EXAM Telematics Networks (192620000)

29 October 2013, 13:45–17:15

- This is an open-book exam: you are allowed to use the book ("Computer Networking" by Kurose & Ross), the reader, copies of the lecture slides, and a dictionary. Use of a simple (non-graphical) calculator is allowed (but not needed).
  Use of other written material, such as your own notes, is not allowed, nor is the use of laptops, notebook computers, graphical calculators, mobile phones, etc. ***Please remove any such material and equipment from your desk, now!***
- You are allowed to answer in either English or Dutch.
- You should always explain or motivate your answers, with so much detail that the grader can judge whether you understand the material; so just saying "yes" or giving a formula without explanation is not enough.
- Besides the exam, you are also given a questionnaire about the course. Please do fill out that form, and hand it in when leaving the room. Of course, you may fill out the questionnaire after handing in the exam answers, so the questionnaire doesn't cost you time that would be better spent on the exam itself.
- Note that your exam will not be graded unless/until you have also completed the *Wireshark* assignment, and that 10% of your final grade will be determined by the homework multiple-choice questions.

1. Information and communication theory

2 pt (a) We have a formula for the Shannon capacity of the (memoryless) analog channel. This allows us to calculate, for a given noise level and bandwidth, the signal strength needed for a given information rate.

Does this mean that a real system can only work when the signal strength is *precisely at* that level, is *above* that level, or is *below* that level? Explain.

3 pt (b) Assume we have a binary channel of 1000 bits/s (raw speed, not Shannon capacity), with the property that all odd-numbered bits (i.e., the 1st, 3rd, 5th bit and so on) are always received correctly, while the even-numbered bits have a 10% probability of being received incorrectly. Calculate the (Shannon) capacity of this channel.
(Give a formula if you don't have a calculator with you.)

3 pt (c) Consider a random IP packet intercepted somewhere in the internet. How much information (in the Shannon-sense) is there in the *checksum* field of this IP header? Does this depend on whether we also look at (i.e., know) the contents of the other fields of the IP header, and if so, how? Explain.
(Note: you may or may not be able to give a precise number for the amount of information; if not, give an indication of the order of magnitude and what it depends on.)

3 pt (d) The IP header checksum is normally used only to *detect* errors, but could it also be used to *correct* single-bit errors? Explain.

Continued on next page...

2. Faulty middleboxes

A "middlebox" is a device sitting somewhere on the path between internet hosts, which does something more than just forwarding packets as a normal router should; examples of middleboxes are firewalls and NATs. Middleboxes may do undesirable things to your packets, like modify them or drop them inappropriately, either due to bugs in the software of the middlebox, or due to excessive paranoia of the designer.

In this problem, we will consider a middlebox that sits between a single host (let's call it "our host") and the rest of the internet. You are asked, for each of the middlebox behaviours given below, to tell what consequences this has (for example, "outgoing TCP connections will fail", or "the host will catch fire"), and explain how/why that happens. Where applicable, be careful to distinguish between (outgoing) connections set up by our host to another host somewhere on the internet, and (incoming) connections set up by another host to ours.

A few remarks:

- The words 'incoming' and 'outgoing' are from the point of view of "our" host.
- Unless otherwise specified, IPv4 is assumed.
- Whenever the middlebox modifies a packet, it also recomputes checksums that cover the modified part of the packet.
- Apart from the described behaviour, the middlebox simply forwards the packets as it should.
- It *may* be that the action of the middlebox is innocent, i.e., has no consequences that the endhosts or users notice; in that case, say so, and explain why it is innocent.

2 pt (a) In every incoming TCP packet, the middlebox sets the ACK flag to 0.
What are the consequences? Why?

2 pt (b) In every incoming IP packet, the middlebox sets the last bit of the checksum field to 0.
What are the consequences? Why?

2 pt (c) In all outgoing TCP SYN packets that do not have the window scaling option, the middlebox inserts a window scaling option with a scaling factor of 8.
What are the consequences? Why?

2 pt (d) In every incoming TCP packet, it sets the CWR bit to 0.
What are the consequences? Why?

2 pt (e) The middlebox changes the version field of all IPv4 and IPv6 headers: it replaces the value 4 by the value 6, and the other way around. (Assume that both our host and the rest of the internet are capable of both IPv4 and IPv6.)
What are the consequences? Why?

Continued on next page...

3. Addressing

2 pt (a) Why do network-layer addresses usually have hierarchical structure?
In other words: why can't we just assign them randomly (while still making sure they are unique, of course) ?

IPv6 has 128-bit addresses. In the most common IPv6 address assignment scheme, the last 64 bits of this are used to identify the host within a network.

2.5 pt (b) What is the advantage of reserving so many bits for identifying the host within the network, even though no network will ever have anywhere near 2^{64} hosts?

2.5 pt (c) This addressing scheme with 64 bits for identifying the host and the other 64 bits for identifying the network looks suspiciously like the old "class B" addresses from IPv4.
Should we therefore expect the same problems that plagued IPv4 with class A/B/C addresses (and were resolved by introducing CIDR) to re-occur with IPv6? Explain.

Course codes (like 192620000 for this course) can be considered addresses: they "address" individual courses within the total set of courses offered by the university. (In fact, just like IP addresses, course codes might well have a hierarchical structure, with e.g. some digits indicating the faculty, some for the study program, etc.)

3 pt (d) Until a few years ago, the UT used course codes of 6 digits. Use the *host-density ratio* to judge whether the change from 6 to 9 digit codes was needed.
(If you need to make estimates or assumptions, state them.)

---

4. Cheating with TCP congestion control

1.5 pt (a) Explain in your own words what the "slow start" phase of TCP congestion control is.

Someone found out that Google "cheats" a bit with the TCP specification. He concluded this from observing that the RTT between him and Google is 20 ms, and that loading an entire 8 kB webpage effectively only took 40 ms, using 1.5 kB segments.

2.5 pt (b) Why is this impossible with standard TCP?

Someone else designed the "optimistic ack" cheating method: let the receiver send acks for segments that it has not yet received.

2.5 pt (c) Explain why this increases the rate at which data is sent to the receiver, assuming the sender follows the standard TCP congestion control algorithm, and assuming no packets are lost.

2.5 pt (d) Would this "optimistic ack" method also work on a "long fat pipe" with the sender running TCP-CUBIC ?

3 pt (e) If the round-trip-time is long, and the receiver does not go too far ahead with its ACKs, the sender will never see ACKs for data that it has not yet sent.
Design a solution (possibly an extension to the TCP protocol) which enables the sender to detect even in those circumstances that the receiver is cheating (i.e., is acking packets that is has not yet received).

Continued on next page...

5. Real-time traffic

Assume we have a data source whose packet flow can be described by a leaky-bucket model with rate r tokens/second and bucket size B tokens (with 1 token per packet).

2 pt (a) If the source sends packets at constant intervals, how much time should there be between two consecutive packets to still obey that leaky-bucket model? Is this a minimum or a maximum?

Suppose this source sends a maximum sized burst, then keeps quiet for 2 seconds, and then sends a burst consisting of k packets.

3 pt (b) Give an expression for the maximal allowed size k of that second burst, in terms of r and B.

Now assume this source, sending a burst of k packets every 2 s, with each packet containing S_k bits, shares a link in the network with another source which sends a burst of ℓ packets every 2 s, with each packet containing S_ℓ bits. The link speed is R bits/sec.

3 pt (c) Suppose we are given a choice between using Round Robin or Fair Queueing scheduling at this link. Which choice minimizes the delays for "our" packets? (Note that your answer may depend on the parameters, like "choose RR if $r = B$ and FQ otherwise".) Explain your answer.

3 pt (d) If FIFO scheduling is used, what is the maximum possible delay for "our" source (the one sending bursts of k packets), in terms of the given parameters?

End of this exam.