



## 6. Übung zur Protokolldesign WS 08/09

### Aufgabe 1: (100 Punkte) RDT 2.2 Funktionalität

Erweitere nun dein Programm vom letzten Blatt auf die Funktionalität von RDT 2.2. (Zuverlässiger Datentransfer über einen Kanal mit Bitfehlern).

Dazu muss dein Programm einige neue Felder im RDT Header benutzen:

- **Magic:** Die ersten 6 Bytes eines RDT 2.2 Pakets haben immer den Wert `RDT2.2` in ASCII-Codierung.
- **Sequenznummer:** Ein 1 Byte Feld, es hat entweder den Wert 0 oder 1, je nach dem, welches Paket das gerade ist.
- **Acknummer:** Ein 8 bit Feld, es hat entweder den Wert 0 oder 1, je nach dem, für welches Paket wir ein ACK schicken.
- **ACK:** Ein 1 bit Flag, es ist gesetzt, wenn die ACK Nummer im Paket gültig ist.
- **Checksumme:** Ein 8 bit Feld. Die Checksumme wird gebildet, indem ein Exklusiv-Oder über alle Oktetts des Packets gebildet wird. Bei dieser XOR-Bildung wird das Checksummenfeld selbst ignoriert.

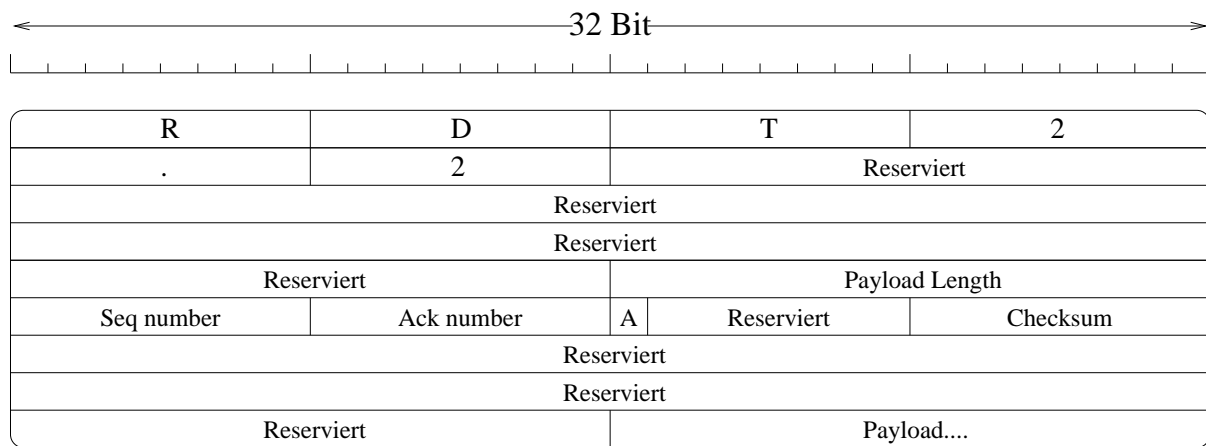
RDT 2.2 kennt sogenannte Piggy-Back ACKs. Das heisst, ein Packet kann gleichzeitig Daten transportieren, und ein vorhergegangenes Paket der Gegenrichtung bestätigen. Dein Programm *muss* nicht in der Lage sein, solche Piggy-Back ACKs zu erzeugen, es muss sie allerdings richtig verarbeiten können, wenn es sie erhält. Natürlich *darf* es sie verschicken. (Vorsicht: Bonuspunkte...)

Um das in deinen Code vom letzten Blatt einzufügen, müssen an folgenden Stellen Änderungen vorgenommen werden:

- **`statemachine_recv`** Diese Funktion implementiert nun die vollständige Funktionalität des Empfängers nach Kurose-Ross.
- **`statemachine_send`** Diese Funktion implementiert nun die Funktionalität des RDT 2.2 Senders, bis auf den Übergang von "Warte auf Aufruf von oben," nach "Warte auf ACK,,".
- **`send_all`** Diese Funktion stellt den fehlenden Übergang im Sender dar. Die Abfrage "Kann senden,," muss nun nicht nur abfragen, ob Daten vorhanden sind, sondern auch, ob man sich im richtigen Zustand befindet.

Schreibe die Funktion `send_with_error`, um Fehler bei der Übertragung zu simulieren. Diese Funktion soll anstelle des normalen `send` Systemaufrufs verwendet werden, und die gleiche Signatur aufweisen. Dein Programm soll die Wahrscheinlichkeit für einen einzelnen bit-Fehler im Paket in Prozent von der Kommandozeile akzeptieren. Die Funktion `send_with_error` soll mit der entsprechenden Wahrscheinlichkeit ein zufälliges bit im Paket vertauschen, und dann `send` aufrufen.

Hier nochmal der neue RDT-Header zum Überblick:



Abzugeben sind:

- Der Quelltext Deines Programmes

—Ende der Aufgabenstellung für das 6. Aufgabenblatt.—

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