

### **Aufgabe 4.1: JSON**

#### **a) Vor- und Nachteile gegenüber XML**

Durch die Tags in XML können kleine Datenbestände sehr aufgebläht und unübersichtlich wirken. JSON ist sehr spartanisch gehalten und dadurch leicht zu erlernen und leicht zu parsen. Da JSON außerdem gültiges Javascript ist, ist es sofort mit Javascript testbar. Einziger Nachteil ist das JSON metadaten nicht so gut darstellen kann wie XML.

#### **b) XML in JSON**

```
"course": {
  "names": [
    { "name": "Computer Networks I", "lang": "English" },
    { "name": "Netzwerke I", "lang": "Deutsch" }
  ],
  "type": "L",
  "members": [
    { "id": "1088", "firstname": "Michaela", "lastname": "Meier" },
    { "id": "1090", "firstname": "Josef", "lastname": "Kantner",
    "visiting": []}
  ],
  "location": {
    "room": "R0.058", "street"="Lothstr. 64", "city": "Munich"
  }
}
```

### **Aufgabe 5.1: SMTP Protokoll**

#### **a) Microsoft Telnet:**

```
telnet > open mailrelay4.rz.fh-muenchen.de 25
220 mailrelay4.rz.fh-muenchen.de ESMTP Postfix
telnet > EHLO mailrelay4.rz.fh-muenchen.de
250 mailrelay4.rz.fh-muenchen.de
telnet > MAIL FROM: <pmasch@hm.edu>
250 2.1.0 Ok
telnet > RCPT TO: <pm@mm-muc.de>
250 2.1.5 Ok
telnet > DATA
Subject: Wunschzettel - bitte beachten
From: Sarah Meier <sarah@kindergarten.de>
To: Walter Weihnachtsmann <weihnachtsmann.walter@heaven.org>
```

Hi Walter Weihnachtsmann,

MfG,

Sarah

250 2.0.0 Ok: queued as 4CA75380FBC1C

**X-Envelope-From:** <pmasch@hm.edu>; Der tatsächliche Absender der Email.

**X-Envelope-To:** <pm@mm-muc.de> ; Von hm.edu Mailserver eingefügt.  
Zum erkennen des empfangers

**X-Delivery-Time:** 1447947927 ; Unix timestamp, wann die mail empfangen wurde

**X-UID:** 4755 ; eindeutige id

**Return-Path:** <pmasch@hm.edu> ; Fehlermeldungen werden an diese Adresse geschickt

**Authentication-Results:** strato.com 1;

spf=pass

smtp.mailfrom="pmasch@hm.edu"; ; MAIL FROM:

dkim=none;

domainkeys=none;

dkim-adsp=none

header.from="sarah@kindergarten.de"

**X-Strato-MessageType:** email

**X-RZG-CLASS-ID:** mi

**Received-SPF:** pass

(strato.com: domain hm.edu designates 129.187.244.104 as permitted sender)

mechanism=ip4;

client-ip=129.187.244.104;

helo="mailrelay4.rz.fh-muenchen.de";

envelope-from="pmasch@hm.edu";

receiver=smtp.rzone.de;

**identity**=mailfrom; ; Sender Policy Framework um das Fälschen der Absenderadresse zu verhindern

**Received:** from mailrelay4.rz.fh-muenchen.de ([129.187.244.104])

by smtp.rzone.de (RZmta 37.14 OK)

with ESMTP id k04680rAJFjRAQ8

for <pm@mm-muc.de>;

Thu, 19 Nov 2015 16:45:27 +0100 (CET)

; Die Email wurde von mailrelay4.rz.fh-muenchen.de gesendet und am 19.11.2015 um 16:45:27 von smtp.rzone.de (strato) Mailserver empfangen

**Received:** from localhost (localhost [127.0.0.1])

by mailrelay4.rz.fh-muenchen.de (Postfix) with ESMTP id

0C0FF380F6470

for <pm@mm-muc.de>; Thu, 19 Nov 2015 16:45:27 +0100 (CET)

; Die Email wurde von localhost gesendet und am 19.11.2015 um 16:45:27 von mailrelay4.rz.fh-muenchen.de ESMTP empfangen Mailserver

**X-Virus-Scanned:** amavisd-new at hm.edu ; teilt dem Empfänger mit das die mail auf Viren überprüft wurde.

**X-Amavis-Alert:** BAD HEADER SECTION, Missing required header field: "Date"

; teilt dem Empfänger das Ergebnis des Virenskans mit. In diesem Fall, das im Header Date fehlt.

**Received:** from mailrelay4.rz.fh-muenchen.de ([127.0.0.1])  
by localhost (mailrelay4.rz.fh-muenchen.de [127.0.0.1])  
(amavisd-new, port 10024)  
with LMTP id GT9Ky0-X4-gF for <pm@mm-muc.de>;  
Thu, 19 Nov 2015 16:45:26 +0100 (CET)

; Die Email wurde von mailrelay4.rz.fh-muenchen.de gesendet über localhost und am 19.11.2015 um 16:45:26 von mailrelay4.rz.fh-muenchen.de auf port 10024 empfangen mit dem Local Mail Transfer Protocol empfangen. Die mail ist für [pm@mm-muc.de](mailto:pm@mm-muc.de). Dort wird sie auf viren überprüft.

**Received:** from mailrelay4.rz.fh-muenchen.de (c066.hm.vpn.lrz.de [129.187.110.66])  
by mailrelay4.rz.fh-muenchen.de (Postfix) with ESMTTP id 4CA75380FBC1C

for <pm@mm-muc.de>; Thu, 19 Nov 2015 16:44:37 +0100 (CET)  
; Die Email wurde von mailrelay4.rz.fh-muenchen.de gesendet über vpn und am 19.11.2015 um 16:44:37 von mailrelay4.rz.fh-muenchen.de (Postfix) Mailserver empfangen. Die mail ist für [pm@mm-muc.de](mailto:pm@mm-muc.de) (ENVELOPE-TO)

**Subject:** Wunschzettel - bitte beachten

**From:** Sarah Meier <sarah@kindergarten.de>

**To:** Walter Weihnachtsmann <weihnachtsmann.walter@heaven.org>

**Message-Id:** <20151119154527.0C0FF380F6470@mailrelay4.rz.fh-muenchen.de>

**Date:** Thu, 19 Nov 2015 16:45:27 +0100 (CET)

## **Aufgabe 5.2 DNS Server BIND**

### **Wichtig:**

Da vmware benutzt wurde, sind die Ip-Adressen 192.168.178.1-2 für host und virtual switch bereits belegt. Darum wurden die IP-Adressen 3-4 benutzt für PC A und B.

**c)**

```
> dig A www.wikipedia.org
www.wikipedia.org      600    IN      A          91.198.174.192
Query Time: 2ms
> dig AAAA www.wikipedia.org
www.wikipedia.org      600    IN      AAAA       2620:0:862:ed1a::1
Query Time: 49ms
```

**d)**

Die *named.conf.options* datei in */etc/bind* wird um die hosts die Anfragen stellen dürfen und die forwarders erweitert.

```
> dig @localhost www.ieee.org
```

```
; <<>> DiG 9.9.5-3-Ubuntu <<>> @localhost www.ieee.org
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61726
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 13, ADDITIONAL:
1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.ieee.org.                IN      A
```

```
;; ANSWER SECTION:
www.ieee.org.                3079 IN      CNAME www.ieee.org.edgekey.net.
www.ieee.org.edgekey.net.    20521 IN      CNAME e1630.c.akamaiedge.net.
e1630.c.akamaiedge.net.      19 IN      A      23.211.171.92
```

```
;; AUTHORITY SECTION:
.                253 IN      NS      g.root-servers.net.
.                253 IN      NS      j.root-servers.net.
.                253 IN      NS      m.root-servers.net.
.                253 IN      NS      i.root-servers.net.
.                253 IN      NS      k.root-servers.net.
.                253 IN      NS      c.root-servers.net.
.                253 IN      NS      b.root-servers.net.
.                253 IN      NS      a.root-servers.net.
.                253 IN      NS      e.root-servers.net.
.                253 IN      NS      f.root-servers.net.
.                253 IN      NS      l.root-servers.net.
.                253 IN      NS      h.root-servers.net.
.                253 IN      NS      d.root-servers.net.
```

```
;; Query time: 514 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Nov 22 14:14:06 CET 2015
;; MSG SIZE rcvd: 336
```

**e)**

Wir konfigurieren die zone „netze1.cs.hm.edu“ in */etc/bind/named.conf.local*:

```
zone "netze1.cs.hm.edu" {
    type master;
    file "/etc/bind/zones/db.netze1.cs.hm.edu"; # zone dateipfad
};
```

Wir überprüfen die Eingabe mit **>sudo named-checkconf**

Als nächstes erstellen wir das Zonefile *db.netze1.cs.hm.edu* im extra angelegten Ordner *zones*. Wir fügen zuerst einen *Start of Authority(SOA)* Eintrag hinzu, weil dieser der erste Resource Record in der Zone file sein muss.

Danach fügen wir den NS record und unsere A records für die Hosts mit den ip 192.168.0.3-4 und 254 ein.

Wir überprüfen das zonefile mit **>sudo named-checkzone netze1.cs.hm.edu db.netze1.cs.hm.edu**

Und starten den bind service neu **>sudo /etc/init.d/bind9 restart**

**d)**

```
> dig @localhost a.netze1.cs.hm.edu
; <<>> DiG 9.9.5-3-Ubuntu <<>> @localhost a.netze1.cs.hm.edu
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55702
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;a.netze1.cs.hm.edu.      IN      A

;; ANSWER SECTION:
a.netze1.cs.hm.edu.      604800  IN      A      192.168.0.3

;; AUTHORITY SECTION:
netze1.cs.hm.edu.        604800  IN      NS      s.netze1.cs.hm.edu.

;; ADDITIONAL SECTION:
s.netze1.cs.hm.edu.      604800  IN      A      192.168.0.254

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Nov 22 13:44:59 CET 2015
;; MSG SIZE rcvd: 95
```

**f)**

```
> dig @localhost netze1.cs.hm.edu MX
; <<>> DiG 9.9.5-3-Ubuntu <<>> @localhost netze1.cs.hm.edu MX
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45293
```

```
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1,
ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;netze1.cs.hm.edu.          IN      MX

;; ANSWER SECTION:
netze1.cs.hm.edu.          604800   IN      MX      10 mail.netze1.cs.hm.edu.

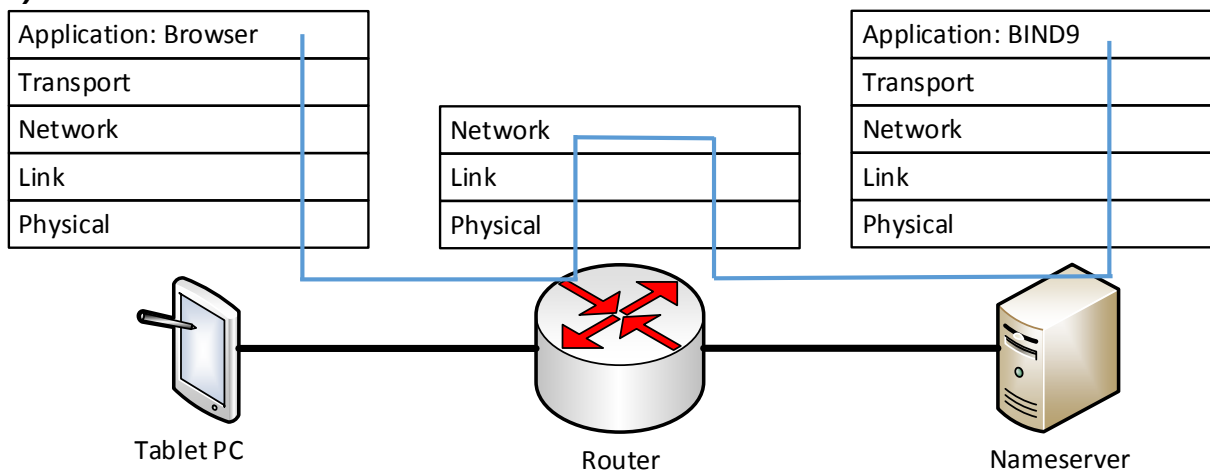
;; AUTHORITY SECTION:
netze1.cs.hm.edu.          604800   IN      NS      s.netze1.cs.hm.edu.

;; ADDITIONAL SECTION:
mail.netze1.cs.hm.edu.      604800   IN      A        192.168.0.254
s.netze1.cs.hm.edu.         604800   IN      A        192.168.0.254

;; Query time: 2 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun Nov 22 14:10:03 CET 2015
;; MSG SIZE rcvd: 114
```

### **Aufgabe 5.3: Rekursive und Iterative DNS Anfragen**

a)



### **DNS-Querie**

Identification: 0x4de3	Flags: 0x0100 Standard query
Number of questions: 1	Number of answer RRs: 0
Number of authority RRs: 0	Number of additional RRs: 0
<a href="http://www.gi.de">www.gi.de</a> IN A	

### **UDP:**

Source Port: 60583	Destination Port: 53
Length: 50	Checksum: 0

### **b) Iterative DNS-Anfragen**

1. Vom Tablet PC DNS-Query zum DSL-Router
2. Vom DSL-Router DNS-Query zum Root Name Server
3. Vom Root Name Server DNS-Answer zum DSL-Router
4. Vom DSL-Router DNS-Query zum gTLD Server
5. Vom gTLD DNS-Answer zum DSL-Router
6. Vom DSL-Router DNS-Query zum Authorative DNS-Server von networkutopia
7. Vom Authorative DNS-Server DNS-Answer zurück zum DSL-Router
8. Vom DSL-Router DNS-Answer zurück zum Tablet PC

### **c) Rekursive DNS-Anfrage**

1. Vom Tablet PC DNS-Query zum DSL-Router
2. Vom DSL-Router DNS-Query zum Root Name Server
3. Vom Root Name Server DNS-Query zum gTLD
4. Vom gTLD DNS-Query zum Authorative
5. Vom Authorative DNS-Answer zurück zum gTLD
6. Vom gTLD DNS-Answer zurück zum Root Name Server
7. Vom Root Name Server DNS-Answer zurück zum DSL-Router
8. Von DSL-Router DNS-Answer zurück zum Tablet PC

## **Aufgabe 6.1: Durchsatzmessung mit UDP und TCP**

### **a)**

Da wir den Goodput d.h. den Durchsatz an Nutzdaten bestimmen wollen, sollte der Anteil an Nutzdaten deutlich größer sein als der Overhead (Header) der Protokolle TCP und UDP.

Der verwendete PC benutzt als Betriebssystem: Windows 10 x64, hat ein Realtek PCIe GBE Family Controller als Netzwerkkarte und einen i7-2600k Prozessor. Außerdem läuft der Server auf JVM Version 1.8.0\_66 mit einer Buffergröße von 64kByte (UDP).

### **b) (NetworkPerformance6.1.xlsx)**

Der verwendete Client PC benutzt als Betriebssystem: Windows 10 x64, hat ein Realtek PCIe GBE Family Controller als Netzwerkkarte und einen Intel i7-2600k Prozessor. Außerdem läuft der Server auf JVM Version 1.8.0\_66 mit einer Buffergröße von 64kByte (UDP).

Der verwendete Server Laptop hat ein Intel Ethernet I217-LM Netzwerkkarte, ein Intel i5 Prozessor 2,5GHz und Windows Version 6.3 Build 9600 als Betriebssystem mit Java VM 1.8.0\_25

Die Verbindung fand über ein Ethernet Kabel mit maximaler Bandbreite von 100Mbit/s und einem 8-Port LevelOne Switch GSW-0809 statt.

Die Änderungen sind marginal.

c) Auch hier ändert sich der Goodput kaum.

### **Aufgabe 6.2: Datenübertragung über UDP mit Alternating Bit Protokoll**

a)

RTT = 50ms;

$$\frac{(4 * 1024 * 1024 \text{ Byte})}{\frac{1200 \text{ Byte}}{\text{Packet}}} = 3495,25 \text{ SnW Pakete}$$

Delay for 1 Packet = RTT(SnW-Paket + Ack) = 50ms

Total Time = SnW-Pakete\*RTT = 3495,25 SnW Pakete \* 50ms = 174.762,5ms  
=174,76s

b)

3495,25/10 = 349,525 Pakete gehen verloren und müssen erneut gesendet werden:

349,525 \* 200ms = 69.905ms = 69,905s

(3495,25Pakete) \* 50ms = 174.762,5ms

Total time = 174.762,5ms + 69.905ms = 244.667,5ms = 244,67s

c)

$$\text{DataThrouput with ABP} = \frac{1200 \text{ Byte}}{0,05 \text{ s}} = 24000 \frac{\text{Byte}}{\text{s}} \approx 23,44 \frac{\text{kByte}}{\text{s}}$$

$$\frac{23,44 \frac{\text{kByte}}{\text{s}}}{1024 \frac{\text{kByte}}{\text{s}}} = 2,29\%.$$

Anstatt alternating bit protokoll, verwendet man zum beispiel das Go-Back-N protokoll oder Selective Repeat.

Oder man ändert den Timeout von 200ms auf 100ms.