CS5565 Sample Midterm Spring 2005

Answers are shown in this style.

1 DNS (10 pts)

Suppose your local DNS resolver needs to resolve www.kernel.org on behalf of a client. Suppose it has already sent a query to a root server at 198.41.0.4 and received the following reply:

```
; <<>> DiG 9.2.3 <<>> @a.root-servers.net +norecurse www.kernel.org
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31394
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 6

;; QUESTION SECTION:
;www.kernel.org.                        IN      A

;; AUTHORITY SECTION:
org.                    172800  IN      NS      TLD1.ULTRADNS.NET.
org.                    172800  IN      NS      TLD2.ULTRADNS.NET.
org.                    172800  IN      NS      TLD3.ULTRADNS.org.
org.                    172800  IN      NS      TLD4.ULTRADNS.org.
org.                    172800  IN      NS      TLD5.ULTRADNS.INFO.
org.                    172800  IN      NS      TLD6.ULTRADNS.CO.UK.

;; ADDITIONAL SECTION:
TLD1.ULTRADNS.NET.      172800  IN      A       204.74.112.1
TLD2.ULTRADNS.NET.      172800  IN      A       204.74.113.1
TLD3.ULTRADNS.org.      172800  IN      A       199.7.66.1
TLD4.ULTRADNS.org.      172800  IN      A       199.7.67.1
TLD5.ULTRADNS.INFO.     172800  IN      A       192.100.59.11
TLD6.ULTRADNS.CO.UK.    172800  IN      A       198.133.199.11

;; Query time: 11 msec
;; SERVER: 198.41.0.4#53(a.root-servers.net)
;; WHEN: Sun Mar 20 19:59:13 2005
;; MSG SIZE  rcvd: 290
```

What type of query will the resolver send next and to whom? What will the value of the query's name field be?

The resolver will send a DNS query with (Type=A, Name="www.kernel.org") to 204.74.112.1, which is the IP address of the top-level domain server for the .org domain.

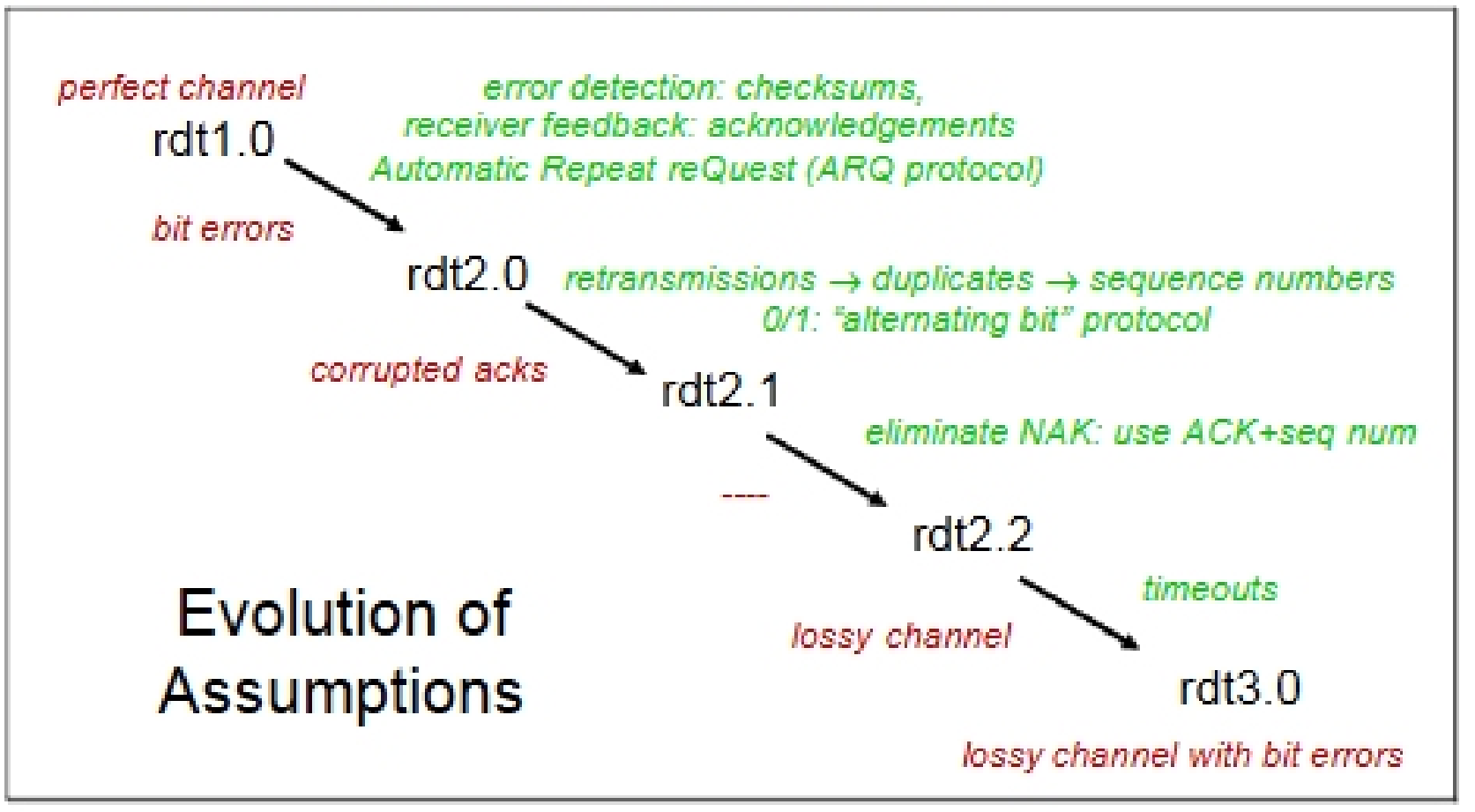
Note that it would not send a query with (Type=NS, Name="kernel.org") because it cannot know for certain whether a domain kernel.org exists – it could be, theoretically, that www.kernel.org is a host in the .org domain. For instance, www.cs.vt.edu is a host in the vt.edu domain, and there does not exist an NS

entry for cs.vt.edu, so a resolver would ask vt.edu's nameserver for (Type=A, Name="www.cs.vt.edu"), not (Type=NS, Name="cs.vt.edu") – we discussed this example in class. On the other hand, 204.74.112.1 will not answer the A query directly, rather, it will return a NS entry for kernel.org.

2 Alternating Bit Protocols (16 pts)

a) Rdt3.0 Receiver FSM (6 pts)

In lecture we discussed how the mechanisms used by reliable data transfer protocols changed as the different assumptions regarding the underlying channel evolved. We considered the spectrum from a reliable, perfect channel to a lossy one that can have bit errors.
You may recall this slide that summarizes this evolution:



We discussed the sender and receiver FSMs for the different versions of the resulting protocols, with the exception of the receiver FSM used in rdt3.0.
Does the receiver FSM used in rdt3.0 differ from the receiver FSM used in rdt2.2? If so, how? If not, why not? Be specific!

The receiver FSM used in 3.0 is the same as in rdt2.2. The sender handles packet loss by retransmitting after a timeout. These retransmissions can introduce duplicates in the channel if the packet wasn't actually lost, but the rdt2.2 receiver FSM can already deal with such duplicates since they could have been caused by corrupted acks.
Note that rdt2.2 and rdt3.0 are also identical in that the receiver will resend the last ack if it receives a corrupted packet: this will give the sender early indication to retransmit the packet: it would be inefficient to remove this provision from rdt3.0, even though the sender would eventually retransmit anyway.

b) Reordered Packets (10 pts)

Give a sender and receiver timeline that shows that an alternating-bit protocol such as rdt3.0 can fail if the underlying channel can reorder packets. (Show the timeline for the sender on the left, for the receiver on the right, and include the type and sequence numbers of messages exchanged.)

One such scenario is shown below. A key characteristic of failure is that the receiver will accept an old copy of data sent by the sender as new data. Note that "reordering" means that if packets A and B are sent in the order A first, B second, then they arrive in the order B first, A second. It does not mean that the channel somehow flips the sequence numbers in the headers of the packets. It also does not mean that the sender FSM violates its rules and somehow sends the same sequence number twice. Furthermore, do not confuse "reordering" with "out-of-order". Out-of-order means that the packet that arrives doesn't have the expected sequence number. If there are duplicates, a receiver must deal with out-of-order packets even when there is no reordering in the underlying channel. Alternating bit protocols such as rdt3.0 use {0, 1} as sequence numbers (hence the name) and do not use pipelining. Any scenario in which the sender sends packet #1 (or more) without having seen ack #0 is wrong. Recall that the cardinality of the sequence number space must be at least twice the window size, so {0, 1} allows for only a window of size 1 – that is, no pipelining.