



## 4th Assignment Protocol Design WS 08/09

### Question 1: (100 points) Simplified DNS client (part two)

- (a) Now you should send the DNS request to a DNS server and parse the corresponding reply packet. This may contain arbitrary many answers in sections **ANSWERS**, **AUTHORITYS** and **ADDITIONALS**. The exact number is inserted in the corresponding header fields by the server. A single answer is encoded in a Resource Record (RR):

multiple octets	16	16	32	16	multiple octets
<b>Name</b>	<b>Type</b>	<b>Class</b>	<b>TTL</b>	<b>Rdlength</b>	<b>Rdata</b>

Details can be found in **RFC 1035, Sec. 4.1.3**. For now your program needs only support for answers with type **A** and class **IN**, other cases can be ignored. It is furthermore still sufficient to process only the first RR in the **ANSWERS** section; **AUTHORITYS** and **ADDITIONALS** can be ignored.

Your program must first parse the complete header of the answer packet, and print the individual fields to standard out. Errorcodes sent by the server must be treated properly and the program must terminate with a suitable error message.

Subsequently your program must output the **QUESTIONS** and **ANSWERS** sections, where you can assume exactly one question and one answer. The output format of your program should be roughly similar to the **dig** command.

An additional challenge is, that DNS uses a compression algorithm for names. For this part of the question you need not understand the compression algorithm. Just assume, that the **Name** field in the **ANSWERS** section is exactly two bytes long due to compression, and output the string **compressed** instead of the real DNS name.

#### Submit:

- The source code of your program
- The output of your program, if you query for **www.heise.de**. You can use **130.149.220.253** or **130.149.2.12** as nameserver.

- (b) Extend your program to cope with compressed DNS names. A detailed explanation of the algorithm can be found in **RFC 1035, Sec. 4.1.4**.

#### Submit:

- The source code of your program
- The output of your program, if you query for **www.heise.de**.

- (c) Add support for **AUTHORITYS** and **ADDITIONALS**. Add these sections to your program's output. Add support for the types **CNAME** and **NS**. You can still assume exactly one RR in the **QUESTIONS** section. Your output format should further on be roughly similar to **dig**.

#### Submit:

- The source code of your program
- The output of your program for **www.heise.de** and **www.google.de**.

- (d) Extend your program with the option to perform iterative queries, and to deal with the related answers. Output all answer packets. The query should be done automatically, i. e., you should need to run your program only once to do a full iterative query.

**Submit:**

- The source code of your program
- The output of your program, while doing an iterative resolution of `www.heise.de`.

**For details on the submission: see online FAQ.**

**Submission Deadline:** Wed, 19 Nov 2008, 23:59h