Prüfungsprotokoll Vertiefungsgebiet Kommunikationsnetze

Ewgenij Sokolovski

31. August 2008

Hier folgt das Protokoll meiner Vertiefungsprüfung bei Professor Spaniol. Geprüft wurden die Fächer: Datenkommunikation, Verteilte Systeme, Modellierung und Bewertung von Kommunikationssystemen sowie Sicherheit in Kommunikationsnetzen. Die Prüfung der Vorlesung „Verteilte Systeme“ hat Prof. Spaniol dem Beisitzer, Dr. Dirk Thissen, überlassen.

Die Prüfung fand am 29.08.2008 statt. Beginn 10:55 - Ende 11:25. Die Prüfung wurde mit der Note 1.0 bewertet.

1 Sicherheit in Kommunikationsnetzen

Spaniol: Zu Beginn die Sicherheit. Was sagt Ihnen der Begriff DES?
Ich: Es ist ein symmetrisches Verschlüsselungsverfahren. Der Schlüssel hat eine Länge von 56 Bits.
Spaniol: (unterbricht) Aktive Schlüsselbits.
Ich: Ja, in Wirklichkeit sind es 64 Bits, es werden aber nur 56 benutzt.
Spaniol: (unterbricht) Die anderen dienen der Sicherheit der Schlüsselübertragung.
Ich: Ja, genau. Mit diesem Schlüssel können dann Textblöcke verschlüsselt werden, die dann an den anderen Kommunikationspartner gesendet werden.
Spaniol: Warum kritisiert man DES eigentlich?
Ich: Weil die Schlüssellänge von 56 Bits dazu führt, dass DES heutzutage problemlos durch einen Brute-Force-Angriff gebrochen wird.
Spaniol: Also problemlos nicht, aber mit einem nicht sehr hohen Aufwand. Was gibt es noch für Kritikpunkte?
Ich: Da gibt es noch die S-Boxen, mit denen verschlüsselt wird. Man weiß nicht so genau, warum sie so und nicht anders gewählt wurden. Deswegen gibt es Vermutungen, dass da vielleicht eine Hintertür eingebaut wurde.
Spaniol: Na ja, es wurde schon durch mehrere Behörden herausgearbeitet und standardisiert, es würde auffallen, wenn da etwas nicht stimmen würde.
Ich: Ja, aber wenn man nichts gefunden hat, heißt es ja noch nicht, dass es nichts gibt. Und solange die Kriterien und Richtlinien für die Boxen nicht offen gelegt wurden, kann man alles vermuten. Es hat sich allerdings herausgestellt, dass die S-Boxen eine gute Verschlüsselungsqualität haben, so wie sie sind. (Ich hab das Gefühl, dass Prof. Spaniol auf etwas Anderes und nicht auf die S-Boxen eigentlich hinaus wollte, weiß aber nicht auf was)
Spaniol: Sie haben da Varianten von DES erwähnt, die verwendet werden, weil

DES an sich heutzutage gebrochen werden kann.
Ich: Ja, da gibt es das Triple-DES.
Spaniol: Was heißt denn hier „Triple“? Wird da drei Mal verschlüsselt oder einmal verschlüsselt mit einem längeren Schlüssel?
Ich: Es wird drei Mal verschlüsselt, jedes Mal mit einem anderen Teilschlüssel.
Spaniol: Müssen denn alle diese Teilschlüssel verschieden sein?
Ich: Ja.
Spaniol: Nein, die zwei äußeren Teilschlllüssel des Schlüssels können auch gleich sein, aber nicht so wichtig. Es gibt auch andere Alternativen zu DES...
Ich: Ja, da gibt es auch IDEA.
Spaniol: Oder AES.
Ich: Ja, AES.
Spaniol: Beschreiben Sie mal, wie IDEA funktioniert.
Ich: IDEA ist auch ein symmetrisches Verschlüsselungsalgorithmus. Der Schlüssel ist 128 Bit lang, was einen Brute-Force-Angriff unmöglich macht. Mit diesem Schlüssel werden dann die Textblöcke verschlüsselt.
Spaniol: (unterbricht) Warum hat sich denn IDEA nicht durchgesetzt?
Ich: Ähhmm, es hat sich doch durchaus durchgesetzt.
Spaniol: Nee, nicht wirklich. Da gibt es noch Marktdominanz von DES, hat Standardisierungsgründe und welche Organisation welches Verfahren durchsetzt... Sagen Sie, wenn man verschlüsselt, dann verschlüsselt man Textblöcke einzeln mit einem Verfahren, dann haben gleiche Originaltextblöcke auch gleiche verschlüsselte Entsprechungen. Daraus kann der Angreifer dann Informationen gewinnen, z.B. über Tabellen.
Ich: Ja, das kann passieren, wenn man den Text direkt in Blöcke unterteilt und sie einzeln verschlüsselt. Das ist aber eine naive Vorgehensweise. Es gibt bessere Methoden. Zum Beispiel man verschlüsselt den ersten Block und XORt das Ergebnis mit dem zweiten Textblock, bevor man ihn seinerseits verschlüsselt. Und so weiter (Schlüsselwort: Cipher Block Chaining(CBC)). Dann wird kein verschlüsselter Block gleich dem Anderen sein.
Spaniol: Und was ist mit dem ersten Textblock? Womit kann man ihn verknüpfen?
Ich: Man kann eine Zufallszahl generieren, eine Nonce.
Spaniol: Oder man kann eine Zahl vorher vereinbaren, es gibt verschiedene Möglichkeiten. Sagen Sie, wenn wir kommunizieren möchten, dann müssen wir ja einen Schlüssel für die Verschlüsselung der Kommunikation vereinbaren. Wie macht man das?
Ich: Man macht das mit der Public-Key-Kryptographie.
Spaniol: (unterbricht) Und wie genau?
Ich: Es gibt dafür das Verfahren von Diffie-Hellmann, man...
Spaniol: (unterbricht) Ja, Diffie-Hellmann. Da hat man einen öffentlichen Schlüssel und einen Privaten, da wählt man eine Zahl...
Ich: (unterbreche) Ja, und setzt dann dieses g hoch diese Zahl
Spaniol: (wirft ein) *mod p*
Ich: Ja, genau, *mod p*, wobei p eine Primzahl ist, und der Partner wählt auch eine Zahl und setzt g hoch diese Zahl
Spaniol: (wirft ein) und dann tauschen sie diese g^a bzw. g^b aus.
Ich: und der eine setzt das, was er bekommen hat, hoch die von ihm gewählte Zahl *mod p* und der Andere macht dasselbe. Und dann haben sie einen gemeinsamen Schlüssel.

Spaniol: Auf welchem Problem basiert dieses Verfahren denn?
Ich: Auf dem Problem des diskreten Logarithmus, dass man also nicht a berechnen kann, wenn man $g^a \ mod \ p$ kennt.
Spaniol: Ist das denn sicher, dass man es nicht kann?

Ich: Also es ist nicht bewiesen worden, allerdings hat es noch bis jetzt niemand geschafft, dieses Problem zu lösen. Wenn es mal irgendeinem Schlaukopf gelingen sollte, dann hat man natürlich Pech.
Spaniol: Man muss nicht nur die Kommunikation verschlüsseln können. Öfters müssen die Kommunikationspartner sich vor dem Beginn der Kommunikation gegenseitig authentifizieren.
Ich: Ja, und das kann man ebenfalls mithilfe der Public-Key-Kryptographie machen. Da gibt es zum Beispiel die Möglichkeit mit RSA. Derjenige, der sich authentifizieren möchte, signiert eine Nachricht, zum Beispiel den aktuellen Zeitstempel, mit seinem privaten Schlüssel und schickt sie an seinen Kommunikationspartner. Der Partner überprüft dann die Signatur mit dem publizierten öffentlichen Schlüssel.
Spaniol: Also man hätte dann den Aufwand, dass man zwei Mal verschlüsseln müsste.
Ich: ??? Wieso?
Spaniol: Man muss ja ein Mal mit dem privaten Schlüssel signieren und dann die ganze Signatur mit dem öffentlichen Schlüssel des anderen Kommunikationspartners verschlüsseln, um die Daten auch sicher zu übertragen.
Ich: Also man muss die ganze Nachricht gar nicht noch ein Mal verschlüsseln. Wenn nur ein Zeitstempel signiert wurde, gibt es da ja nichts Geheimes. Die Nachricht kann unverschlüsselt gesendet werden, der Gegenüber soll nur die Signatur verifizieren.
Spaniol: Aber wenn man schon mit der ersten Nachricht einen Teil der eigentlichen Information übertragen möchte, dann muss man trotzdem die Authentifikationsnachricht noch Mal verschlüsseln.
Ich: Ja, OK, in diesem Fall schon. Dann kommt in der Tat der Aufwand der doppelten Verschlüsselung. Es gibt aber auch andere Methoden der gegenseitigen Authentifikation. Zum Beispiel der Lamports Hash.
Spaniol: Oh ja, dann erzählen Sie mal davon.
Ich: (hab dann das Verfahren mit dem Lamports Hash erklärt)
Spaniol: Es gibt da eine so genannte small n Attack...
Ich: Ja, (habe dann diesen Angriff erläutert)

2 Modellierung und Bewertung von Kommunikationssystemen

Spaniol: Wenn wir über Kunden in einem System sprechen, dann gibt es da mehrere Möglichkeiten, wie sie im System ankommen. Die einfachste Annahme ist die Poissonverteilung. Was können Sie dazu sagen?
Ich: Ja, man nimmt an, dass die Kunden mit einer Rate λ am System ankommen. Und sie kommen unabhängig voneinander ab. Dann spricht man von einer

Poissonverteilung.
Spaniol: Also sie kommen linear mit der Zeit an.
Ich: Ja, soll ich die Formel für die Poissonverteilung aufschreiben?
Spaniol: Ja, wenn Sie sie auswendig können.
Ich: (Hab dann die Formel auf einem Blatt Papier niedergeschrieben. Das war auch das einzige Mal, wo ich während der gesamten Prüfung etwas schreiben musste.)
Spaniol: Wir haben verschiedene Wartesysteme. Da gibt es zum Beispiel das $M/M/1$-System. Da kommen die Kunden poissonverteilt an.
Ich: Ja, und sie werden auch poissonverteilt abgearbeitet, also bezüglich der Bedienzeiten. Und es gibt einen Bediener.
Spaniol: Wie analysiert man denn ein solches System?
Ich: Öhhmmm (wusste nicht genau, was er denn jetzt von mir will), was meinen Sie jetzt?
Spaniol: Also der Zustand eines Wartesystems ist als die Anzahl der Kunden in diesem System definiert...
Ich: Ah so, ja, die Analyse besteht darin, dass man die Wahrscheinlichkeiten für den jeweiligen Zustand ermittelt. Also die Wahrscheinlichkeit, dass sich zu einem bestimmten Zeitpunkt eine bestimmte Anzahl der Kunden in dem betrachteten Wartesystem befindet.
Spaniol: Haben Sie die Zustandswahrscheinlichkeiten für $M/M/1$ im Kopf?
Ich: Ja, die Formel lautet $p_i = (1 - \rho) \cdot \rho^i$
Spaniol: Da gibt es bei den Verteilungen die Memory-Less Eigenschaft, was ist das?
Ich: (Hab es dann erklärt)
Spaniol: Und was gilt für die Zwischenankunftszeiten bei einem Poissonprozeß?
Ich: Sie sind dann exponentialverteilt.
Spaniol: Die Memory-Less Eigenschaft gilt nicht mehr bei dem $M/G/1$-System.
Ich: Ja, hier sind die Bedienzeiten nicht mehr exponentialverteilt, sondern es gilt eine allgemeine Verteilung, die nicht unbedingt memory-less ist.
Spaniol: Wie kann man dann ein solches System analysieren?
Ich: Man verwendet dafür eingebettete Markov-Ketten. Da die Bedienzeiten nicht mehr memory-less verteilt sind, kann man die übliche Analyse nicht anwendet. Deswegen betrachtet man nur die Zeitpunkte unmittelbar nachdem ein Kunde das System verlässt. Diese Zeitpunkte ergeben eine homogene Markov-Kette, eine eingebettete markov-Kette. Und somit kann man sie mit den üblichen Methoden analysieren.
Spaniol: Man könnte auch andere Zeitpunkte betrachten, zum Beispiel die unmittelbar vor der Neuankunft eines Kunden.
Ich: Ja, das würde auch gehen.
Spaniol: Was hätte das denn für einen Nachteil?
Ich: Ähh (kein Plan)...
Spaniol: Ja, der Nachteil wäre, dass das System nie leer sein würde. OK, bei dem $G/M/1$-System sähe alles umgekehrt aus.
Ich: Ja, da sind die Ankünfte allgemein verteilt, also nicht unbedingt memory-less, während die Bedienzeiten exponentialverteilt sind.
Spaniol: Wie kommt man auf die Wahrscheinlichkeitsformel für dieses System?
Ich: ??? Wie jetzt?
Spaniol: Durch Lösen eines Gleichungssystems.
Ich: Ah ja, natürlich.

Spaniol: OK. Außer den Wartesystemen an sich gibt es noch zusammengekoppelte Wartesysteme.
Ich: Die Wartenetze.
Spaniol: Ja, und die können verschiedener Art sein, bei manchen kommen die Kunden von außerhalb an und verlassen dann das System irgendwann, während bei anderen die Kunden immer im Netz bleiben.
Ich: Ja, das sind die offenen und die geschlossenen Wartenetze.
Spaniol: Ja, bei den offenen gibt es die Jacksonnetze. Wie wird der Zustand davon charakterisiert?
Ich: Der Zustand eines Wartenetzes ist die aktuelle Verteilung der Kunden an seinen Komponenten – den einzelnen Wartesystemen.
Spaniol: Was ist so Besonderes an der Berechnung der Zustandswahrscheinlichkeiten?
Ich: Sie werden nach der Produktform berechnet. Also man betrachtet einzelne Komponenten, die Wartesysteme, isoliert.
Spaniol: Ja, die Produktform und die Isolation. So, außer den Jackson-Netzen und den Gordon-Newel-Netzen gibt es noch andere Netze, die auch mit der Produktformel berechnet werden? Allgemeinere Netze?
Ich: Ja, die BCMP-Netze, sie sind allgemeiner und...
Spaniol: (unterbricht) Ja, OK, dann erzählen Sie mir davon.
Ich: Die BCMP-Netze sind allgemeine Netze, die mit der Produktform zu analysieren sind. Die Verallgemeinerung verglichen mit den Jackson- oder Gordon-Newel-Netzen ist die Einführung von verschiedenen Kundenklassen, verschiedenen Bedienzeitverteilungen und Abfertigungsverfahren außer FIFO. Und die Bedienzeiten werden mit Cox-Verteilungen beschrieben, die...
Spaniol:(unterbricht) Aber es sind nicht alle Abfertigungsstrategien mit allen Bedienzeitverteilungen kombinierbar.
Ich: Ja, die FIFO-Strategie ist mit einer Cox-Verteilung nicht kombinierbar, dann muss man die Exponentialverteilung nehmen.
Spaniol: Ja, OK. Ich übergeben jetzt an Herrn Thissen.

3 Verteilte Systeme

Thissen: Wir hatten in Verteilten Systemen das Thema „Replikation“. (Oh, Mist, das hab ich nur oberflächlich gelernt, kam in keinem Protokoll vor) Welche Arten davon kennen Sie?
Ich: Also es gibt die primäre Replikation, wo ein Replikatorserver im Hintergrund steht und nur dann einspringt, wenn der arbeitende Server ausfällt.
Thissen: Es können auch mehrere Replikatorserver sein.
Ich: Ja, es können auch mehrere sein. Dann gibt es noch die distributive Replikation. Dabei stehen mehrere arbeitende Server, die alle dieselben Datensätze haben. Sie beantworten dann die Anfragen. Die Antwort liefert immer der Server, der zurzeit am wenigsten belastet ist.
Thissen: Nur einer von denen?
Ich: Ja, derjenige mit der geringsten Last, oder nach anderen Kriterien ausgesucht.
Thissen: Also das wäre dann eine spezielle Implementierung, normalerweise wird die Anfrage an alle geschickt...

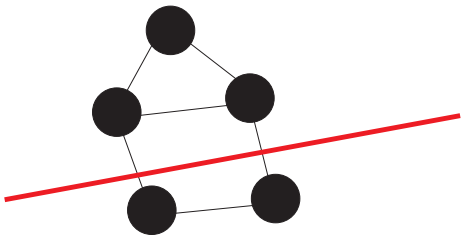

Abbildung 1: Das getrennte Netzwerk

Ich: Ja, dann wird durch Mehrheitsbeschluss entschieden, welche von diesen Antworten an den Kunden zurückgeliefert wird. Also, wenn 4 Server eine 1 zurückliefern, und 2 – eine 0, dann wird eine 1 an den Kunden zurückgegeben.
Thissen: Also ein Quorumentschluss. Das ist dann eine aktive Replikation.
Ich: Ja, genau.
Thissen: Und wenn wir jetzt mehrere Server mit Datenkopien haben, dann müssen wir dafür sorgen, dass sie alle den gleichen Inhalt aufweisen. Wie macht man das?
Ich: Die Server tauschen regelmäßig Nachrichten aus, die die neuesten Updates enthalten.
Thissen: Regelmäßig heißt was? Wie oft?
Ich: Na ja, so oft wie möglich bzw. sobald ein Update auf einer Datenkopie passiert, muss eine entsprechende Nachricht an alle anderen Server geschickt werden.
Thissen: Es gibt ja mehrere Arten von Konsistenz: Strikte Konsistenz, Lineare Konsistenz, Sequentielle Konsistenz, Kausale Konsistenz... Welche Konsistenz haben wir in diesem Fall?
Ich: Die strikte Konsistenz (na ja, mehr geraten als gewusst.).
Thissen: Ja. Wie ist es denn, wenn wir ein Netzwerk von Replikatoren haben und die Verbindung zwischen zwei Teilen davon kaputtgeht, wie können wir sicherstellen, dass am Ende trotzdem alle Server konsistente, also dieselben Datensätze haben.
Ich: ??? Wie meinen Sie das jetzt?
Thissen: (malt die Abbildung 1 auf) Es war mal ein Netzwerk und jetzt sind da einige Verbindungen ausgefallen, wo die rote Linie verläuft. Danach wird auf jedes Teil des Netzwerkes von Kunden zugegriffen. Wie stellt man die Konsistenz wieder her?
Ich: (bin ziemlich ratlos) Ähh... Also man darf dann nur lesende Zugriffe zulassen, damit die Datensätze auf allen Replikatorservern gleich bleiben.
Thissen: Nein, es müssen auch Schreibzugriffe zugelassen werden.
Ich: Also wenn die Verbindungen nicht repariert werden, dann kann man ja in diesem Fall keine Konsistenz gewährleisten.
Thissen: Nein, später werden die Verbindungen repariert, und danach muss die Konsistenz sichergestellt werden.
Ich: Also man kann jeweils an den Servern Updatelogs führen und später anhand dieser Logs die Konsistenz wiederherstellen.
Thissen: Nee, das geht ja nicht. Wenn die jeweiligen Transaktionen ihre Veränderungen an den Datenbanken schon committed haben, dann kann man nichts mehr verändern.

Ich: Hmm (je weiter desto verwirrter wurde ich), man könnte jetzt keine Commits zulassen, solange die Verbindungen zwischen den Replikatoren nicht wiederhergestellt sind... Wollen Sie jetzt auf die Serialisierbarkeit von Transaktionen hinaus?
Thissen: Nein, darauf will ich gar nicht hinaus. Sie haben dieses geteilte Netzwerk, was können Sie machen, damit die Konsistenz nach der Reparatur der Verbindungen wiederhergestellt werden würde?
Ich: (also da stand ich schon total auf dem Schlauch:))) Nee, das weiß ich nicht.
Thissen: Sie müssen dann einfach wieder per Quorum entscheiden.
Ich: Ah soooo, dann guckt man einfach, dass drei der Replikatoren den Wert X haben, und die anderen zwei – den Wert Y, und dann entscheidet man sich auf Grund des Mehrheitsbeschlusses, dass alle Replikatoren den Wert X übernehmen sollen.
Thissen: Genau. (übergibt wieder an Spaniol)

4 Datenkommunikation

Spaniol: Es gibt mehrere Verfahren der Repräsentation von Bits. Da gibt es die NRZ. Was ist das?
Ich: Das ist ein Verfahren, bei dem weder 1 noch 0 durch eine Nullspannung dargestellt werden, sondern die 1 durch die positive und die 0 durch die negative Spannung.
Spaniol: Und was für Probleme können da auftauchen?
Ich: Durch eine lange 0- oder 1-Folge kann der Takt verloren gehen, der Empfänger weiß dann nicht mehr, wie viele 1 oder 0 er jetzt denn bekommen hat.
Spaniol: Da gibt es dann eine differentielle Variante von...
Ich: Ja, dabei wird die 1 als Spannungssprung dargestellt.
Spaniol: Kann es dabei trotzdem zu Gleichstrom kommen?
Ich: Ja. **Spaniol:** (unterbricht) Genau, wenn eine zu lange 0-Folge vorkommt. Wir haben mal den Manchester-Code in der Vorlesung besprochen. Was ist das?
Ich: Das ist eine Kodierung auf der Ebene der physikalischen Schicht. Die 0 und die 1 werden als Spannungssprünge dargestellt.
Spaniol: Und welchen Vorteil hat der Code?
Ich: Die Taktung wird mit übertragen. In der Mitte eines jeden Bits erfolgt ein Spannungssprung. Damit kann sich dann der Empfänger immer synchronisieren.
Spaniol: Es gibt noch eine differentielle Variante davon...
Ich: Ja, den Differentielle Manchester Code. Hier geschieht ein Spannungssprung nur bei einer 1, bei einer 0 bleibt der Spannungslevel gleich. Die Taktsprünge sind wie bei dem normalen Manchester Code.
Spaniol: Genau. Wie ist es mit dem Gleichstrom zum Beispiel bei dem $4B/5B$ Code?
Ich: Bei dem $4B/5B$ werden 4 Bits durch 5 Bits kodiert.
Spaniol: (wirft ein) Und wie?
Ich: Mit 5 Bits hat man 32 Kombinationen, daraus wählt man 16, um die 4-Bit-Zahl darzustellen.
Spaniol: Kann es dabei zu Gleichstrom kommen?
Ich: Nein, da die 16 Kombinationen so ausgewählt sind, dass es nicht passiert.
Spaniol: Wie viele Nullen können denn da nacheinander auftreten?
Ich: Öhhmm, 4.

Spaniol: Nein, es sind 3, so sind die Kombinationen ausgewählt. Wie kann man denn eventuelle Fehler bei dem Manchester Code entdecken?
Ich: ??? Ähh, wie meinen Sie das?
Spaniol: Ja, bei der Übertragung können ja Fehler auftreten, wie kann man sie entdecken?
Ich: Mit dem Manchester Code???
Spaniol: Ja, es gibt ja die Parität...
Ich: Ah so, sie meinen, wie man überhaupt Übertragungsfehler entdecken kann! Ich dachte, Sie meinten das in Bezug auf den Manchester Code.
Thissen, Spaniol: Nein, Nein, überhaupt:)))
Ich: OK, da gibt es ein Mal die Paritätsbits. Man definiert, dass die Anzahl von z.B. Einsen in einer Nachricht gerade sein soll, und fügt dementsprechend eine 1 oder eine 0 ans Ende der Nachricht hinzu. Damit kann man dann Einbitfehler feststellen.
Spaniol: Ja, und Dreibitfehler auch.
Ich: Ja, eigentlich jede ungerade Anzahl von Fehlern. Dann gibt es noch den Hamming-Code...
Spaniol: (unterbricht) Ja, damit kann man sogar die Fehler korrigieren.
Ich: Ja, das ist ein Korrekturcode.
Spaniol: OK, und da gibt es noch die zweifache Parität...
Ich: Ja, in diesem Fall ordnet man eine Nachricht in Form einer Matrix an und fügt unten sowie an der Seite Paritätbits hinzu, die dann für die jeweilige Zeile bzw. Spalte zuständig sind. Damit kann man Einbitfehler nicht nur feststellen, sondern auch korrigieren.
Spaniol: Und was ist, wenn mehrere Bits korrupt sind?
Ich: Dann geht das nicht.
Spaniol: Oder man kann sich sogar falsch korrigieren. OK, sagen Ihnen die Begriffe Differentiated Services und Integrated Services etwas?
Ich: Ja, das bezieht sich auf Quality of Service. Bei Differentiated Services geht es darum, dass verschiedene Dienste wie Video, Audio, E-Mail usw. verschiedenen Bedienklassen zugeordnet werden, ihre Datenpakete werden dementsprechend markiert. Abhängig von der Markierung wird dann jeweils entschieden, welche Ressourcen dem Dienst zur Verfügung gestellt werden und wie die entsprechenden Datenpakete behandelt werden.
Spaniol: (unterbricht) Wissen Sie, was MPLS ist?
Ich: Ja, bei diesem Verfahren werden die Datenpakete markiert, und anhand der Markierungen werden dann Routen gewählt. Die Pakete mit derselben Markierung werden immer entlang derselben Route übertragen.
Spaniol: Bleiben die Markierungen dabei während der gesamten Übertragung unverändert?
Ich: Nein, sie werden von jedem Router auf dem Pfad verändert.
Spaniol: Warum?
Ich: Damit später zwischen den Paketen unterschieden werden kann, welche wohin müssen.
Spaniol: Wo werden denn die Markierungen platziert?
Ich: Sie werden vor dem IP-Header angebracht.
Spaniol: Jetzt an der äußeren oder an der inneren Seite, also ist das die Schicht 3.5 oder 2.5?
Ich: 2.5.
Spaniol: Wie kann der Datenstrom reguliert werden?

Ich: Dafür gibt es Konzepte wie Leaky Bucket oder Token Bucket.
Spaniol: Während Datenpakete geroutet werden, können unter Anderem wegen dieser Konzepte manche davon verloren gehen. Wie stellt man so etwas fest?
Ich: Jedes Paket wird mit einem ACK bestätigt. Wenn irgendein ACK fehlt, dann wurde das entsprechende Paket nicht zugestellt.
Spaniol: Dann hat man die Nummer des verloren gegangenen Pakets, und ab wo müssen dann die Pakete neu übertragen werden?
Ich: Man muss dann ab dieser Nummer alle Pakete neu übertragen.
Spaniol: Wie kann man es anders machen?
Ich: Meinen Sie jetzt das „Fast Retransmit“?
Spaniol: Nein, das meine ich nicht.
Ich: Man kann nur das nicht zugestellte Paket noch Mal übertragen. Allerdings braucht man dann auf der Empfängerseite einen Puffer, wo er die empfangenen Pakete zwischenspeichert...
Spaniol: (unterbricht) und das nachgesendete reinsortiert.
Ich: Ja, er muss dann in diesem Puffer die entsprechenden Algorithmen anwenden können.
Spaniol: OK, an diesem Punkt beende ich die Prüfung.

Irgendwo dazwischen fragte Spaniol, ob ich eine entsprechende Anwendung aus Schicht 7 kenne, ich nannte dann Outlook Express. Ich weiß aber nicht mehr, worauf er das jetzt bezog.

5 Anmerkungen

Ich fand, dass Herr Spaniol ein sehr netter Prüfer ist. Wie auch andere Leute habe ich zuvor Gerüchte gehört, dass er angeblich launisch sein soll. Was an diesen Gerüchten wahr ist, vermag ich nicht zu urteilen. Ich kann nur für meine Prüfung sagen, dass es nicht der Fall war. Prof. Spaniol ist sehr freundlich, gibt Hilfestellungen und drängt einen nicht in die Ecke. Ich kann ihn auf jeden Fall als Prüfer empfehlen. Die Prüfungsathmosphäre war sehr angenehm und locker. Insgesamt ging die Prüfung in die Breite und nicht in die Tiefe. Details wurden kaum gefragt. Man musste viel eher ein gutes Gesamtwissen, Gesamtverständnis von der Materie haben.

Was mir auffiel, war, dass im unterschied zu anderen Prüfern, Prof. Spaniol sehr gerne selber redet und viel erzählt. Beziehungsweise er unterbricht einen oft und erzählt selber weiter. Unangenehm ist es aber nicht. Man muss nur aufpassen, dass man nach Möglichkeit immer wieder etwas vom eigenen Wissen einwirft, während er erzählt, sodass er sehen kann, dass man sich auskennt.

Diese Prüfung war meine letzte Prüfung, und damit verlasse ich das fröhliche Studentendasein mit dem lang ersehnten Grad eines Dipl. Inform.:))) Viel Spaß bei deinem Büffeln!!! Und hoffentlich hilft dir dieses Protokoll ein wenig dabei!!!