



4. Blatt Praktikum Protokolldesign WS 07/08

Aufgabe 1: (100 Punkte) Vereinfachter DNS-Client – Teil II

Nachdem Ihr für das letzte Übungsblatt eine DNS-Anfrage konstruiert habt, sie an einen Server gesendet habt und die Antwort gespeichert habt, sollt Ihr nun die Antwort des Servers empfangen und dann ausgeben.

Ein DNS-Antwortpaket kann jeweils beliebig viele Antworten in den Bereichen **ANSWERS**, **AUTHORITYS** und **ADDITIONALS** haben. Die konkrete Anzahl trägt der Server in die entsprechenden Felder im Header ein. Eine einzelne Antwort wird durch einen Resource Record (RR) kodiert:

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

Genauereres findet sich in **RFC 1035, Abschnitt 4.1.3**.

Für diese Aufgabe könnt Ihr entweder euer eigenes Programm zum Erstellen einer DNS-Anfrage oder ein von uns zur Verfügung gestelltes Perl-Modul benutzen. Ihr könnt dieses Modul mittels `require "/afs/net.t-labs.tu-berlin.de/home/praktikum/data/4.uebung/dns_utils.pm"`; in euer Programm einbinden. In dem Verzeichnis findet ihr kurzes Stück Beispielprogramm, das beschreibt, wie man das Modul benutzt.

- (a) Interpretiere und gib die Serverantwort aus. Dein Programm sollte zunächst den kompletten Header des Antwortpaketes auswerten und die einzelnen Felder ausgeben. Hierbei muss darauf geachtet werden, dass man bei eventuellen Fehlercodes vom Server die weitere Bearbeitung mit einer passenden Fehlermeldung beendet. Es ist ausreichend, wenn Dein Programm nur Antworten vom Type **A** und Class **IN** unterstützt, andere Fälle brauchen nicht betrachtet zu werden.

Anschließend ist der **QUESTIONS**- und **ANSWERS**-Bereich auszugeben, wobei man nur die erste Question und die erste Answer ausgeben muss. **AUTHORITYS** und **ADDITIONALS** können ignoriert werden. Das Ausgabeformat Deines Programmes sollte möglichst genau die selben Informationen wie die Ausgabe von **dig** enthalten.

Ein Problem ergibt sich noch daraus, dass DNS bei den Antworten einen Kompressionsalgorithmus verwendet, um bei DNS-Namen Platz zu sparen. In diesem Aufgabenteil musst Du noch nicht verstehen, wie diese Kompression funktioniert. Gehe einfach davon aus, dass das Name-Feld im **ANSWERS**-Bereich durch die Kompression genau 2 Bytes lang wird und gib anstatt dem richtigen DNS-Namen einfach die Zeichenkette **compressed** aus.

Abzugeben sind:

- Der Quelltext Deines Programmes
- Die Ausgabe Deines Programmes, wenn Du es für `www.heise.de` und `www.net.t-labs.tu-berlin.de` aufrufst. Als Nameserver kannst Du z. B. `130.149.220.253` verwenden.

- (b) Erweitere Dein Programm, so dass es auch mit komprimierten DNS-Namen bei den Antworten zurechtkommt. Eine genaue Erklärung des Algorithmus findest Du in **RFC 1035, Abschnitt 4.1.4**.

Abzugeben sind:

- Der Quelltext Deines Programmes
- Die Ausgabe Deines Programmes, wenn Du es für `www.heise.de` und `www.net.t-labs.tu-berlin.de` aufrufst.

- (c) Dein Programm ist nun so zu vervollständigen, dass es alle Antworten (ANSWERS, AUTHORITYS und ADDITIONALS) ausgibt. Dabei soll es bei der Ausgabe zusätzlich zu Type A auch Type NS unterstützen. Bei unbekanntem Typen soll dein Program `unknown type` ausgeben. Beim QUESTIONS-Bereich kann weiterhin von genau einer Question ausgegangen werden. Die ausgegebenen Informationen sollen wieder denen von `dig` entsprechen.

Abzugeben sind:

- Der Quelltext Deines Programmes
- Die Ausgabe Deines Programmes für `www.heise.de` und `www.net.t-labs.tu-berlin.de`.

- (d) Erweitere Dein Programm um die Möglichkeit von iterativen Anfragen und die Fähigkeit, mit entsprechenden Antworten umzugehen. Sollte also in einem Antwortpaket nicht die angefragte Information enthalten sein, so muss Dein Programm selbständig beim nächsten Server in der Kette weiterfragen. Gib alle Antwortpakete aus, die du auf dem Weg zum Ziel erhältst.

Abzugeben sind:

- Die Ausgabe Deines Programmes, die eine iterative Auflösung von `www.heise.de` zeigt. Starte deine Anfrage beim Nameserver `193.0.14.129`¹.

Details zur Abgabe der Aufgaben: siehe FAQ (http://www.net.t-labs.tu-berlin.de/teaching/ws0708/PD_labcourse/)

Abgabe: bis Dienstag, den 20. November 2007, 11:59 h s. t.

¹Das der K Rootserver