Q1 was taken by 8 students only. The average was 59%, but, given the small sample, the standard deviation was high, at 22.6. There were no systematic misunderstandings. The only point to note is that item (f), on web services, was poorly answered by all.

Q2 was taken by 147 (i.e., almost all the) students. The average was high, at 71.6%, with the standard deviation at 23.1. There were no systematic misunderstandings. The only point to note is that there is confusion as to what a stateful/stateless protocol is. Too many students wrongly believe that a stateful protocol is one in which a connection is held (in some sense, blocking other processes from connecting), but this is a simplistic view. In fact, as the name indeed implies, a stateful protocol is one in which answering a message can force the participants to keep state, e.g., remember a message, thereby giving rise to the notion of a per-client session.

Q3 was taken by 149 students (almost all the class). The average was probably lower than we hoped for, but not disastrous (except for some students who clearly came to the exam unprepared). The questions were not easy and required some insight into several aspects of VoIP. There were a few students who were completely clued up about this subject and wrote excellent answers, though nobody got 100%. Common misunderstandings that cost marks were that increasing the buffer sizes (part d) would solve the overflow/underflow problem due to inevitable mismatches in the speech sampling rates at either side of a VoIP call. Remembering that the world's rotation is slowing down, and the nature of leap-seconds, gives a strong clue to the correct answer, but many students were, perhaps too depressed by this analogy. Increasing the buffer sizes (at BOTH ends by the way) only provides temporary relief to the synchronisation dilemma (or the end of the world). Many people talked about discarding (at the slow end) and introducing (at the fast end) whole 'packets' to maintain the balance. This is like waiting 400 years to adjust our atomic clocks and introducing leap-hours which we would certainly notice even if it was done in the middle of the night. A packet contains 160 samples and you cannot wait until you are a whole packet ahead (or behind) before discarding (or introducing) voice samples. Only a few students explained how to decide when to make an adjustment, by monitoring the contents of the buffer. This problem beautifully illustrates a very important aspect of real time programming that students will certainly meet again, for example in the second year Mobile Systems course COMP28512. Many students preferred the leaky bucket analogy and answered part (b) reasonably well despite the fact that it is quite hard. Remember that it is not a dripping tap (the CPU has a good plumber). The tap is turned fully on or off when the CPU chooses (within reason). The regular drip, drip drip from the hole in the bottom, (say 8000 drops per second) feeds the D to A converter at a constant rate. The A to D converter drips its voice samples into the bucket again at a constant rate. Remember that 'jitter' is 'variation in network delay' which affects the time instants that packets arrive. The concept of a 'jitter buffer' (part e) keeping a 'reservoir' of packets (yes now packets) at each end of a VoIP link is not easy to describe, but many people made a reasonable job of this.