illustrate the flaw with the following scenario. All the link costs are equal to 1.

- $R1$ can reach LAN A via $R4$ with cost = 1 (route 1) and via $R2$ with cost = 2 (route 2).
- $R2$ can reach LAN A via $R4$ with cost = 1 (route 1) and via $R1$ with cost = 2 (route 2).

Section B: Please answer both questions in this section. Each question carries 22 marks.

10. (Initiating communications into a private IP network) Recall that three blocks of IP addresses have been allocated as *private addresses*. These special addresses can be used only **within** an autonomous system (AS), such as the PolyU network (i.e., not routeable between ASes). Therefore, a private address, whether in the source or destination address field, has to be translated into a public IP address before being forwarded outside an AS. The devices that are responsible for translating the addresses are known as *network address translation* (NAT) routers. We use IP and ip to denote a public address and a private address, respectively.

    Figure 8 illustrates how NAT works. The left network is a private network; the right is a public network. Host B (Bob) with private address ip_B sends a packet to host A's (Alice's) public address IP_A. The NAT router R receives the packet and then translates the source address into IP_R, the router's external, public address before forwarding it. As a result, A believes that she is talking to a host with address IP_R. Since R keeps an address translation table, he can translate the destination address in A's return packet before forwarding it to B.

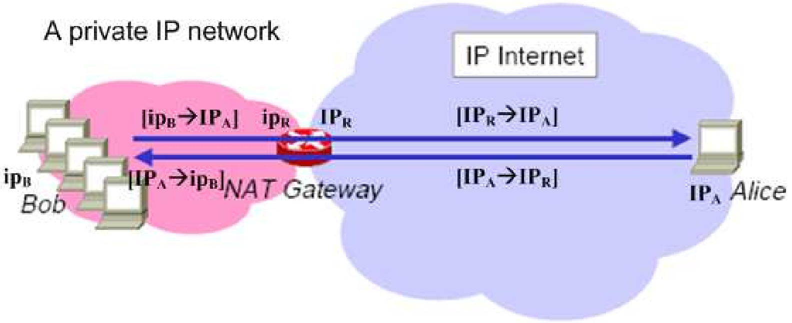

Figure 8: An example to illustrate the NAT operation.

Note that in the example above the private network initiates communication with a public network. However, a public network cannot initiate communication with a private network, because the normal DNS servers do not resolve domain names into private addresses and private addresses are not nonrouteable. In this question, we consider a scheme that can solve this problem. Consider Figure 9 for an illustration. In this scheme, there are three main entities:

- W is an agent router with a public address IP_W.
- R is a NAT router on the private network.
- A special DNS server which is not shown in the figure.

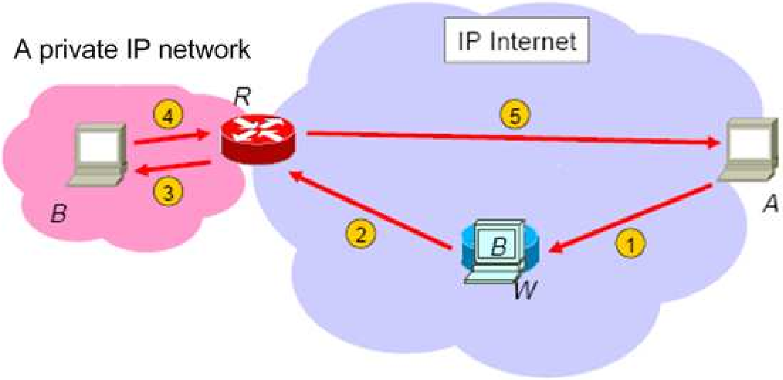

Figure 9: The five steps for A to initiate communication with B.

In the following, we explain how A's first packet is delivered to B and how B's reply packet is sent to A, as depicted in Figure 9:

0. (This step is not shown in the figure) A first contacts a special DNS server to obtain B's address; the server subsequently returns IP_W to A. Moreover, the DNS server also notifies W of the binding information (e.g., IP_R, IP_A, ip_B) for W to carry out step 2.
1. After receiving the DNS reply, A sends a packet to B: [$IP_A \rightarrow IP_W$] (notation: [source IP address, destination IP address]).
2. When W receives this packet,
   - It translates the packet into [$IP_A \rightarrow ip_B$] (i.e., change the destination IP address to B's).
   - Tunnel the translated packet to R; the tunneled packet is represented by // [$IP_W \rightarrow IP_R$ [$IP_A \rightarrow ip_B$]].
3. When R receives this tunneled packet,
   - Create a translation table entry for this packet, if have not done so ($ip_B \rightarrow IP_W$).
   - De-tunnel the packet and send it to B.
4. B sends a reply packet [$ip_B \rightarrow IP_A$] back to A.
5. When R receives B's reply packet, it translates the packet into [$IP_W \rightarrow IP_A$] before sending it out.

Please answer the following questions regarding this scheme:

(a) (2 marks) In step 2, after W translates the destination address in A's packet to B's address, what other IP header field(s) needs modification and why? If the IP payload is a TCP packet, what other TCP header field(s) needs modification and why?

(b) (2 marks) In step 3, what is the purpose for R to create a translation table entry when receiving W's packet?

(c) (3 marks) Suppose that the original packet $[IP_A \rightarrow ip_B]$ is too big for the MTU of the W-R tunnel; moreover, the don't fragment bit is turned on. Therefore, a router on the W-R path will drop the oversized packet and return an ICMP error message. Will A be able to receive this ICMP error message?

(d) (3 marks) Suppose that when B receives A's packet, it returns an ICMP error message (e.g., port unreachable). Will A be able to receive this ICMP error message?

(e) (2 marks) In step 5, besides translating the source address, what else does R perform on the IP packet and its payload which is a TCP packet?

(f) (2 marks) In this scheme, is B aware of W? Explain your answer.

(g) (5 marks) So far we have assumed that R is the only border router in the private network. That is, all the inbound and outbound traffic must go through it. However, if there are other border routers, B's reply packet may not go through R. In this case, the address translation cannot be performed correctly. Propose a simple scheme to ensure that R will receive B's reply packet. Note that R has a private address ip_R.

(h) (3 marks) Note from step 5 that R essentially spoofs IP_W; this spoofed packet, however, may be dropped by a firewall. Propose a simple scheme that would allow R sends this packet back to W.

11. (TCP over a long pipe) In this question, we consider the performance issues when TCP is used over a network that has a long propagation delay. In particular, we consider that a sender transmits TCP packets over a terrestrial link and then a satellite link to a receiver. We denote the terrestrial link's round-trip time (RTT) and the satellite link's RTT as T_t and T_s, respectively; normally, T_s is much larger than T_t. The overall RTT is therefore $T_t + T_s$. In Figure 10, we show an example that it takes three RTTs to transmit seven TCP data packets to the receiver. To make it simple, we assume that it takes no time to transmit data packets and ACKs; therefore, we can use a single line for each window of packet transmissions and similarly for the ACKs.

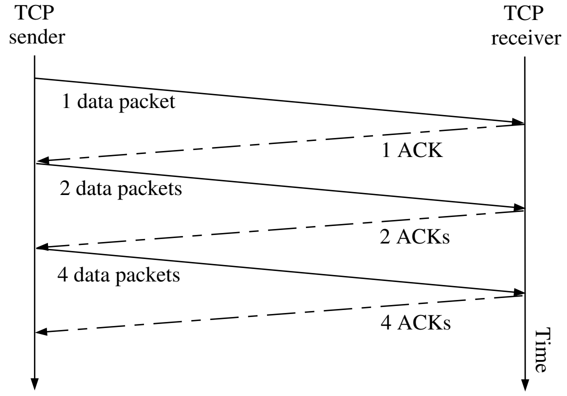

Figure 10: It takes three RTTs to send seven TCP data packets.

There are several well-known problems connected to "TCP over satellite." In this question, we concentrate on the problem related to the large RTT. Recall that a TCP sender's `cwnd` is doubled after every RTT (if the receiver acknowledges every data segment). Therefore, a longer RTT will decrease the sending rate. One approach to speeding up the TCP's throughput is to put a TCP proxy between the terrestrial link and the satellite link. Similar to the case of Web proxy, this TCP proxy splits the TCP connection into two, as depicted in Figure 11.

(a) (2 marks) Does this TCP proxy solution help speed up the data transmissions as compared the one without the TCP proxy? You may use the seven-packet transmission to explain your answer.

An engineer proposes a seemingly better TCP proxy to increase the connection throughput. The approach is to let the proxy to send early ACKs to the sender. Figure 12 illustrates the approach.

(b) (2 marks) Using this new TCP proxy, how much time does it take the sender to transmit seven TCP data packets?

(c) (2 marks) What is the major factor responsible for the throughput increase with the new TCP proxy?

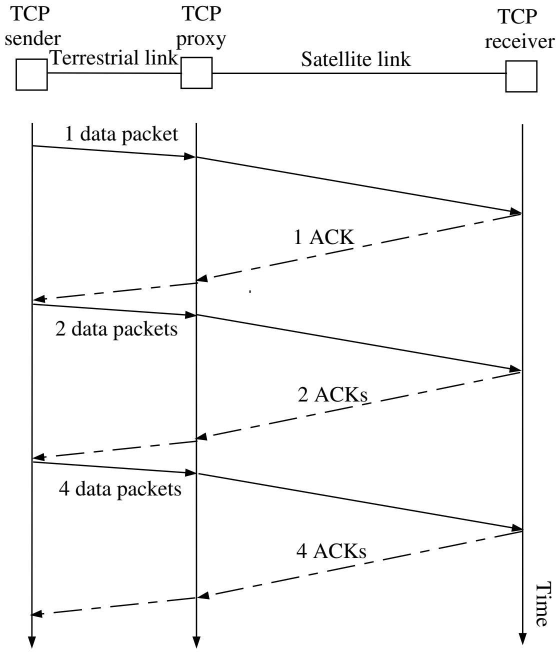

Figure 11: Sending seven TCP data packets over a terrestrial link and a satellite link via a TCP proxy.

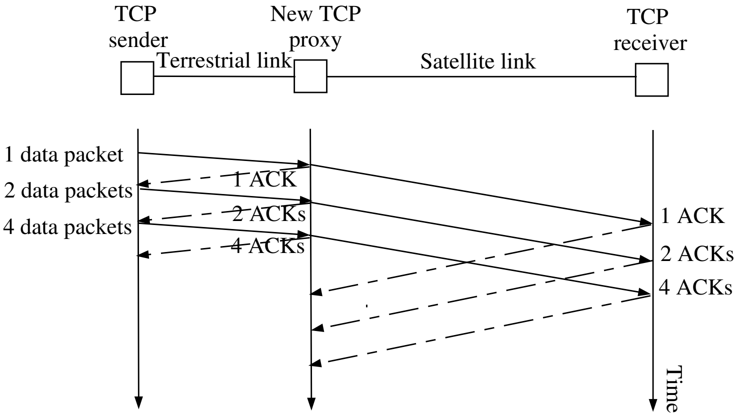

Figure 12: The new TCP proxy responds with early ACKs as soon as it receives the data packets.

(d) (2 marks) There are different data buffering requirements for the two TCP proxies. The new TCP proxy has to store the data packets until it receives the ACKs. How about the old TCP proxy?

Now let's look at some detailed design issues for the new TCP proxy. As the first design, we try to make the TCP proxy as simple as possible. Therefore, we do not implement the congestion control and flow control schemes in the proxy. That is, it does not look at the receiver's advertised window and does not have the `cwnd`. However, it still performs either timeouts or fast retransmit for packet losses. Therefore, the proxy continues to send data as long as in-sequenced TCP data are available. Furthermore, if the proxy retransmits nth packet, it will continue to send the subsequent packets ($(n + 1)$th, $(n + 2)$th, etc) even they have been sent before.

(e) (2 marks) Recall that a regular TCP sender keeps three main state variables: `snd_una`, `snd_nxt`, and `snd_wnd`. Does the proxy still need to keep these variables and why?

(f) (2 marks) Recall that a regular TCP receiver keeps two main state variables: `rcv_nxt` and `rcv_wnd`. Does the proxy still need to keep these variables and why?

(g) (4 marks) Suppose that the proxy has finished sending 20 data packets and has no more data to send; it has received the ACKs for the 15 packets. After that the proxy receives five consecutive window updates with 0 bytes in their advertised window fields. How would the proxy respond to these zero window updates? (Hint: A TCP sender should also process the ACK values in the window updates.)

Now we turn to the TCP connection opening and TCP connection closing phases. To simplify it, we consider that only the TCP sender in Figure 12 performs active open and active close. There are again two possible ways for the proxy to handle the opening and closing. For the connection opening, the proxy can either

- return the SYN/ACK to the sender immediately after receiving the SYN from the sender, or
- return the SYN/ACK only after receiving the SYN/ACK from the receiver.

The connection closing has the similar choices.

(h) (4 marks) Which way should the proxy use for the connection opening and connection closing, and why?

— End of the Examination Paper —