

Aufgaben zu Kapitel 2


Aufgabe 1 (Kontrollfragen zu 2.1)

- a.) Nennen Sie fünf standardisierte Internet-Anwendungen und die Anwendungsprotokolle, die sie benutzen.

Lösung: WWW (HTTP), Dateitransfer (FTP), Email (POP3, IMAP, SMTP), Telnet (TELNET), Namensdienst (DNS).

- b.) Welche Information wird von einem Prozess auf einem Host benutzt, um einen Prozess, der auf einem anderen Host läuft, zu identifizieren. 

Lösung: Fully Qualified Address (FQA): Port Number + Protocol Number + IP-Adresse.
Oder DNS: Fully Qualified Domain Name (FQDN) + Protocol Name + Service Name


- c.) Warum nutzen HTTP, FTP, SMTP, POP3 und IMAP als Transportprotokoll TCP anstelle von UDP? 

Lösung: Sie benötigen einen zuverlässigen Transportdienst (kein Data Loss).

- d.) Können Sie sich eine Anwendung vorstellen mit den Anforderungen „no data loss“ und „timing“? 

Lösung: Industrielle Echtzeitanwendungen, z.B. Motion Control bei Robotern.

- e.) Macht es Sinn, einen verbindungsorientierten Dienst auf UDP aufzusetzen?

Lösung: Dies könnte interessant sein, um beispielsweise eine leichtgewichtige Version eines Kommunikationsprotokolls für LANs zu realisieren. TCP wurde für WANs entworfen, d.h. variable RTT und Datenrate. Ein TCP ohne den Overhead der Fluss- und Staukontrolle wäre beispielsweise im LAN interessant. Gleichzeitig kann man beim Aufsetzen auf UDP für die Implementation Sockets benutzen. 

Aufgabe 2 (Kontrollfragen zu 2.2 bis 2.5)

- a.) Stellen Sie sich eine E-Commerce Site vor, die für Ihre Kunden einen Warenkorb verwalten will. Beschreiben Sie, wie dies mit Hidden Fields in Formularen und/oder Cookies realisiert werden kann.

Lösung:



In beiden Fällen wird die Auswahl der Waren im Browser über HTML-Formulare getroffen und zum Server geschickt.

1.) Bei Verwendung von Hidden Fields in Formularen wird der aktuelle Inhalt des Warenkorbs solange zwischen Webbrowser und Webformularen hin- und hergeschickt, bis die Warenauswahl abgeschlossen ist.



2.) Bei Verwendung von Cookies wird vom Server einmalig ein Session Identifier generiert und als *Session Cookie* beim Client gesetzt (Set-Cookie Header). Die HTTP Requests vom Client im Zusammenhang mit dem Warenkorb schicken den Cookie im HTTP Header wieder zum Server. Durch den Session Identifier kann der Datensatz des Warenkorbs auf der Serverseite verwaltet und zugeordnet werden.

Ergänzung:



Session Cookies sind zu unterscheiden von *Persistent Cookies*. *Persistent Cookies* sind Einträge in der Datei COOKIES.TXT (Firefox) oder im Verzeichnis COOKIES (Internet Explorer) auf dem Rechner des Web Client. Sie enthalten folgende Informationen (RFC2965):

- Domain, die den Cookie gesetzt hat und lesen kann
- Information, ob alle Computer der Domain Zugriff auf den Cookie haben
- Pfad innerhalb der Domain, in der der Cookie gültig ist
- Information, ob ein Cookie-Zugriff nur bei SSL-Verbindung (verschlüsselt) möglich ist
- Zeitangabe für die Lebensdauer des Cookies (UNIX Timestamp)
- Name und Wert des Cookies (meist der Session Identifier)

```
# HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file! Do not edit.
```

```
.amazon.de      TRUE  /      FALSE  969923413      session-id      028-3352013-8536562
.amazon.de      TRUE  /      FALSE  969923413      session-id-time  969922800
```

b.) Was ist der Unterschied zwischen HTTP mit nicht-persistenten und mit persistenten Verbindungen?



Lösung: Bei HTTP mit nicht-persistenten Verbindungen wird pro angefordertem Objekt eine neue TCP-Verbindung geöffnet und verwaltet. Nach Übertragung des Objekts schliesst der Server die Verbindung. Zur Steigerung der Performanz können TCP-Verbindungen aber parallel genutzt werden.



Bei HTTP mit persistenten Verbindungen lässt der Server die TCP-Verbindung offen, nachdem er ein Objekt darüber geschickt hat. Weitere Anforderungen und Antworten können somit zwischen Client und Server über dieselbe Verbindung geschickt werden. Insbesondere kann eine komplette Webseite über eine einzige persistente Verbindung übertragen werden. Der HTTP Server schließt die Verbindung typischerweise erst nach einem (konfigurierbaren) Timeout Intervall (Default bei Apache 15s).

c.) Warum spricht man bei FTP von „out of band“ – Kontrolle ?



Lösung: FTP benutzt für die Übertragung von Kontrollinformationen (user name, password, commands) eine von den Daten (Dateien, Directory Listings) getrennte TCP-Verbindung.

d.) Was versteht man unter „Handshaking“ bei SMTP?

Lösung: Client und Server identifizieren sich gegenseitig mit ihren FQDNs, bevor sie Informationen übertragen.

Beispiel SMTP:



S: 220 nice.rz.hs-mannheim.de ESMTP Sendmail 8.8.8/8.8.8; ...



C: HELO n-mail.n.hs-mannheim.de

Die Handshaking-Phase kann insbesondere zum Prüfen von DNS Blacklisting genutzt werden.

e.) Nehmen Sie an, Sie versenden eine Email, die lediglich ein Microsoft Excel Attachment als Daten enthält. Wie werden die Kopfzeilen (mit MIME-Zeilen) aussehen?



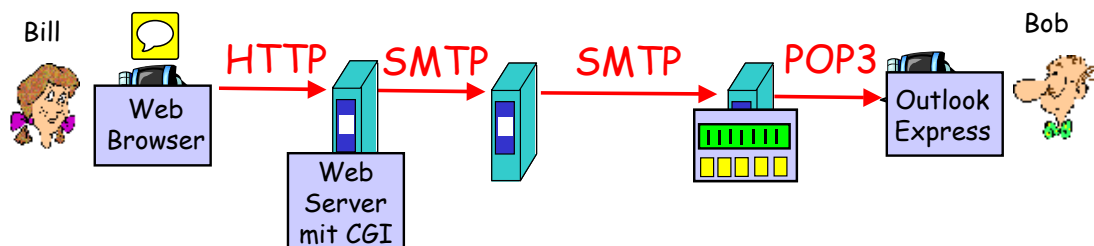
Lösung:

From: koe@n.hs-mannheim.de
To: e.koerner@hs-mannheim.de
Date: Tue, 19 Sep 2000 18:13:41 +0200
MIME-Version: 1.0
Subject: Excel Sheet
Content-Type: application/ms-excel;
name="Soe.xls"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="Soe.xls"



f.) Nehmen Sie an, Bill mit einem Web-basierten Email-Account (z.B. Yahoo!mail oder Hotmail) sendet eine Nachricht an Bob, der seine Mail mit POP3 abrufen. Zeigen Sie die Client-Server-Kette und benennen Sie die beteiligten Anwendungsprotokolle.

Lösung:



g.) Vergleichen Sie die Protokolle HTTP und SMTP im Hinblick darauf, wie die Protokolle spezifiziert sind, in welche Richtung die Information transferiert wird und wie die TCP-Verbindungen genutzt werden.



Lösung: HTTP und SMTP sind Client/Server-Protokolle, deren Nachrichten als ASCII-Text spezifiziert sind. Beide verwenden Status Codes und Status Phrases in den Antworten der Server. Bei HTTP wird Information vom Server „gezogen“ („pull“-Modell), bei SMTP wird Information auf den Server „geschoben“ („push“-Modell). Bei HTTP können mehrere Objekte einer Webseite über verschiedene TCP-Verbindungen gesendet werden, auch wenn persistente Verbindungen genutzt werden. Bei SMTP werden die MIME-Nachrichtenteile einer Email über eine TCP-Verbindung transferiert. SMTP nutzt also immer persistente Verbindungen.



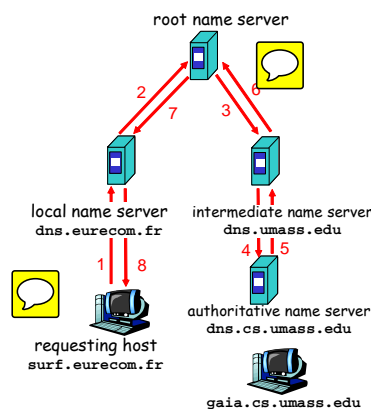
Anmerkung: Im Web2.0 werden Uploads mit HTTP zur Gewohnheit, sodass man nicht mehr von einem reinen „pull“-Modell sprechen kann.

- h.) Nennen Sie Gründe, warum es wenig Sinn macht, den DNS mit einer zentralisierten Datenbank bei einer zentralen Institution zu betreiben.

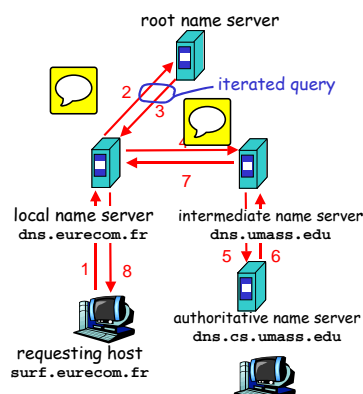


Lösung: Würde die zentrale Datenbank ausfallen, wäre der DNS komplett ausgefallen („single point of failure“). Außerdem wäre die zentrale Datenbank einer ungeheuren Last ausgesetzt (keine Lastverteilung). Je nach Entfernung zur zentralen Datenbank können starke Laufzeitverzögerungen bei DNS-Abfragen auftreten. Der Verwaltungsaufwand bei der zentralen Institution wäre nicht zu bewältigen. Ein zentraler DNS wäre also nicht skalierbar!

- i.) Verändern Sie den DNS-Ablauf in der folgenden Abbildung derart, dass der Root Name Server dadurch entlastet wird.



Lösung: Interaktion 3 und 6 können iterativ als Interaktion 4 und 7 zwischen dns.eurecom.fr und dns.umass.edu stattfinden.



- j.) Jeder Internet Host besitzt einen lokalen Nameserver und einen authoritative Nameserver. Beschreiben Sie die Rolle dieser Server.



Lösung: Jeder Host besitzt einen Konfigurationseintrag für den lokalen Nameserver, und richtet eine DNS Query zuerst an diesen. Typischerweise liegt der lokale Nameserver geographisch nahe am Host (er steht beim ISP oder in der Firma).



Für Hosts werden der DNS-Name und die IP-Adresse in der DNS-Datenbank eingetragen. Der Nameserver, der das Resource Record (RR) für den Host permanent speichert, ist der authoritative Nameserver. Der lokale Nameserver hält meistens auch die DNS-Datenbank für alle Hosts eines ISP und nimmt dann auch die Rolle des authoritative Nameserver ein.

- k.) Ist es möglich, dass der Web Server und der Mail Server einer Firma exakt den selben Alias (z.B. `server.foo.com`) für den kanonischen Hostnamen (z.B. `host.foo.com`) benutzen? Was ist der Typ des RR, der den Hostnamen für den Mail Server enthält? Was ist der Typ des RR, der die IP-Adresse für den Web Server enthält?



Lösung: Die Frage impliziert, dass der Web Server und der Mail Server auf demselben Host (hier: `host.foo.com`) laufen. Dort werden die Serverprozesse durch unterschiedliche Portnummern unterschieden. Es ist also möglich, denselben Alias zu verwenden. Der RR für den Hostnamen des Mail Servers ist vom Typ MX, derjenige für die IP-Adresse des Web Servers ist vom Typ A.



- l.) Sind folgende Anwendungsprotokolle zustandslos oder zustandsbehaftet?

SMTP: Zustandsbehaftet.

DNS : Zustandslos.

HTTP: Zustandslos.

FTP: Zustandsbehaftet - Nutzeridentität und aktuelles Arbeitsverzeichnis müssen am Server während einer Sitzung gespeichert sein.

Aufgabe 3 („Wahr oder falsch?“ zu 2.2)

- a.) Nehmen Sie an, ein Nutzer fordert eine Web Seite an, die aus einigem Text und zwei Bildern besteht. Für diese Seite wird der Client einen Request senden und drei Responses empfangen.



Lösung: Falsch. Der Client wird drei HTTP GET Requests senden.

- b.) Zwei verschiedene Web Seiten (z.B. <http://www.swt.hs-mannheim.de/index.html> und <http://www.swt.hs-mannheim.de/vt.html>) können über dieselbe persistente Verbindung geschickt werden.



Lösung: Richtig. Mehrere Web Pages von demselben Server können über eine einzige persistente Verbindung geschickt werden.

- c.) Mit nicht-persistenten Verbindungen ist es möglich, in einem einzigen TCP Segment zwei verschiedene HTTP Request Messages zu übermitteln.



Lösung: Falsch. Bei nicht-persistenten Verbindungen kann nur ein Objekt über die Verbindung angefordert werden, d.h. auch nur ein HTTP Request in einem TCP Segment.



- d.) Das Feld `Date:` im Header der HTTP Response Message zeigt an, wann das Objekt in der Response zum letzten Mal verändert wurde.

Lösung: Falsch. Das Feld `Date:` im Header der HTTP Response Message gibt das Datum der Response an. Das Feld `Last Modified:` gibt an, wann das Objekt in der Response zum letzten Mal verändert wurde.

Aufgabe 4 (Kontrollfragen zu Web Services)

- a.) Können zwei unabhängige Web Services ohne Benutzereinfluss miteinander kommunizieren?



Lösung: Ja, Ketten von Web Service-Aufrufen sind möglich.

- b.) Welche Attribute werden im .NET Framework / C# benutzt, um eine Klasse als Web Service und eine Methode als Web Method zu kennzeichnen?

Lösung: In der Vorlesung wurde spezifisch die Realisierung von Web Services mit der WCF betrachtet. Eine Klasse, die Methoden für andere Rechner zur Verfügung stellt, wird mit dem Attribut `[ServiceContract]` gekennzeichnet. Das `[OperationContract]` - Attribut identifiziert dann eine Methode eines Web Services.

- c.) Welcher Standard-Namensraum wird verwendet, wenn ein ASP.NET Web Service erzeugt wird?

Lösung: Jeder XML Web Service benötigt einen eindeutigen Namensraum, damit Client-Anwendungen ihn von anderen Diensten im Web unterscheiden können. Der Standard-Namensraum bei ASP.NET ist `http://tempuri.org`. Man sollte ihn jedoch ändern, um Namenskollisionen zu vermeiden.

- d.) Welche Rolle spielt SOAP bei Web Services?

Lösung: SOAP ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Method Calls durchgeführt werden können. SOAP stützt sich auf andere Standards: XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht zur Übertragung der Nachrichten. Die gängigste Kombination ist SOAP über HTTP und TCP.

- e.) Ein Web Service benötigt als Input Parameter ein Integer-Array. Welche der drei Protokolle HTTP-GET, HTTP-POST und SOAP eignen sich für die Übergabe eines Arrays?

Lösung: Hier kommt nur SOAP in Frage, da sich Arrays weder in einen URI noch in den Entity Body des HTTP Requests standardmäßig serialisieren lassen.

f.) Welches ist die standardisierte Beschreibungssprache für Web Services?

Lösung: Es handelt sich um die Web Service Description Language (WSDL), eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Web Services zum Austausch von Nachrichten auf Basis von XML.

g.) Warum benötigt ein Web Service Client einen Proxy, um mit einem Web Service zu kommunizieren? Wie wird dieser Proxy erzeugt?

Lösung: Einen Web Service möchte man wie ein lokales Objekt aufrufen. Der Proxy agiert als Stellvertreter für den entfernten Web Service. Er führt die Umwandlung von Datentypen der Programmiersprache in XML-Daten durch, initiiert den Aufruf der Web Service-Methoden und empfängt die Antwort, die als XML-Dokument kodiert ist.

Der Proxy kann aus der in WSDL beschriebenen Funktionalität automatisch generiert werden. Bei ASP.NET Web Services wird für den Proxy eine Klasse erzeugt.

Aufgabe 5 (Kontrollfragen zu 2.6)

a.) Was ist ein Overlay Network? Nehmen Router daran teil? Was sind die Verbindungen in einem Overlay Network?



Lösung: Der Begriff Overlay Network beschreibt ein Netzwerk, das auf ein bestehendes Netzwerk aufsetzt, das sogenannte Underlay. Hauptmerkmale eines Overlay Network sind:

- (logisches) Netz oberhalb existierender Infrastruktur → Menge von TCP-Verbindungen oder UDP-Assoziationen
- oftmals eigener Adressraum mit eigener Adressierung, unabhängig vom Underlay → NodeId bei DHTs (z.B. 128-bit Hash)
- ggf. Einsatz eigener Wegewahlverfahren (z.B. DHT)



Bei einem Overlay Network über das Internet nehmen die Router als Knoten nicht teil.

b.) Wie wird eine Anfrage in einem query-flooding Netzwerk wie Gnutella behandelt? Wie treten Peers einem query-flooding Netzwerk bei?



Lösung: Query – Nachrichten werden in einem query-flooding Netzwerk über existierende TCP-Verbindungen zwischen den Peers weiter geleitet. Ein QueryHit wird bei Gnutella v0.4 über den umgekehrten Pfad zurück gesendet, da die Query-Nachrichten nicht den Identifikator des Suchenden enthalten. Letztere Ineffizienz wurde in Gnutella v0.6 beseitigt.



Der Ablauf beim Beitritt zum query-flooding Netzwerk ist auf Folie 1.2-10 beschrieben.

c.) Inwieweit ist Instant Messaging mit einem zentralen Verzeichnis ein Hybrid von Client-Server und P2P-Architekturen?

Lösung:




- Chat zwischen zwei Nutzern geschieht P2P

Hinweis: In vielen aktuellen Instant Messaging-Anwendungen wie z.B. WhatsApp kommt das Extensible Messaging and Presence Protocol (XMPP, RFC6120-6122)

zum Einsatz. Chats können mit XMPP entweder über einen Server oder über Peer-to-Peer-Sitzungen geführt werden.

- Zentralisierter Dienstbestandteil: Erkennung der Präsenz und des Standorts eines Clients
- Client registriert seine IP-Adresse und Status beim zentralen Server, sobald er online ist
- Client kontaktiert zentralen Server, um die IP-Adresse seiner Freunde herauszufinden

 d.) Betrachten Sie die Verteilung einer Datei der Größe $F=15$ Gbit an N Peers. Der Server hat einen Upstream der Rate $u_s = 30$ Mbit/s und jeder Peer hat einen Downstream der Rate $d_i = 2$ Mbit/s und einen Upstream der Rate u . Erstellen Sie eine Tabelle für $N = 10, 100$ und 1000 und $u = 300$ Kbit/s, 700 Kbit/s, und 2 Mbit/s mit allen Kombinationen von N und u , jeweils für Client-Server und P2P-Verteilung. Errechnen Sie die minimale Dauer für die Verteilung der Datei.

Lösung:

Formel für die minimale Dauer der Verteilung der Datei im Client/Server-System:

$$D_{cs} = \max \{NF/u_s, F/d_{min}\}$$

Formel für die minimale Dauer der Verteilung der Datei im P2P-System:

$$D_{p2p} = \max \{F/u_s, F/d_{min}, NF/(u_s + \sum_{i=1}^N u_i)\}$$

wobei $F = 15$ Gbit

$u_s = 30$ Mbit/s

$d_{min} = d_i = 2$ Mbit/s


Client Server

		N		
		10	100	1000
u	300 Kbit/s	7500s	50000s	500000s
	700 Kbit/s	7500s	50000s	500000s
	2 Mbit/s	7500s	50000s	500000s

Peer to Peer

		N		
		10	100	1000
u	300 Kbit/s	7500s	25000s	45454s
	700 Kbit/s	7500s	15000s	20548s
	2 Mbit/s	7500s	7500s	7500s

e.) Nehmen Sie an, Alice überträgt an Bob mit BitTorrent während einem 30-Sekunden-Intervall Chunks. Wird Bob den Gefallen erwidern und seinerseits Chunks an Alice liefern?

 Lösung: Nicht unbedingt. Alice muss mit ihrer Rate in die Top-4 von Bob gelangen, um mit Sicherheit Chunks von Bob zu bekommen.

- f.) Betrachten Sie einen neuen Peer Alice, der in BitTorrent eintritt, ohne Chunks zu besitzen. Kann sie ohne Chunks in die Top-Four Uploader der anderen Peers gelangen? Wie bekommt Alice ihren ersten Chunk?



Lösung: Nein, Alice kann zunächst nicht in die Top-Four Uploader der anderen Peers gelangen. Sie muss warten, bis sie von Bob zufällig ausgewählt wird und als „optimistically unchoked“ Peer die ersten Chunks bekommt.

- g.) Nehmen Sie an, Bob tritt einem BitTorrent Torrent bei, aber er möchte nichts an andere Peers abgeben („free-riding“). Bob behauptet, er kann eine komplette Kopie der Datei erhalten, die von dem Schwarm geteilt wird. Ist diese Behauptung richtig? Warum?



Lösung: Ja, diese Behauptung ist richtig, solange sich viele Peers für eine lange Zeit in dem Schwarm aufhalten. Bob kann dann durch „optimistic unchoking“ anderer Peers alle Chunks einer Datei bekommen.

- h.) Bob behauptet außerdem, dass er sein „free-riding“ effizienter gestalten kann, indem er eine Gruppe von Computern mit unterschiedlichen IP-Adressen im Labor seiner Fakultät dafür benutzt. Wie kann das funktionieren?



Lösung: Ja, auch diese Behauptung ist richtig. Er kann auf jedem Rechner einen „free-riding“ Client laufen lassen und die gesammelten Chunks in einer einzigen Datei zusammen führen. Er könnte sogar ein kleines Planungsprogramm schreiben, um die verschiedenen Rechner unterschiedliche Chunks der Datei anfordern zu lassen. Dies entspricht einer Art von Sybil – Attacke in P2P-Netzwerken.

Anmerkung: Der Roman „Sybil“ von Flora Rheta Schreiber (1973) beschreibt das Leben von Sybil Dorsett, einer Frau mit multipler Persönlichkeitsstörung (sh. Wikipedia für Details).

- i.) Betrachten Sie ein Overlay Network mit N aktiven Peers, bei dem jedes Paar von Peers eine aktive TCP-Verbindung hat. Nehmen Sie zusätzlich an, dass die TCP-Verbindungen alle zusammen M Router durchqueren. Wie viele Knoten und Kanten gibt es im Overlay Network?



Lösung: Das Overlay Network hat N Knoten. Zur Vollvermaschung werden $N(N-1)/2$ Kanten benötigt.

- j.) Betrachten Sie eine vermaschte Overlay-Topologie, d.h. jeder Peer behält die Übersicht über alle anderen Peers im System. Was sind die Vor- und Nachteile eines derartigen Entwurfs? Was sind die Vor- und Nachteile von kreisförmigen („circular“) DHTs ohne Shortcuts?

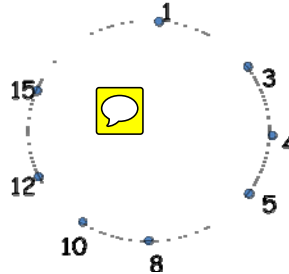


Lösung: In Peer-to-Peer-Systemen muss ein Kompromiss gefunden zwischen der Anzahl Peers, die jeder Knoten kennen muss und der Anzahl Nachrichten, die versendet werden muss, bis ein Query aufgelöst wird.

Eine vermaschte Overlay-Topologie ist nicht praktikabel für Millionen von Peers. Dafür wird ein Query aber mit einer einzigen Nachricht aufgelöst. Bei kreisförmigen DHTs ohne Shortcut muss ein Peer nur zwei Nachbarn kennen. Dafür wächst aber die Anzahl

benötigter Nachrichten, um einen Query aufzulösen, proportional mit der Anzahl N der Peers.

- k.) Betrachten Sie folgende ringförmige DHT. Jeder Peer kennt seine zwei Vorgänger- und zwei Nachfolger-Knoten.



Peer 3 bemerkt, dass Peer 5 den Ring verlassen hat („peer churn“), da Peer 5 auf Ping nicht mehr reagiert. Wie aktualisiert Peer 3 die Zustandsinformation über seine zwei Nachfolger? Welcher Peer ist dann sein zweiter Nachfolger?



Lösung: Peer 3 befragt seinen ersten Nachfolger Peer 4 nach dem Identifier seines unmittelbaren Nachfolgers und erhält Peer 8. Danach macht Peer 3 den Peer 8 zu seinem zweiten Nachfolger.

Peer 3 weiß übrigens, dass Peer 5 ursprünglich der erste Nachfolger von Peer 4 war, daher würde Peer 3 warten, bis Peer 4 seinen ersten Nachfolger aktualisiert hat.

- l.) Nehmen Sie an, der neue Peer 6 möchte der kreisförmigen DHT aus Aufgabe k.) beitreten und kennt initial nur die IP-Adresse von Peer 15. Welche Schritte laufen ab?



Lösung: Peer 6 würde zuerst an Peer 15 eine Nachricht senden mit der Anfrage, wer Vorgänger und Nachfolger von Peer 6 in der DHT sein werden. Diese Anfrage wird durch die DHT bis zu Peer 5 weiter geleitet. Peer 5 merkt, dass er der Vorgänger von Peer 6 sein wird und dass sein aktueller Nachfolger, Peer 8, der Nachfolger von Peer 6 sein wird. Als Nächstes sendet Peer 5 diese Vorgänger- und Nachfolger-Information zurück an Peer 6. Peer 6 kann nun der DHT beitreten, indem er Peer 8 zu seinem Nachfolger macht und Peer 5 mitteilt, dass er Peer 6 als seinen unmittelbaren Nachfolger eintragen soll.

- m.) Betrachten Sie eine DHT mit Knoten- und Schlüssel-Identifikatoren im Bereich $[0,63]$.



Nehmen Sie an, es gibt 8 Peers mit Identifikatoren 0, 8, 16, 24, 32, 40, 48, und 56.

Nehmen Sie an, jeder Peer hat einen Shortcut Peer. Bestimmen Sie für jeden der 8 Peers den Shortcut Peer, so dass die Anzahl Nachrichten bei beliebigen Anfragen eines beliebigen Peers minimiert wird.

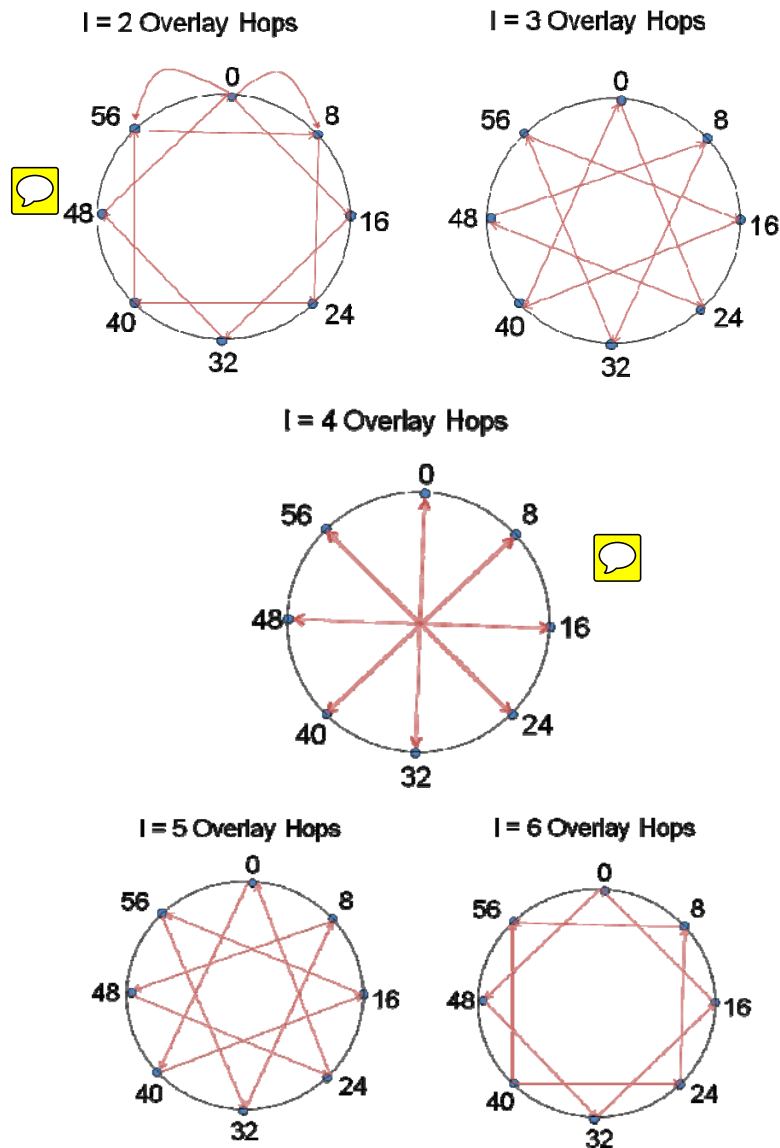
Dabei gelten folgende Annahmen über Schlüssel und Anfragen:

- Die Schlüssel sind gleichverteilt im Schlüsselbereich, und alle 8 Peers sind im Schnitt verantwortlich für die gleiche Anzahl Anfragen
- Die Anfragen nach Schlüsseln sind gleichverteilt im Schlüsselbereich. D.h. eine Anfrage für einen beliebigen Schlüssel kommt mit der gleichen Wahrscheinlichkeit zustande

Ohne Verlust der Allgemeingültigkeit kann man die Betrachtung auf einen Peer, z.B. Peer 0, beschränken. Betrachten Sie alle Möglichkeiten: Peer 0 verwaltet Shortcuts zu

einem Peer, der zwei Overlay Hops im DHT ID Ring entfernt ist, oder drei Overlay Hops entfernt, oder vier usw.

Vervollständigen Sie die unten stehende Tabelle. Der beste Shortcut ist derjenige, bei dem die durchschnittliche Anzahl Nachrichten pro Anfrage am Geringsten ist.



Anfragen für Schlüssel im Bereich		[1,8]	[9,16]	[17,24]	[25,32]	[33,40]	[41,48]	[49,56]	[57,0]	Gesamtzahl Nachrichten
Anzahl Overlay Hops bei Shortcutting mit Entfernung l zum Shortcut	2	1	1	2	2	3	2 (3)	1	0	12
	3	1	2	1	2	3	2	1	0	12
	4	1	2	2 - 3	1	2	2 - 3	1	0	12
	5	1	2	3	2	1	2	1	0	12
	6	1	2	3	2 - 4	2	1	1	0	13
Lösung: Die Gesamtzahl benötigter Nachrichten ist mindestens 12, wenn jeder Peer einen Shortcut mit einem anderen Peer verwaltet.										

- n.) Da DHTs Overlay Networks sind, ist ihre Performanz nicht wirklich optimiert, da in dem physikalischen Underlay Network zwei benachbarte Knoten sehr weit auseinander sein können. Z.B. könnte ein Peer in Asien sein und sein Nachbar in Nordamerika. Wenn wir zufällig und gleichmäßig die Identifikatoren auf neue beitretende Peers verteilen, kann dies durchaus problematisch sein. Erklären Sie warum. Wie wird sich dies auf die Performanz der DHT auswirken?



Lösung: Korrekt, die Zuweisung von Schlüsseln an Peers bezieht das Underlay Network nicht in Betracht, sodass sehr wahrscheinlich Diskrepanzen auftreten, die die Suchleistung erheblich verschlechtern.

Es kann z.B. einen logischen Pfad $P1: A \rightarrow B \rightarrow C$, bestehend aus zwei logischen Links geben. Gleichzeitig gebe es einen Pfad $P2: A \rightarrow D \rightarrow E \rightarrow C$, bestehend aus drei logischen Links.

Es kann sein, dass sowohl A und B als auch B und C physikalisch sehr weit auseinander liegen. Aber A, D, E und C könnten physikalisch sehr nah beieinander liegen. In anderen Worten, ein kurzer logischer Pfad kann einem längeren physikalischen Pfad entsprechen.

- o.) Skype nutzt P2P-Techniken für zwei wichtige Funktionen. Welche sind dies?

Lösung:



- Speichern und Suchen von IP-Adressen der eingeloggtten Peers mit DHT (User Location)
- Durchquerung der Network Address Translation mit Relay Nodes (NAT Traversal)

- p.) Ist es möglich, Live TV (IPTV) mit einer P2P-Architektur zu verteilen? Überlegen Sie einen Ansatz. Können Sie eine entsprechende Software nennen?



Lösung: Ja, wobei aber ggf. beträchtliche Verzögerungen des Live-Signals akzeptiert werden müssen. Ähnlich wie bei BitTorrent können die Peers Chunks der z.B. letzten 300 Sekunden eines Live Streams austauschen. Ältere Chunks werden gelöscht. Eine einzige Quelle muss das Live-Signal in Chunks zerlegen und an die Peers verteilen.

Beispielhafte Implementierungen sind PPLive und Zattoo.