Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Esolution
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Grundlagen Rechnernetze und Verteilte Systeme

Exam:	IN0010 / Hausaufgabe 11	**Date:**	Monday 15th June, 2020
Examiner:	Prof. Dr.-Ing. Georg Carle	**Time:**	14:00 – 23:59

Working instructions

- Die erreichbare Gesamtpunktzahl beträgt 36 credits.
- Bitte geben Sie bis spätestens Montag, den **20. Juli um 23:59 CEST** über TUMexam ab.
  Bitte haben Sie Verständnis, wenn das Abgabesystem noch nicht reibungslos funktioniert. Wir arbeiten daran!
- Ihren **persönlichen** Link zur Abgabe finden Sie auf Moodle. Geben Sie diesen **nicht** weiter.
- Bitte haben Sie Verständnis, falls die Abgabeseite zeitweilig nicht erreichbar ist.

Bitte nehmen Sie die Hausaufgaben dennoch ernst:

- Neben der Einübung des Vorlesungsstoffs und der Klausurvorbereitung dienen die Hausaufgaben auch dazu, den Ablauf der Midterm zu erproben.
- Finden Sie einen für sich selbst praktikablen und effizienten Weg, die Hausaufgaben zu bearbeiten. Hinweise hierzu haben wir auf `https://grnvs.net.in.tum.de/homework_submission_details.pdf` für Sie zusammengestellt.

Left room from _____ to _____ / Early submission at _____

Problem 1 Domain Name System (DNS) (14 credits)

Hinweis: Angelehnt an Endterm 2015
Zentrale Aufgabe des Domain Name Systems (DNS) ist es, menschenlesbare Namen auf IP-Adressen abzubilden, die dann für die Wegwahl auf der Netzwerkschicht verwendet werden können. Bei dem Namen `asciiart.grnvs.net.` handelt es sich um einen sog. *Fully Qualified Domain Name (FQDN)*.

0 1

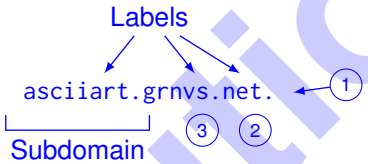
a)* Was ist der Unterschied zwischen einem vollqualifizierten Domain Name (FQDN) und einem nicht-(voll)qualifizierten?

Ein FQDN endet stets mit ., d. h. der Wurzel des Name Spaces.Ein nicht-qualifizierter Domain Name hingegen kann ein einzelnes Label oder eine geordnete Liste durch Punkte getrennter Labels sein, die relativ zu einer anderen Wurzel als . zu sehen sind.

0 1 2

b)* Benennen Sie die einzelnen Bestandteile des FQDNs, sofern es dafür gängige Bezeichnungen gibt.

1. Root (Beginn des Namensraums)
2. Top Level Domain (TLD)
3. Second Level Domain



Da im Alltag zumeist nicht explizit zwischen einem „FQDN“ (also mit terminierenden Punkt) und „Domain Name“ (also ohne terminierenden Punkt) unterschieden wird, da es kontextabhängig klar ist, was von beiden gerade gemeint ist, werden wir[1] im Folgenden auch nur noch dann den Root-Punkt setzen, wenn wir dies besonders hervorheben bzw. deutlich machen wollen.

In Abbildung 1.1 sind ein PC sowie eine Reihe von Servern dargestellt. Wir nehmen an, dass PC1 den Router als Resolver nutzt. Der Router wiederrum nutzt einen Resolver von Google unter der IP-Adresse 8.8.8.8 zur Namensauflösung. Ferner nehmen wir an, dass der Google-Resolver gerade neu gestartet wurde (also insbesondere keine Resource Records gecached hat) und rekursive Namensauflösung anbietet.
Die autoritativen Nameserver für die jeweiligen Zonen sind in Tabelle 1.1 gegeben.

Zone	autoritativer Nameserver
`.`	`d.root-servers.net.`
`com., net.`	`a.gtld-servers.net.`
`google.com.`	`ns1.google.com.`
`grnvs.net.`	`bifrost.grnvs.net.`

Table 1.1: Zonen mit zugehörigen autoritativen Nameservern

0 1 2

c)* Erläutern Sie den Unterschied zwischen einem *Resolver* und einem *Nameserver*.

Nameserver sind autoritativ für eine oder mehrere Zonen („Bereiche“), d. h. sie besitzen eine gültige und aktuelle Kopie der gesamten Zone, für die sie autoritativ sind.
Resolver hingegen extrahieren mittels eine Reihe iterativer Anfragen an die jeweils autoritativen Nameserver die benötigten Information aus dem DNS und geben diese an den anfragenden Client zurück.Resolver können Einträge für begrenzte Zeit cachen, so dass bei erneuter Anfrage derselben Resource Records der Prozess nicht wiederholt werden muss.

d)* Welche Funktion erfüllen `d.root-servers.net` und `a.gtld-servers.net`?

0
1

Der Root-Nameserver ist autoritativ für die Rootzone, d. h. er kennt die Nameserver, welche für die einzelnen TLDs verantwortlich sind, so z. B. `a.gtld-servers.net` als einen der autoritativen Nameserver für net-Domains.
`a.gtld-servers.net` kennt wiederum die zuständigen Nameserver für alle Second-Level-Domains unterhalb der net-TLD.

e)* Erklären Sie den Unterschied zwischen iterativer und rekursiver Namensauflösung.

0
1
2

Rekursive Namensauflösung bedeutet, dass eine DNS-Anfrage an einen Resolver gestellt wird. Dieser wird das endgültige Ergebnis zurücksenden.
Bei iterativer Auflösung hingegen werden schrittweise die autoritativen Nameserver der einzelnen Zonen angefragt.

f) Zeichnen Sie in Abbildung 1.1 alle DNS-Nachrichten (Requests / Responses) ein, die ausgetauscht werden, sobald PC1 auf `asciiart.grnvs.net.` zugreift. Nummerieren Sie die Nachrichten gemäß der Reihenfolge, in der sie zwischen den einzelnen Knoten ausgetauscht werden.

0
1
2
3
4
5

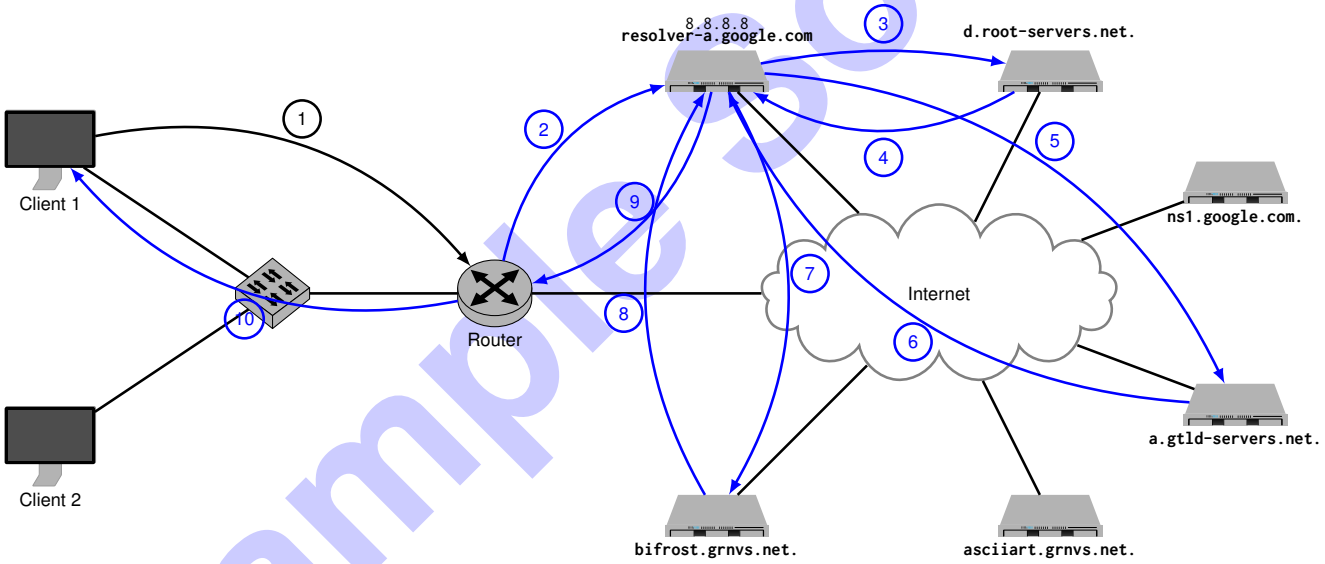


Figure 1.1: Vorlage zu Aufgabe 1f)

g)* Wie wird im DNS sichergestellt, dass kein bösartiger Nameserver Anfragen für andere Domänen beantwortet? (Wir gehen davon aus, dass keine Man-in-the-Middle-Angriffe möglich sind.)

0
1

Dies wird lediglich indirekt dadurch sichergestellt, dass während der iterativen Namensauflösung stets nur die jeweils autoritativen Nameserver kontaktiert werden.Sofern die

- Antwort des Rootservers zuverlässig war und
- die Antwort auf dem Weg vom Rootserver zum anfragenden Nameserver nicht modifiziert wurde

kann ein bösartiger Nameserver keine falschen Antworten liefern – eben da er nie gefragt wird.
Selbstverständlich wird auf diese Weise nicht verhindert, dass DNS-Antworten mittels Man-in-the-Middle-Attacken abgefangen und modifiziert werden können. Dagegen helfen lediglich kryptographische Verfahren, wie sie in der DNSSEC-Erweiterung zu finden sind (nicht in der Vorlesung behandelt).

Problem 2 Kompression: Huffman-Kodierung (12 credits)

Gegeben sei das Alphabet $\mathcal{A} = \{a, b, c, d\}$ und die Nachricht

$$m = \texttt{aabcbdacababbbbcbddbbbaababdbdbb} \in \mathcal{A}^{32}.$$

0 1 2 a)* Bestimmen Sie die Auftrittswahrscheinlichkeiten $p_i \in \mathcal{A}$ der einzelnen Zeichen in m.

Aus den Zeichenhäufigkeiten ergibt sich:

$$p_a = \frac{8}{32} = \frac{1}{4},\ p_b = \frac{16}{32} = \frac{1}{2},\ p_c = \frac{3}{32} \approx 0{,}09,\ p_d = \frac{5}{32} \approx 0{,}16$$

0 1 2 b) Bestimmen Sie den Informationsgehalt $I(p_i)$ der einzelnen Zeichen aus $\mathcal{A}$.

Für den Informationsgehalt erhalten wir:

$$\begin{aligned} I(p_a) &= -\log_2(p_a) = 2\,\text{bit} \\ I(p_b) &= -\log_2(p_b) = 1\,\text{bit} \\ I(p_c) &= -\log_2(p_c) \approx -1{,}6 + 5 = 3{,}4\,\text{bit} \\ I(p_d) &= -\log_2(p_d) \approx -2{,}3 + 5 = 2{,}7\,\text{bit} \end{aligned}$$

0 1 2 c) Die Nachricht m stamme aus einer Nachrichtenquelle X. Bestimmen Sie auf Basis der bisherigen Ergebnisse die Quellenentropie $H(X)$.

Die Quellenentropie ist nichts weiter als die mit den Auftrittswahrscheinlichkeiten gewichtete Summe des Informationsgehalts der Einzelzeichen:

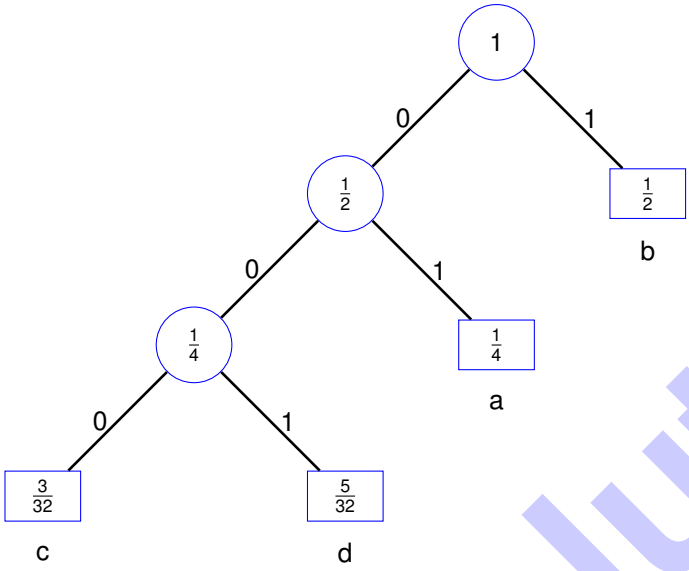
$$H(X) = \sum_{i \in \mathcal{A}} p_i I(p_i) = \frac{1}{4} \cdot 2\,\text{bit} + \frac{1}{2} \cdot 1\,\text{bit} + 0{,}09 \cdot 3{,}4\,\text{bit} + 0{,}16 \cdot 2{,}7\,\text{bit} = 1\,\text{bit} + 0{,}306\,\text{bit} + 0{,}432\,\text{bit} = 1{,}738\,\text{bit}$$

Dies bedeutet, dass sich die Zeichen der Quelle X mit durchschnittlich 1,738 bit pro Zeichen kodieren lassen.

d) Bestimmen Sie nun einen binären Huffman-Code C für diese Nachrichtenquelle.

0 1 2 3

Siehe Vorlesungsfolien. Beginnend bei den beiden Zeichen mit der geringsten Auftrittswahrscheinlichkeit wird ein Baum beginnend bei den Blättern (den Zeichen) konstruiert. Dabei werden in jedem Schritt stets die beiden Knoten bzw. Blätter zusammengefasst, so dass die Summe der Auftrittswahrscheinlichkeiten über alle Knoten bzw. Blätter minimal ist:



Die Kanten werden mit 0 bzw. 1 beschriftet. Der Code lässt sich nun einfach ablesen, indem man von der Wurzel ausgehend die Kantenbeschriftungen abliest: $C = \{a \mapsto 01, b \mapsto 1, c \mapsto 000, d \mapsto 001\}$

Zeichen mit hoher Auftrittswahrscheinlichkeiten erhalten kurze Codewörter. Außerdem lässt sich leicht überprüfen, dass C präfixfrei ist: Kein Codewort ist ein Präfix eines anderen Codeworts. Die Zeichen sind jeweils nur an den Blättern des Baums definiert, nicht jedoch an den inneren Knoten. Dies erleichtert die Dekodierung.

e) Bestimmen Sie die durchschnittliche Codewortlänge von C.

0 1 2

Die durchschnittliche Codewortlänge ergibt sich aus der mit den Auftrittswahrscheinlichkeiten gewichteten Summe der Codewortlängen. Sei $l(c)$ die Länge eines Codeworts in C und $c(i)$ die Funktion, welche ein Zeichen $i \in \mathcal{A}$ auf ein Codewort aus C abbildet. Dann erhalten wir:

$$\bar{l}_C = \sum_{i \in \mathcal{A}} p_i \cdot l(c(i)) = \frac{1}{4} \cdot 2\,\text{bit} + \frac{1}{2} \cdot 1\,\text{bit} + 0{,}09 \cdot 3\,\text{bit} + 0{,}16 \cdot 3\,\text{bit} = 1\,\text{bit} + 0{,}27\,\text{bit} + 0{,}48\,\text{bit} = 1{,}75\,\text{bit}$$

f) Vergleichen Sie die durchschnittliche Codewortlänge von C mit der Codewortlänge eines uniformen[2] Binärcodes.

0 1

Der kürzeste uniforme Code hat eine durchschnittliche Codewortlänge von $\bar{l}_U = 2$. Die Ersparnis beträgt also etwa 12,5 %.

[2] Ein Code heißt *uniform*, wenn alle Codewörter dieselbe Länge aufweisen.

Problem 3 SMTP (Hausaufgabe) (10 credits)

Für daheim: Mailversand von der Kommandozeile
Sie haben in der Vorlesung gelernt, dass mithilfe des Programms `telnet` bzw. mit dem s_client von OpenSSL Verbindungen zu Webservern aufgebaut werden können.
Ihre Aufgabe ist nun, sich mithilfe des s_client mit dem SMTP-Servers Ihres Mailproviders zu verbinden und sich selbst eine e-Mail zu schicken. Die notwendigen Schritte können Sie sich aus dem RFC 5321 erschließen. Alternativ ist es auch möglich entsprechende Tutorials zu konsultieren. Der initiale Befehl könnte folgendermaßen aussehen: `openssl s_client -crlf -connect <smtp.server.org>:465`
Sollten Sie CRAM-MD5 verwenden wollen, könnte Ihnen die folgende Bash-Funktion zur Berechnung der Response auf eine vom Server gegebene Challenge helfen:

```
cram() {
  challenge=$1
  username=$2
  challenge=$(echo -n $challenge|base64 -d)
  echo "Challenge is: $challenge"
  read -sp "Password for $username: " password
  echo ""
  hash=$(echo -n "$challenge" |openssl md5 -hmac "$password" -hex|cut -d" " -f 2 )
  response=$(echo -n "$username $hash" |base64)
  echo "Response for server is: "
  echo $response
}
```

Rufen sie Sie dann folgendermaßen auf:
`cram <CHALLENGE> <USERNAME>`

0 1 2 3 4

a)* Pasten Sie den Output.

220 gmx.com (mrgmx003) Nemesis ESMTP Service ready
EHLO <FQDN of localhost>
250-gmx.com Hello <FQDN of localhost> [XXX.XXX.XXX.XX]
250-SIZE 69920427
250 AUTH LOGIN PLAIN
AUTH LOGIN
334 VXNlcm5hbWU6
<output of: echo -n "<USERNAME>" | base64>
334 UGFzc3dvcmQ6
<output of: echo -n "<PASSWORD>" | base64>
235 Authentication succeeded
mail from:<YOUR AUTHENTICATED MAIL ADDRESS>
6250 Requested mail action okay, completed
rcpt to:<YOUR RECIPIENT ADDRESS>
250 OK
data
354 Start mail input; end with <CRLF>.<CRLF>
From: <SENDER>
To: <RECIPIENT>
Subject: <SUBJECT>

Hello world
.
250 Requested mail action okay, completed: id=0Mcmmn-1XHes72NnN-00Hxc8
QUIT
DONE
–
Verwendet man CRAM-MD5, nimmt man statt AUTH LOGIN
AUTH CRAM-MD5
334 <CHALLENGE>
<RESPONSE>
235 Authentication succeeded

b) Ermitteln Sie, welche Authentifizierungsmethoden der von Ihnen gewählte SMTP-Server anbietet und erläutern Sie den Unterschied zwischen den Methoden.

0 1 2 3 4

- PLAIN erwartet den Usernamen und das Passwort base64-codiert in der Form “NULusernameNUL-password”.
- LOGIN erwartet zuerst den Usernamen base64-codiert und als anschließende Nachricht das Passwort, ebenfalls base64-codiert
- MD5-CRAM ist eine Challenge-Response-Authentifizierung. Der Server sendet base64-codiert einen eindeutigen String, welcher in der Form “username hmac(”challenge“, ”password“)” zurückgesendet werden muss. So wird vermieden, dass das Passwort ungehasht übertragen wird. Die übertragenen Authentifizierungsinformationen können aufgrund der stets wechselnden Challenge auch nicht für Replay-Angriffe genutzt werden.
- SCRAM-SHA-1 ist ebenfalls eine Challenge-Response-Authentifizierung. Sie hat gegenüber MD5-CRAM zum Vorteil, dass der Server während des Dialoges dem Client beweist, dass er tatsächlich im Besitz des (gehashten) Passwortes des Clients ist. Dies ist eine Maßnahme zur Erkennung von MitM-Angriffen. Zusätzlich kann der Server die Passwörter der Clients (im Gegensatz zur Verwendung von CRAM-MD5) beliebig stark gehasht und gesalted in seiner Datenbank abspeichern. Bei Dialog muss dem Client vor dem Austausch der Authentifizierungsinformationen die Hashfunktion (inkl. Rundenanzahl) und der Salt mitgeteilt werden, damit der Client den entsprechenden Hash ausrechnen und damit weiterarbeiten kann.

c) Warum ist überhaupt eine Authentifizierung notwendig? Welche Probleme werden dadurch behoben oder zumindest verringert?

0 1 2

Mailserver ohne Authentifizierung nehmen jegliche Versandaufträge an.Daher können sie zum SPAM-Versand missbraucht werden.

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.