TECHNISCHE UNIVERSITÄT BERLIN

Fakultät IV – Elektrotechnik und Informatik
Fachgebiet Intelligente Netze und Management verteilter Systeme
Prof. Anja Feldmann, Ph. D.
Harald Schiöberg, Bernhard Ager, Vlad Manilici

6. "Ubung zur Protocol Design WS 08/09

Question 1: (100 points) RDT 2.2 Functionality Extend your program from the last sheet with RDT 2.2 functionality. (Reliable datatransfer over a channel with bit errors)

To do this your program has to use some new fields in the RDT header:

- Magic: The first six Bytes of a RDT 2.2 packet always have the value `RDT2.2` in ASCII encoding.

- Sequence number: A 1 Byte field. It contains either the value 0 or 1, depending on witch packet it is.

- Ack number: A 1 Byte field. It contains either 0 or 1, depending on which packet is being acked.

- Checksum: A 8 bit field. The Checksum is calculated as an "exclusive-or" over all octets of the packet. The checksum field itself is ignored when computing the checksum.
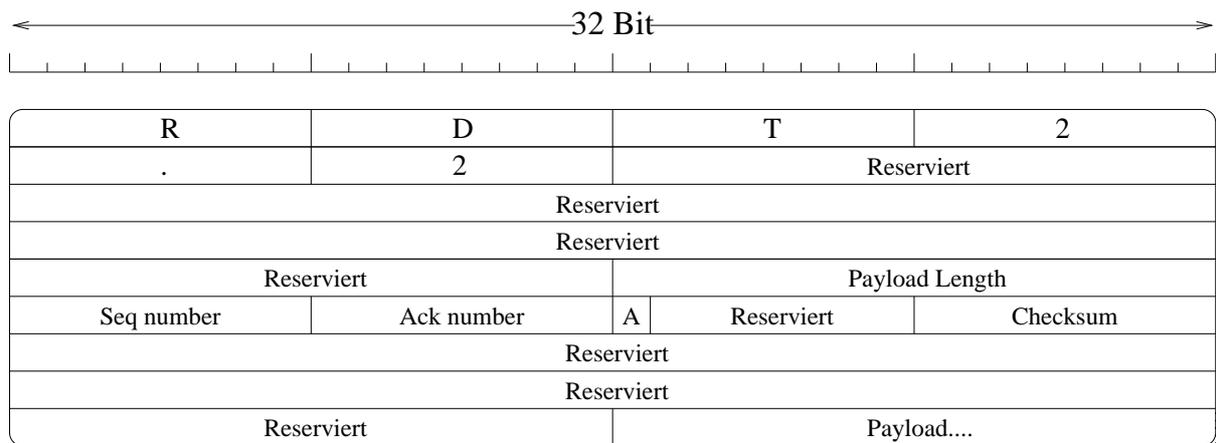
RDT 2.2 knows so-called piggy-back ACKs. That means a packet can transport Data and acknowledge a preceding packet of the opposite direction. Your program *needs not to send such piggy-back ACKs, but it* must be able to deal with them, if they are received. Of course, your program *may send them. (Attention: bonus points)*

To add all that to the code of the last assignment, these locations in your code needs to be changed:

- `statemachine_recv` *This function now implements the complete functionality of the receiver according to Kurose-Ross.*

- `statemachine_send` *This function now implements the functionality of the RDT 2.2 sender, except the transition from "wait for call from above" to "wait for ACK".*

- `send_all` *This function contains the missing transition. The query "can send" needs not only to test for data, but also whether we are in the right state of the state machine.*

Write the function `send_with_error` *to simulate transmission errors. This function should be used instead of the normal* `send` *system call, and should have the same signature. Your program shall accept the probability of a single bit-error within a packet from the command line (in percent). The function* `send_with_error` *should flip a random bit of a packet with the given probability and then call the original* `send`

Here is the new RDT header for your reference:

<--32 Bit-->

R	D	T	2
.	2	Reserviert	
Reserviert			
Reserviert			
Reserviert		Payload Length	
Seq number	Ack number	A	Reserviert	Checksum
Reserviert			
Reserviert			
Reserviert		Payload....	

To submit:

- The source code of your program

—Ende der Aufgabenstellung f'ur das 6. Aufgabenblatt.—

Abgabe: bis Dienstag, 2. Dezember 2008, 23:59h s. t.