**Comments:**

Q 1
a)This part was generally answered correctly
b)Most answers for this part were correct. However, a number described scalability in relation to a network and not an application; the question explicitly indicated that the content was a network applications.
C)This part explicitly asked about the coordination of information; it was not about access to information, which was the next part. Many answers described the DNS lookup process. This does not ensure the world-wide uniqueness of names; i.e. how the name space hierarchy is managed. Some answers also described the updating of secondary servers, this process only coordinates local information.
D)A significant number of answers indicated that the way to distribute load in the web is CDN. However, CDN do not encompass all web based applications. In particular, they are about distributing the load of accessing information; they make no attempt to address of issue of load associated with managing or updating information. Many answers failed to mention that one of the simplest ways of spreading load is caching.
E)As with the previous part, claiming that a CDN would solve the problem, failed to address how the load of updating the information is distributed. A number of answers proposed some mechanism to spread the load over multiple servers by distribution the information across multiple servers. However, it was not always clear how the load on the central server to find the server holding required information was managed. Many answers described the operation of the server side of the architecture, but omitted the client-server interactions. Finally, almost all answers failed to described the need for a globally unique identifier/key for all users; user's names would not be guaranteed globally unique.

Q 2
a)Answers lacked the clarity that IP does host-to-host communication and that UDP/TCP to process-to-process communication. The identification aspect also lacked the clarity that IP uses a host (IP address) and that UDP/TCP use a port number.
B)Many answers described how ARP can be used to get the IP address of a remote host with which a local host wishes to communicate. However, there is more fundamental information a host requires, e.g. its own address, the address of a local gateway and a DNS server, that a host requires before it can participate in any communication. This fundamental information was not listed by many answers.
C)Generally there was a slight lack of detail in the answers given.
D)Several answers showed confusion between the two recovery approaches. A significant number of answers failed to address to comparison part of this question.
E)Although a number of good answers to this part based on a three-way handshake where produced, a significant number proposed solutions that depended on public/private keys. In the description of the scenario, no public/private keys are available, so such solutions do not address the scenario situation. A number of the proposed solutions failed to address the 'maximise the security of the shared key' aspect of the question by not minimising its use.

Q3.a
This could be interpreted in two valid ways. Firstly a simple request for DNS and ARP as the examples or as a request for DNS and some other protocol to provide a valid application port number. Due to this, provided DNS was chosen some other protocols such as IP, TCP, UDP and in some cases NAT were allowed for the second protocol. Shockingly, some students did not even provide DNS as an answer.
Q3.b
For the simple DNS + ARP case the issue is that ATM does not support broadcast. Therefore the ATM systems must be configured with the address of an ARP provider; this was hard but a few students did remember it being mentioned.
In the DNS + other case no major differences will occur. There may be differences in performance plus of course the frames sent are ATM rather than Ethernet. A large number pointed this out though some left the answer blank.
Q3.c
Most had little problem with this. A few seemed to confuse routers with switches. Some forgot that switches often have two sides, one external facing connected to other switches and routers and the rest for directly connected hosts. Only the switch is dead, the rest of the network may continue to function. Directly connected hosts still work but have no

network connection.
Q3.d
The vast majority knew the answer to this but some answers were too terse/concise. The task is easy but the technician does have to move all the connecting wires. Several answers were ambiguous so less than the full 4 marks were awarded.
Q3.e
There were lots of very good answers to this. Too many people used it as an opportunity to unload all there knowledge of CSMA/CD and CA often without clearly defining the differences. CSMA/CD is no longer needed in most Ethernets as wires are not shared by multiple hosts and are full duplex.
The key word to mention here was "share" which was often missed out. Other answers only discussed wired XOR wireless but not both.
I was shocked how many students stated categorically that collisions in wireless are due to the use of electromagnetic waves in wireless. A few even stated that electromagnetic waves are not used in wired communication, though more of you implied this.
Q3.f
This was mainly quite weakly answered. Collision avoidance is a key issue for wireless networks, however, many clearly miss the point of the long discussion at the start of the wireless material about licensing and different sharing/multiplexing methods. Protocols such as CSMA/CA etc. only apply once all these other collision avoidance ideas have been applied.
Q4.a
This was mainly answered OK but with insufficient detail/analysis for 7 marks. In a lightly loaded network it is assumed the VoIP and other applications all work mainly OK. But bursts of traffic will cause significant jitter (variation is delay) partly because TCP frames are big and often occur in bursts with longish gaps in-between.
In a congested network all services will suffer from delays, larger jitter. The delays will vary due in part to TCP slowing down when it detects congestion and due to the bursty nature of most non constant bit rate traffic.
Q4.b
This was supposed to be quite easy but most answers made a mess of it by either failing to read the question properly, not sorting out the meaning of the phrases, not applying common sense or just getting the simple arithmetic wrong. Far too many answers did the arithmetic by some magic means showing little or no working just wrong answers.
Most assumed there was an exact answer when there is not! This is because most attempts simply ignored the ordinary best effort traffic (presumably mainly TCP).
The sums if done correctly produce a number of calls greater than the maximum number you were told (up to 30). The key is that everything is "best effort" throughput demand for calls, traffic and (not stated but may stop/start) the video all varies. Each video and each call therefore uses shared resources, the left over bandwidth for other traffic being very small with video + 30 calls. In practice, 30 calls is probably too many.
Luckily, most of the marks were for the maths and for the method of analysis used not the answer.
Q4.c
Firstly the question asks for a diagram so one must be provided. For top marks it needed to show queues, how the fairness is achieved, merged output etc. . For 8 marks quite a lot of detail is wanted in the diagram and the associated explanation.
The traffic needs to be split into classes to which different service is applied. Several answers suggested use of a Round Robin queue to merge these different classes to the output. Round robin gives equal shares to all so is not suitable without modification.
Some answers suggested a priority queue with video and/or calls given priority over best-effort traffic. This can help but does not work well in practice because it is inherently unfair only forwarding the highest priority frames whenever some of these are waiting in a queue. The best (known) queuing is Weighted Fair Queuing which will ensure every class gets a share of the output stream according to its weighting assignments. WFQ works well but on its own does not handle variation in the arrival rate of frames very well. Too few arrivals wastes sending opportunities and too many arriving frames may block with no allowance for previous low usage. Therefore, a token bucket is used to police each class of data controlling the mean and peak rates the flow can achieve.
Answers without splitting of flow classes were not good. Not showing queues and token buckets in the diagram produced lower marks. Only 1 or 2 answers attempted to label or show queue weights and token bucket size, token rate etc. .