EXAM Telematics Networks (262000)

20 January 2009, 13:30–17:00

- This is an open-book exam: you are allowed to use the book ("Computer Networking" by Kurose & Ross), and the reader, and copies of the lecture slides. Furthermore, you are allowed to use a dictionary.
  Use of other written material, such as your own notes, is not allowed, nor is the use of laptops, notebook computers, PDAs, mobile phones, etc. ***Please remove any such material and equipment from your desk, now!***
- Although the questions are only written in English, you are allowed to answer in either English or Dutch.
- Copying pieces of text literally from the book, reader or slides is not allowed: such answers are worth 0 points. You should compose your answers yourself.
- You should always explain or motivate your answers; e.g., just the word "yes" or a number is not enough.
- This exam consists of 5 problems on 4 pages.
- 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!
- Note also that 10% of your final grade will be determined by the homework yes/no questions.

1. Transition to IPv6

(a) What does tunneling IPv6 over IPv4 mean, and why is it useful?

(b) Could the reverse, i.e. tunneling IPv4 over IPv6, be useful too? Why not, or what for?

Originally, a set of IPv6 addresses had been set aside as "IPv4-compatible IPv6 addresses"; this was the range 0::0/96.

(c) What does this notation 0::0/96 mean? More precisely: what range of addresses is denoted by it?

The idea was that 0::0/96 could be used to automatically give any host that has an IPv4 address, a trivial way to also get a usable IPv6 address. This goal is very similar to the goal of 6to4, which was developed later.

(d) What is the most important difference between this 0::0/96 proposal and 6to4?

One nice thing about 6to4 is that its tunnels can be set up automatically.

(e) Would this also be true for the 0::0/96 proposal? How, or why not?

Continued on next page...

In order for IPv6 addresses to be practically usable, applications need to be able to find them on the basis of a host name. This can be done by adding so-called AAAA records to the DNS, so that for example a DNS query for `www.google.com` returns both an A record containing its IPv4 address, and an AAAA record containing its IPv6 address. In reality however, Google's DNS servers do not return AAAA records.
Very recently, Google has started returning AAAA records, but in a somewhat weird way: their DNS servers now look at the IP address from which the query comes (usually IPv4, since DNS is still mostly IPv4 even in networks that also already support IPv6), and only if that address is on a "white list", the answer contains both the A and AAAA records, otherwise only the A record. To get their IPv4 addresses onto this "white list", network administrators have to send an e-mail to someone at Google.

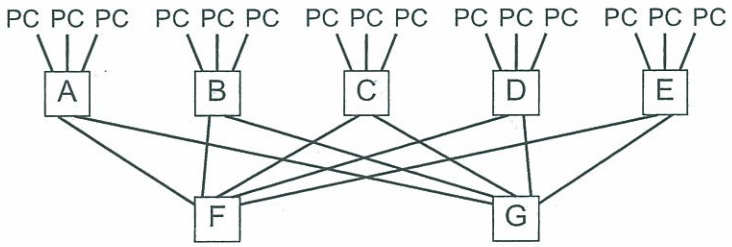
(f) Why do you think Google does it like this? Why not simply always give also the AAAA records?

(g) Is this approach scalable?

---

2. Bridges

Consider this network, connecting 15 PCs, using 7 bridges (or switches) labeled A through G, which use the spanning-tree protocol. All links are 100 Mbit/s ethernet.



Note that the system administrator still has to assign the bridge IDs, and thus determine which bridge becomes the root.

(a) Give a choice for the root bridge such that *not* all traffic takes the shortest path; explain.

(b) Give a choice for the root bridge such that all traffic *does* take the shortest path; explain.

(c) In your choice in the previous question, some links and even an entire bridge are not used. Why is it still useful to keep them in the network?

(d) How could multicast traffic be handled in a network consisting of bridges?

(e) How could IPv6 traffic be handled in a network consisting of bridges?

Continued on next page...

3. QoS

One important goal of the design of ATM was supporting real-time traffic (requiring "Quality of Service"), which was also the goal of IntServ and DiffServ.

(a) Is ATM's architecture more similar to IntServ or to DiffServ? Explain.

(b) Do scheduling algorithms play a role in the QoS provisioning of ATM? Explain.

One concern in the provisioning of QoS is to ensure that a malicious user cannot destroy the quality of other users' service.

(c) In a DiffServ system, how can this be ensured? Explain.

SONET was originally designed for telephone traffic and providing it with a guaranteed QoS.

(d) Do scheduling algorithms play a role in the QoS provisioning of SONET? Explain.

Nowadays, SONET is often used as a physical layer for IP.

(e) Does this mean that if you want to upgrade your existing best-effort IP network to support QoS using IntServ or DiffServ, this is easier to do if the physical layer is a SONET network, instead of say point-to-point full-duplex ethernet links? Explain.

---

4. The TCP timestamp option

One of the options introduced in TCP is the "timestamp" option, which serves two purposes.

(a) Explain how the timestamp option can be useful for improving or simplifying the measurement of the round-trip-time.

(b) Explain how the timestamp option can be used to protect against wrapped sequence numbers.
Be precise: describe exactly a rule (i.e., an algorithm) that the receiver can follow to decide whether or not to accept an arriving TCP segment. Note that the answer is not in the study material; you are asked to design this algorithm yourself (and probably come up with something similar to what is in the relevant RFC).
For efficiency reasons, your algorithm should not require the receiver to store much information (e.g., a list of all sequence numbers and corresponding timestamps is not allowed). Also, be careful to ensure that if there is some reordering without excessive delays, segments are not ignored unnecessarily.

The timestamp option has a 32-bit field for storing the time. However, the unit to be used (e.g, seconds or milliseconds) is not specified: it's up to the implementor to decide what unit to use. As a consequence, the two sides of a TCP connection may use different units.

(c) Why doesn't this cause problems for the RTT measurement?

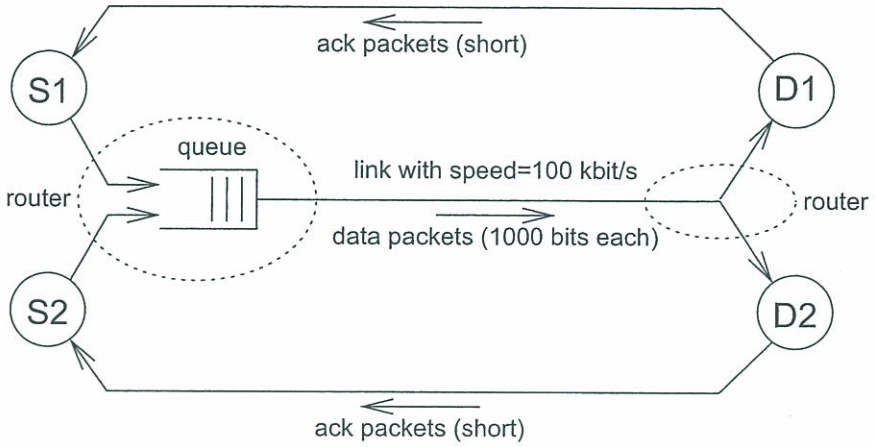
(d) Why doesn't this cause problems for the protection against wrapped sequence numbers?

(e) Considering the two purposes, and the 32-bit size, would *nanoseconds* be an appropriate unit? Or *milliseconds*? Or *seconds*? Explain!

Continued on next page...

5. Congestion control

Consider the network sketched below, consisting of two sources (S1 and S2) and two destinations (D1 and D2), with S1 sending data to D1, and S2 to D2. S1 and S2 have an unlimited amount of data they want to send; and D1 and D2 process any incoming data immediately, and always advertise a receive window of 20000 bits. All links for which no speed is indicated, may be assumed to be extremely fast.



Two routers are used as indicated, so the data packets share a link and a queue (which uses FIFO scheduling). We assume (for simplicity) that the acknowledgements from D1 and D2 to S1 and S2 have independent paths through the network, as indicated. Furthermore, all propagation and processing delays are negligible, so only the queueing and transmission delays of the data packets play a role (while for the acks, even those are negligible). Finally, the header lengths may be neglected here for simplicity, so you may assume that the 1000 bit packets really are 1000 data bits when discussing throughputs.

At first, let us assume **S2 is not sending anything**, so effectively there is only data going from S1 to D1, and the buffer size (i.e., maximum queue length) is just **1 packet**.

(a) If standard TCP (i.e., Reno) is used, what will happen? Discuss at least the throughput, the window size, and the queue length.

Next, assume that **both S1 and S2 start** sending at the same time, and the buffer size is **infinite**.

(b) If standard TCP is used, what will happen? Discuss at least the throughput, the window size, and the queue length.

(c) Would installing ECN be useful in this situation? Why or why not?

Next, assume that source S1 is sending without any form of congestion control at a fixed rate of one packet every 20 ms, and that source S2 is using a TCP-Vegas-inspired algorithm to set its window: it will (gently) increase or decrease its congestion window such that it measures a round-trip time of about 60 ms.

(d) What will happen? Discuss at least the throughput, the window size, and the queue length.

End of this exam.