TECHNISCHE UNIVERSITÄT BERLIN
Fakultät IV – Elektrotechnik und Informatik
Fachgebiet Intelligente Netze und Management verteilter Systeme
Prof. Anja Feldmann, Ph.D.
Gregor Maier, Jörg Wallerich, Andi Wundsam

4th Work Sheet Praktikum Protokolldesign WS 07/08

Question 1: (100 points) **Simple DNS-Client – Part II**

On the last work sheet you have constructed a DNS request, sent it to a server and collected the reply. This week you will parse and analyse this reply.

The DNS reply packet can contain any number of entries in the `ANSWERS`, `AUTHORITYS`, and `ADDITIONALS` sections. The actual number of entries per section is written in the appropriate header fields by the server. A single entry in a reply section is encoded as a Resource Record (RR).

multiple octets	16	16	32	16	multiple octets
Name	**Type**	**Class**	**TTL**	**Rdlength**	**Rdata**

Details are specified in **RFC 1035, Section 4.1.3**.

For this exercise you can either reuse your own code from last week or you can use a perl module provided by us. You can use this module by specifying `require "/afs/net.t-labs.tu-berlin.de/home/praktikum/da` in your program. In the directory you will also find a small example program, which demonstrates the use of our perl module.

(a) Parse and output the server's reply. First your program should analyze the complete header and output the header fields. If the server returned an error, the program should exit with an appropriate error message. Your program only has to handle entries with type `A` and class `IN`.

Now you should print the `QUESTIONS` and `ANSWERS` sections. You only have to print the first entry in the `QUESTIONS` and `ANSWERS` section. `AUTHORITYS` and `ADDITIONALS` can be ignored. Your output should contain the same information that the `dig` program prints. *It is not required to clone `dig`'s output format!*

A problem is, that DNS uses a compression algorithm to save space when encoding DNS names. For the time being it's not required to understand how the compression works. Note that the `Name` field in the `ANSWERS` section is exactly 2 bytes long if the name is compressed. Output the string `compressed` instead of the DNS name in this case.

Deliverables:

- The sourcecode of your program
- The output of your program when you call it for `www.heise.de` and `www.net.t-labs.tu-berlin.de`. You can use `130.149.220.253` as nameserver.

(b) Enhance your program, so that it can handle compressed DNS names in answers. A description of the algorithm can be found in **RFC 1035, Section 4.1.4**.

Deliverables:

- The sourcecode of your program
- The output of your program when you call it for `www.heise.de` and `www.net.t-labs.tu-berlin.de`.

(c) Complete your program, so that it prints all entries in the `ANSWERS`, `AUTHORITYS`, and `ADDITIONALS` sections. In addtion to type `A` your program should also be able to handle type `NS`. If you encounter an unknown type, your program should print `unknown type`. You should print the same information that the `dig` prints.

Deliverables:

- Der Quelltext Deines Programmes
- The sourcecode of your program
- The output of your program when you call it for `www.heise.de` and `www.net.t-labs.tu-berlin.de`.

(d) Enhance your program, so that it can do iterative queries and that it can handle the answer packets encountered on the way. I.e., if a server reply does not contain the desired answer, your program has to automatically ask the next server in the chain. Print all reply packets your program receives.

Deliverables:

- The sourcecode of your program
- The output of your program, showing an iterative resolution for `www.heise.de`. Use `193.0.14.129` as the nameserver to query[1].

Submission details: look at the FAQ (`http://www.net.t-labs.tu-berlin.de/teaching/ws0708/PD_labcourse/`)
Due Date: Tuesday, November 20, 2007, 11:59 h s. t. (am)

[1] This is the K Rootserver