GdI - CheatSheet

Exercise 1

ISO/OSI - Schichtenmodell

Anw.	Away!	7	Application
	Pizza	6	Presentation
	Salami	5	Session
Trans.	Throw	4	Transport
	Not	3	Network
	Do	2	Data Link
	Please	1	Physical

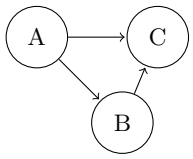
Zuordnung OSI

Prot.	Spec	Medium	OSI
	IP		3
TCP			4
		Koaxkabel	1
FTP			7
UDP			4
HTTP			7
VoIP			7
		Sichtverb.	1

OSI-Geräte

Geräte	Standard	OSI
	802.11	2
Hub		1
Switch		2
Router		3
Gateway		3
	802.3	2 (?)
	MobileIP	3

ARP



A möchte an B schicken, anschliessend C an A. Wie viele ARP-Pakete?

1. A schickt ARP-Broadcast (B,C speichern ARP von A)
2. B schickt ARP an A

C hat die Adresse von A schon durch den Broadcast, daher werden nur zwei verschickt.

IP-Adressen

Präfix

IP-Präfix für 4 Subnetze mit 50 Hosts (A,B) und 200 Hosts (C,D):

A,B **/26** $2^2$6 Bit für Netz, 2^6 für Hosts (64)

C,D **/24** $2^2$4 Bit für Netz, 2^8 für Hosts (255)

IPs

A Router-Port: 192.168.1.1/26, Hosts: 192.168.1.2..51

B Router-Port: 192.168.2.1/26, Hosts: 192.168.2.2..51

C Router-Port: 192.168.3.1/24, Hosts: 192.168.3.2..201

D Router-Port: 192.168.4.1/24, Hosts: 192.168.4.2..201

IP-Fragmentation

Hmm...

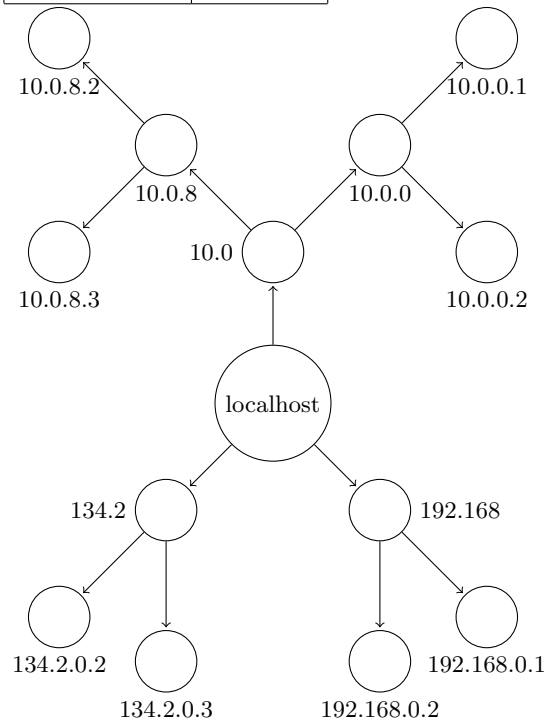
Exercise 2

Forwarding Table

Die Hosts im DSL-Netzwerk müssen über den R1 geroutet werden, da sie nicht nicht geswitched sind oder auf einem Bus kommunizieren.

Trie

Netzwerk	Next Hop
Router 1	
192.168.0.1/24	$H1$
192.168.0.2/24	$H2$
10.0.8.0/21	$R3$
10.0.0.0/8	$R2$



Virtual Circuits

Hosts

VCI	Inport	Outport
Host 192.168.0.1/24		
L0	-	192.168.0.1/24
L1	-	192.168.0.1/24
Host 192.168.0.2/24		
L4	192.168.0.2/24	-
Host 10.0.0.1/8		
L1	10.0.0.1/8	-
L2	10.0.0.1/8	-
Host 10.0.0.2/8		
L2	-	10.0.0.2/8

Router

VCI	Inport	Outport
R1		
L0	10.0.0.254/8	134.2.0.1/16
L1	10.0.0.254/8	134.2.0.1/16
L4	134.2.0.2/16	192.168.0.254/24
R2		
L1	134.2.0.2/16	10.0.0.254/8
L2	134.2.0.2/16	10.0.0.254/8
L3	10.0.0.254/8	134.2.0.2/16
R3		
L0	134.2.0.3/16	10.0.8.1/21
L1	134.2.0.3/16	10.0.8.1/21
L4	10.0.8.1/21	134.2.0.3/16

Exercise 3

Routing im INet

Interior Gateway Protocol

Interior Gateway Protocol (IGP) ist eine Gattungsbezeichnung, die man auf jedes Protokoll anwenden kann, das Informationen über Wegewahl und Erreichbarkeit in einem autonomen System verbreitet. IGP ist ein IP-Protokoll zum Austausch von Routing-Informationen in autonomen Systemen. Obwohl es keinen einzigen Standard für das IGP-Protokoll gibt, ist das RIP-Protokoll das populärste. Das IGP-Protokoll ist topologieunabhängig. Da unterschiedliche Topologien und Netzwerke vorhanden sind, gibt es mehrere Interior-Gateway-Protokolle. Gateways können gleichzeitig verschiedene Routing-Protokolle benutzen, wenn sie die Verbindung zwischen 'autonomen Systemen' und einem übergeordneten Backbone-Netzwerk sind. Hierfür stehen neben dem erwähnten RIP-Protokoll das Hello-Protokoll, das IGRP-Protokoll und das OSPF-Protokoll zur Verfügung.

Exterior Gateway Protocol

Das Exterior Gateway Protocol (EGP) ist auf der Vermittlungsschicht des OSI-Referenzmodells angesiedelt und baut auf dem IP-Protokoll auf. Das EGP wird zur Kommunikation zwischen Routern benutzt und dient dem Verbund mehrerer komplexer Netze, die in sich eine abgeschlossene Welt bilden und nur gelegentlich mit anderen Netzen kommunizieren. Ein solches Netz wird in TCP/IP-Terminologie als autonomes System bezeichnet. Es bildet mit anderen autonomen Systemen im Verbund ein 'Netz von Netzen'. In jedem autonomen System des Netzwerkverbundes wird nun mindestens ein Edge Router (ER) als Exterior-Gateway eingerichtet, der das autonome System mit den anderen autonomen Systemen verbindet.

Abgrenzung

EGP arbeitet mit **Distance Vector Routing** in dem es die Informationen seiner Nachbarn zur Erreichbarkeit anderer Netze sammelt und Informationen an seine eigenen Nachbarn weitergibt.
Ferner ist es limitiert auf baumartig aufgebaute Netze. Hierdurch kann es ab einer Gewissen Netzkomplexität schlichtweg nicht mehr skalieren und ist damit mittlerweile obsolet. IGP hingegen bezeichnet eine ganze Familie von Routingprotokollen, das populärste ist das RIP-Protokoll.

Link State Routing

Beim LSR wird eine Konfigurationsänderung an das komplette Netz geschickt.
Ändert sich ein Link, so wird diese Information direkt an alle beiteiligten Hosts geschickt.

Distance Vector Routing

Beim DVR senden sich die Router gegenseitig Informationen über die von ihnen aus erreichbaren Hosts. Initial versendet jeder Host im Grunde nur, dass er sich selbst über sich selbst erreichen kann.
Darauf aufbauend, kann ein Nachbarhost nun die Information in seine Routingtabelle aufnehmen.
Diese Information sendet er einen Hop weiter an den nächsten Host.
So baut sich nach einigen Informationsaustauschen eine Tabelle mit der Erreichbarkeit des kompletten Netzwerk auf.

3.1.3 Protokollzuorndung

- **RIP** Ein Protokoll der IGP-Familie und nutzt DistanceVectorRouting
- **OSPF** OSPF (Link-State-Routing) ist ein dynamisches Routing-Protokoll innerhalb eines autonomen Systems.
  Es hat das Routing Information Protocol (RIP) als Standard-Interior Gateway Protocol (IGP) abgelöst
- **BGP** Ein DistanceVektorRouting-Protokoll der IGP-Familie

Dijkstra

Die Grundidee des Algorithmus ist es, immer derjenigen Kante zu folgen, die den kürzesten Streckenabschnitt vom Startknoten aus verspricht.
Andere Kanten werden erst dann verfolgt, wenn alle kürzeren Streckenabschnitte beachtet wurden. Dieses Vorgehen gewährleistet, dass bei Erreichen eines Knotens kein kürzerer Pfad zu ihm existieren kann.
Eine einmal berechnete Distanz zwischen dem Startknoten und einem erreichten Knoten wird nicht mehr geändert.
Dieses Vorgehen wird fortgesetzt, bis die Distanz des Zielknotens berechnet wurde (single-pair shortest path) oder die Distanzen aller Knoten zum Startknoten bekannt sind (single-source shortest path).

Step	confirmed	tentative
1	(I,0,-)	(D,5,D);(G,1,G);(E,5,E);(H,3,H)
2	(I,0,-);(G,1,G)	(D,3,G);(E,5,E);(H,3,H)
3	(I,0,-);(G,1,G);(D,3,G)	(E,5,E);(H,3,H);(B,7.5,D)
4	(I,0,-);(G,1,G);(D,3,G);(H,3,H)	(E,5,E);(B,7.5,D);(F,11,H)

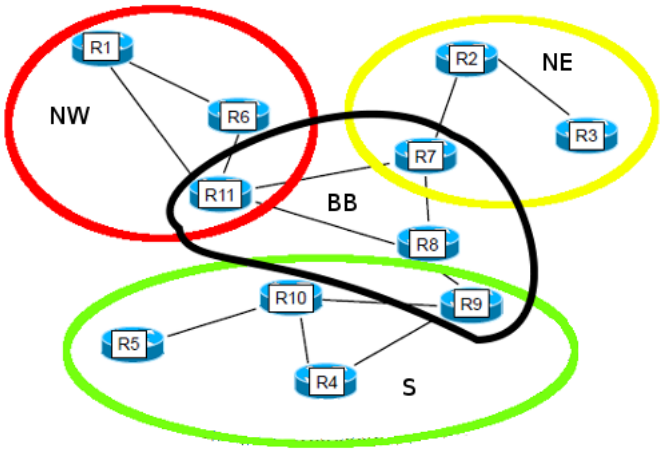
Bellman-Ford

Anders als beim Algorithmus von Dijkstra, dem bekanntesten Verfahren zur Suche nach kürzesten Wegen in Graphen, können die Gewichte der Kanten auch negativ sein, allerdings dürfen keine Kreise negativen Gewichtes vorkommen.
Falls es negative Kreise gibt, findet der Algorithmus nicht zu jedem Knoten den kürzesten Weg. Es ist jedoch möglich, mit dem Algorithmus das Vorhandensein von Kreisen negativen Gewichtes zu erkennen.

Exercise 4

OSPF

Aufteilen des Netzes in drei Regionen



Adresszuweisung

Router	Group	Adress
1	NW	134.2.2.1
2	NE	134.2.3.1
3	NE	134.2.3.2
4	S	134.2.1.1
5	S	134.2.1.2
6	NW	134.2.2.2
7	**BB**	**134.2.4.1**
8	BB	134.2.4.2
9	**BB**	**134.2.4.3**
10	S	134.2.1.3
11	**BB**	**134.2.4.4**

Routing Tabellen der ABRs

R7 (134.2.4.1)

IP	via	Distance
134.2.3.1 (R2)	self	1
134.2.3.2 (R3)	134.2.3.1 (R2)	2
134.2.4/24 (Area NW)	134.2.4.4 (R11)	2
134.2.1/24 (Area S)	134.2.4.2 (R8)	3
134.2.4.4 (R11)	self	1
134.2.4.2 (R8)	self	1
134.2.4.3 (R9)	134.2.4.2 (R8)	2

Vorteile OSPF

- Ein Router in einer Area muss nur die Wege zu Routern in derselben Area kennen. Der genaue Aufbau anderer Areas, auch der Backbone-Area, ist ihm nicht bekannt. Routings in diese Bereiche gehen über den ABR. Dem ABR genügt es, die *Richtung* aller Areas zu kennen. Dadurch wird der Routing-Traffic minimiert.
- Im Gegensatz zum Routing-Information-Protocol garantiert OSPF ein *schleifenfreies* Routing.
- Die Unterteilung in Areas macht die Wartung des gesamten Netzes sehr einfach. Da die genaue Topologie einer Area dem restlichen Netzwerk unbekannt ist, können die Einzelnetzwerke in ihrem Aufbau verändert werden, ohne dass das restliche Netzwerk darüber informiert werden muss. Dadurch ist OSPF vor allem für sehr grosse Netze geeinget, die erweiterbar bleiben sollen.

Verbindung Backbone Area,warum?

Routing in *fremde* Areas gehen bei OSPF immer über die ABRs, die sowohl zum eigenen Bereich als auch zum Backbone verbunden sind. Alle Routinginformationen, die zwischen Areas ausgetauscht werden, gehen über den Backbone. Sollte in OSPF also eine Area (entgegen der Klassifikation) über keine Verbindung zum Backbone verfügen, würde nur Routing innerhalb der eigenen Area möglich.

Effect of congestion

Maximum relative send rate

Jeder Link hat die Bandbreite B, und jeder Link hält 2 Verbindungen. Trivialerweise ist die maximale relative send rate 0.5 .

Loss ratio

Es gilt

$$R = x * B$$

$$\Rightarrow R_{loss} = (x - x_0) * B$$

Somit

$$p = \frac{R_{loss}}{R} = \frac{(x - x_0) * B}{x * B} = \frac{x - x_0}{x}$$

zusammenhang y/R und x

Exercise 5

BGP-advertisements

advertising

UPDATE-Nachrichten (Folie 184)

- $AS1 \rightarrow AS2 : 134/8 : AS1; R1$
- $AS2 \rightarrow AS3 : 134/8 : AS2, AS1; R2$
- $AS2 \rightarrow AS5 : 134/8 : AS2, AS1; R2$
- $AS3 \rightarrow AS4 : 134/8 : AS3, AS2, AS1; R3$
- $AS3 \rightarrow AS5 : 134/8 : AS3, AS2, AS1; R3$
- $AS4 \rightarrow AS5 : 134/8 : AS4, AS3, AS2, AS1; R4$
- $AS5 \rightarrow AS6 : 134/8 :$ $[[AS5], AS4], AS3, AS2, AS1; R5$
- $AS5 \rightarrow AS4 : 138/8 : [AS3], AS2, AS1; R5$
- $AS5 \rightarrow AS3 : 138/8 : AS2, AS1; R5$
- $AS4 \rightarrow AS3 : 138/8 : AS4, AS5, AS2, AS1; R4$

Wie viele Routen kennt R5, welche announced er zu R6?

- Der innere Knoten R5 kennt die Routen über A2,3,4
- Wie in 5.1 anounced er einen Weg der für ihn optimal erscheint. Evtl. nach Kostenaspekt, oder Bandbreite, Stabilität, etc.

Route bricht weg

WITHDRAW-Nachrichten (Folie 186)

- $AS2 \rightarrow AS3 : 134/8$
- $AS2 \rightarrow AS5 : 134/8$
- $AS3 \rightarrow AS4 : 134/8$
- $AS3 \rightarrow AS5 : 134/8$
- $AS4 \rightarrow AS5 : 134/8$
- $AS5 \rightarrow AS6 : 134/8$

Links wieder da, anderer Links brechen weg

- Da es in dem kostenpflichten Links nur AS2 und AS5 betrifft, werden dort keine WITHDRAWS verschickt.
  Sollte AS5 zu AS6 einen Pfad mit dieser Route anounced haben, dann wird er ein UDATE mit einer alternativen Route schicken.
- R3 würde den prefix bei R5 und R4 WITHDRAWen. R3 rekonfiguriert seine Route aus den schon bekommenen Inforamtionen und schickt nun über R5.

BGP - eBGP/iBGP

Mehrere AS_X-Teilnetze, die Frage ist, wie welcher Router in den Netzen die Routen zu den anderen lernt.
Das Ding ist zu schauen, wie viele Hops gebraucht werden, bis die Information von einem zum anderen hoppelt.
Wer zuerst kommt, der malt dann auch zuerst.

Informationswege Teilrouter werden abgefragt

1. Erst breitet sich die Information in AS4 über RIP aus.
2. 4c berichtet über eBGP an 3c
3. 3c nutzt innerhalb AS3 OSPF (3a)
4. AS3 teilt AS1 (dem Router 1c) über eBGP das Prefix mit

What interface of 1d will be configured to route the prefix x?

IF1, da der Router 1a die Information eher hat als 1b.

Welches Interface von RouterX?

Die Hopanzahl zu 1b von 4a ist 3, zu 1a ist sie 6.
Somit ist davon auszugehen, dass 1b die Information eher an 1d schicken wird und somit IF2 konfiguriert wird.

Interface?

Da die Netze alle shared-cost verbunden sind, wird nach einer Heuristik entschieden.
Am wahrscheinlichsten wäre die Entscheidung an hand der Anzahl der Hops zu treffen, dann wäre IF1 die beste Wahl. Evtl. könnte der andere Pfad jedoch mit einer besseren Verbindung werben, dann ist evtl. IF2 besser. Ohne Zusatzinformationen würde ich jedoch IF1 favorisieren.

Exercise 6

Forwarding and routing tables

Stop'N'Wait,GoBackN,Selective Repeat

- 10 Gbit/s Netz
- packet size 4000byte
- rtt 200ms

transmit-Time des Pakets

Es gilt $d_{saw} = \frac{L}{R} + t_{rtt}$, wobei L die Paketlänge ist, R die Übertragungsgeschwindigkeit und t_{rtt} die *round-travel-time*. Im vorliegen Fall wird die Übertragungsdauer der ACK-Pakete vernachlässig, somit gilt $d_{saw} = \frac{4000*8bit}{10^{10}bps} + 20ms = 20,0032ms$

utilization

Es gilt $U_{saw} = \frac{L/R}{t_{rtt}+L/R}$
$\Rightarrow U_{saw} = \frac{0,0032ms}{20,0032ms} = 1,59974 * 10^{-4}$

Effektiver Druchsatz

Effektiv werden für die Übertragung von 4000 byte 20,0032ms benötigt.
Also: $R_{saw} = \frac{4000*8bit}{20,0032ms} = 1599b/ms = 1.599.000b/s$

Fenstergrösse für 90 U_{saw}

$\frac{N * L/R}{rtt + L/R} = 0.9$ $N = 5626$

Warum ist GBN uneffizient bei Paketverlusten?

Bei wiederholten Paketverlusten kommt es bei GBN zur mehrfachen Übertragen von Paketen. Angenommen, Paket x kommt überhaupt nicht an. Der Sender schickt weiterhin Pakete im Fenster n, bis der Timeout für die ACK-Rückmeldung für Paket 3 beim Sender eintritt. Nun wird Paket 3 und **alle** anderen Pakete im Fenster nochmal geschickt.

Wie kann GBN diesbezüglich verbessert werden?

Dieser Nachteil kann z.B. durch einen Buffer beim Empfänger behoben werden. Wird ein fehlerhaftes Paket empfangen, werden trotzdem weiterhin Pakete vom Sender angenommen. Bei einem ACK-Timeout schickt der Sender nun nur das **älteste** noch nicht bestätigte Paket nochmal. Der Empfänger fügt dieses in seinen Buffer ein und kann die Pakete dann komplett weiterleiten.

Selective Repeat

Fenster $[k; k + N - 1]$ **mit N in-flights,welche SequenzNR?**

Paket Nummer $k - 1$ wurde auf Empfängerseite empfangen und bestätigt. Somit wartet der Empfänger auf die Pakete ab der Nummer k.

gehen alle in-flights verloren (data/ack), wieviele SequenzNR werden zur Zuordnung benötigt?

Die Anzahl der Sequenznummern sollte so gross sein, wie die Fenstergrösse.
Durch den Buffer beseht hier die Gefahr, dass ein Paket verloren geht und die Sequenznummer wiederverwendet wird, bevor der Timeout zuschlägt.
So würde der Empfänger die Sequenznummer des zweiten Pakets bestätigen und der Sender würde denken es handelt sich um die Bestätigung des ersten Paketes.

Wieviele SeqNr brauch der GBN zum Erfolg?

Da das Problem aus 6.3.2 nicht auftreten kann, muss der Sequenznummer-Menge nur so gross sein, wie Pakete innerhalb des Timeouts geschickt werden können.
$Timeout/RTT$

Wenn das Netz die Reinfolge verdreht, ändert das was?

SR Bei SR haben wir schon die maximale Sequenznummer-Menge definiert, hier ändert sich nichts.

GBN Durch die extrem vergrösserten Timeouts kann nicht mehr festgestellt werden, welches Paket gerade angekommen ist, sollten die Nummer weiderverwendet werden.
Daher gilt nun auch hier als Sequenznummer-Menge die Fenstergrösse.

Exercise 7

TCP time to receive data

Ablauf:
SYN $\xrightarrow{SYN-ACK}$ ACK + S $\rightarrow$ 2 S $\rightarrow$ 4 S $\rightarrow$ 8 S $\rightarrow$ done
Jeder $\rightarrow$ entspricht einer RTT. Also gibt es 5 RTTs.
Transportdauer beträgt $\frac{15S}{R}$.

$$5\ RTT + \frac{15S}{R}$$

1. $\frac{S}{R} > RTT$: $\quad \frac{20S}{R} > 5\ RTT + \frac{15S}{R} > 20\ RTT$
2. $RTT > \frac{3S}{R}$: $\quad 10\ RTT > 5\ RTT + \frac{15S}{R} > \frac{30S}{R}$
3. $\frac{3S}{R} > RTT > \frac{2S}{R}$: $\quad \frac{30S}{R} > 5\ RTT + \frac{15S}{R} > \frac{25S}{R}$

TCP loss rate

Beweis LossRate irgendwas

Noch nen Beweis von 'rate' aus nächster Aufg...

VerlustWahr. 1500byte,100ms RTT, 10Gbit/s Link?

gegeben:
$MMS = 1500\ Byte,\ RTT = 100ms,\ rate = 10\frac{GBit}{s}$

$$rate = \frac{1.22 \cdot MMS}{RTT \cdot \sqrt{L}}$$
$$rate \cdot RTT = \frac{1.22 \cdot MMS}{\sqrt{L}}$$
$$\frac{rate \cdot RTT}{1.22 \cdot MMS} = \frac{1}{\sqrt{L}}$$
$$L = \left(\frac{1.22 \cdot MMS}{rate \cdot RTT}\right)^2$$
$$L = \left(\frac{1.22 \cdot 1500\ Byte}{1.25\frac{GByte}{s} \cdot 0.1s}\right)^2$$
$$L = 1.8590129435835934e-10$$

Wenn die Wahrscheinlichkeit höher ist, dann wir der Datendurchsatz sinken, da die Windowsize öfter kleiner wird und mehrmals die gleichen Daten gesendet werden müssen.

Exercise 8

DNS-Verkehrsmodelle

Anzahl N_k Domains auf k, k=0 ist root

Es handelt sich um einen vollständigen Baum mit 10 Kindern auf jeder Ebene. Jeder Knoten hat also 10 Nachfolger.
Auf Ebene 0 gibt es einen Knoten, auf der Nächsten Ebene 10, auf der darauffolgenden $10 \cdot 10$, also 100, darauffolgend $10 \cdot 100$ (jeder der hundert hat 10 Kinder) usw. Auf der Ebene k gibt es also 10^k Domänen.

Ebene4-DNS lokaler DNS;iterative Anfragen,mathematischen Ausdruck?

Der lokale Nameserver kennt die obersten m Nameserver bereits, d.h. seine erste Anfrage beginnt gerade bei dem m-ten Nameserver. Somit muss er $k - m + 1$ Anfragen leisten, um an die IP Adresse zu kommen. Somit ist die Zeit für die Anfragedauer gerade $= (k - m + 2) \cdot t$ (Wäre $m = 0$, müsste er jedoch $2 \cdot k$ Anfragen starten, da er sich erst den Baum hocharbeiten müsste, was man nicht machen muss, wenn er zumindestens die Wurzel kennen würde, also $m > 0$ ist)

Antwortzeiten 100ms und m=1,2,3,4,5

$$t = 100ms$$
$$m = 1 : Zeit = k * t = 5 * 100ms = 500ms$$
$$m = 2 : Zeit = k * t = 4 * 100ms = 400ms$$
$$m = 3 : Zeit = k * t = 3 * 100ms = 300ms$$
$$m = 4 : Zeit = k * t = 4 * 100ms = 200ms$$
$$m = 5 : Zeit = k * t = 5 * 100ms = 100ms$$

Wenn DNS rekursiv..

Mathematischer Ausdruck?

Bei der Rekursiven Variante muss man erst k Ebenen nach oben gelangen, dann wieder k Ebenen nach unten, nun schickt man die Antwort zurück, muss also wieder k Ebenen nach oben und wieder k Ebenenen nach unten.
Somit gilt für die Zeit: Zeit $= 4 \cdot k \cdot t$

Zeiten für 100ms und k=0,1,2,3,4

$$k = 0 : 4 \cdot 0 \cdot t = 0$$
$$k = 1 : 4 \cdot 1 \cdot 100ms = 400ms$$
$$k = 2 : 4 \cdot 2 \cdot 100ms = 800ms$$
$$k = 3 : 4 \cdot 3 \cdot 100ms = 1200ms$$
$$k = 4 : 4 \cdot 4 \cdot 100ms = 1600ms$$

Wie lange, wenn der lokale DNS die oberen 1,2,3,4,5 Ebenen schon kennt?

Kennt man die m obersten Nameserver, muss man nicht mehr zweimal nach unten und zweimal nach oben steigen, sondern jeweils nur einmal, und zwar $k - m + 2$ mal oft.
Also gilt: Zeit $= (k - m + 2) \cdot 2 \cdot t$. Somit ergibt sich (für $k = 4$ angenommen, ist aus Aufgabenstellung nicht eindeutig ersichtlich)

$$m = 1 : \text{ Zeit } = (k - m + 2) \cdot 2 = 10 \cdot 100\ ms = 1000\ ms$$
$$m = 2 : \text{ Zeit } = (k - m + 2) \cdot 2 = 20 \cdot 100\ ms = 2000\ ms$$
$$m = 3 : \text{ Zeit } = (k - m + 2) \cdot 2 = 30 \cdot 100\ ms = 3000\ ms$$
$$m = 4 : \text{ Zeit } = (k - m + 2) \cdot 2 = 40 \cdot 100\ ms = 4000\ ms$$
$$m = 5 : \text{ Zeit } = (k - m + 2) \cdot 2 = 50 \cdot 100\ ms = 5000\ ms$$

DNS-Last bei verschiedenen Modellen

A Anfragen gleichverteilt

B 50% werden lokal, der Rest gleichverteilt.

Von den lokalen Nameservern werden insgesamt 10^{n+1} Anfragen generiert, für $n = 4$ haben wir also 10^5 Anfragen pro Sekunde.

Verkehrsmodell A: Sind die Anfragen gleichverteilt, bekommt auf Ebene k ein Nameserver 10^{k+1} Anfragen, also gilt:

$k = 4:$ 10^5 Anfragen
$k = 3:$ 10^4 Anfragen
$k = 2:$ 10^3 Anfragen
$k = 1:$ 10^2 Anfragen
$k = 0:$ 10^1 Anfragen

Verkehrsmodell B: 50% aller Anfragen bleiben auf einer Ebene Lokal

$k = 4:$ 10^5 Anfragen
$k = 3:$ 50000 Anfragen
$k = 2:$ 25000 Anfragen
$k = 1:$ 12500 Anfragen
$k = 0:$ 6500 Anfragen

6.

Verkehrsmodell A:

$k = 4:$ 10^5 Anfragen
$k = 3:$ 10^4 Anfragen
$k = 2:$ 10^3 Anfragen
$k = 1:$ 10^2 Anfragen
$k = 0:$ 10^1 Anfragen

Verkehrsmodell B: Nur beim lokalem Nameserver schlagen alle Anfragen auf. Dieser kann diese dann aber direkt sinnvoll verteilen.

$k = 4:$ 10^5 Anfragen
$k = 3:$ 25000 Anfragen
$k = 2:$ 12500 Anfragen
$k = 1:$ 6500 Anfragen
$k = 0:$ 6500 Anfragen

7.

Verkehrsmodell B:

$k = 4:$ $\frac{10^5}{10^4} = 10$ Anfragen
$k = 3:$ $\frac{10^4}{10^3} = 10$ Anfragen
$k = 2:$ $\frac{10^3}{10^2} = 10$ Anfragen
$k = 1:$ $\frac{10^2}{10} = 10$ Anfragen
$k = 0:$ 10 Anfragen

Verkehrsmodell B:

$k = 4:$ $\frac{10^5}{10^4}$ Anfragen
$k = 3:$ $\frac{50000}{10^3}$ Anfragen
$k = 2:$ $\frac{25000}{10^2}$ Anfragen
$k = 1:$ $\frac{12500}{10}$ Anfragen
$k = 0:$ 6500 Anfragen

NSLookup