Prüfungsprotokoll

Vertiefungsgebiet Communication and Distributed Systems

Datum / Uhrzeit:	27.07.2007, 09:00 Uhr
Dauer:	ca. 45 Minuten
Prüfer:	Prof. Spaniol
Prüfling:	Thomas Brambring (thomas.brambring@rwth-aachen.de)
Note:	1.0
Fachgebiete:	Modeling and Evaluation of Communication Systems, Data Communication and Internet Technology, Distributed Systems, Security in Communication Networks

Gedächtnisprotokoll des Prüfungsverlaufs

Modeling and Evaluation of Communication Systems

Spaniol: Haben Sie etwas dagegen, wenn wir mit Modeling anfangen?

Ich: Von mir aus okay.

Spaniol: Sie kennen ja sicher das ALOHA-Protokoll. Wie modelliert man das? (Ist wohl eine recht gängige Frage bei Prof. Spaniol!)

Ich: Also ich erkläre erstmal, was das überhaupt ist. Mehrere Benutzer konkurrieren hier um einen Kanal eines geostationären Satelliten. Senden zwei dieser Benutzer zur gleichen Zeit, werden beide Nachrichten zerstört. Anschließend müssen die Nachrichten erneut gesendet werden. Die Nachrichtenlänge ist normiert auf $T = 1$. Es gibt zwei Arten von ALOHA: Pure-ALOHA und Slotted-ALOHA. Bei…

Spaniol: Was ist eigentlich besser?

Ich: Slotted-ALOHA, weil dort die Kollisionsphase nur die Länge 1 hat. Bei Pure-ALOHA hat die Kollisionsphase die Länge 2.

Spanio: Gibt es bei Slotted-ALOHA denn irgendwelche Schwierigkeiten?

Ich: Ja, man muss dafür sorgen, dass die Benutzer synchronisiert werden, da die Sendeslots getaktet sind.

Spaniol: Man möchte ja eine Beziehung zwischen dem Durchsatz S und dem Gesamtverkehr G erhalten. Wie sieht denn diese Beziehung im Allgemeinen aus?

Ich: Bei großem G geht S gegen 0 und bei kleinem G geht S gegen G.

Spaniol: Wie modelliert man das genau?

Ich: Man nimmt an, dass der Gesamtverkehr ein Poisson-Prozess mit Parameter G ist.

Spaniol: Was erhält man dann für eine Beziehung?

Ich: Bei Slotted-ALOHA: $S = e^{-G}$ (Ich wollte gerade „mal G“ sagen)…

Spaniol: Mal G. Wofür steht der Term e^{-G}?

Ich: Das gibt die Wahrscheinlichkeit an, dass die Leitung frei ist.

Spaniol: Also die Wahrscheinlichkeit dafür, dass niemand sendet. In welchem Zeitraum?

Ich: In 1 Zeiteinheit bei Slotted-ALOHA und in 2 Zeiteinheiten bei Pure-ALOHA.

Spaniol: Wir haben ja eine Stabilitätsdiskussion gehabt. Was können Sie dazu sagen?

Ich: Wir haben eine Beziehung zwischen S_{in} - das , was rein kommt - und S_{out} - das, was das System verkraftet - hergestellt. Wenn $S_{in} < S_{out}$, dann ist das System nicht ausgelastet. Bei $S_{in} > S_{out}$ ist das System überlastet.

Spaniol: Was kann man dagegen machen?

Ich: Man kann das System stabilisieren. Da gibt es die Möglichkeit…

Spaniol: Genau, man steuert zum Beispiel den Wiederholungsvorgang nach Kollisionen.

Ich: Ja, nach der i-ten Kollision wählt der Benutzer zufällig eine Zahl aus dem Intervall $[1 : K_i]$. Wobei K_i immer größer wird für jede weitere Kollision.

Spaniol: Womit wir dann bei Ethernet wären.

Ich: Bei Ethernet wählt man eine Zufallszahl aus dem Intervall $[1 : 2^i]$ für die ersten 10 Kollisionen. Für die Kollisionen 11 bis 16 wählte man eine Zahl aus $[1 : 2^{10}]$. Danach bricht man ab und nimmt an, dass ein Systemfehler vorliegt.

Spaniol: Wäre eine Stabilisierung auch noch möglich, wenn man nicht abbrechen würde?

Ich: Ähm…

Spaniol: Irgendwann muss man ja abbrechen. Also gut. Gehen wir mal zu Wartesystemen. Was ist denn das $M/M/1/K$-System?

Ich: Dies ist ein Wartesystem bei dem die Zwischenankunftszeiten exponentialverteilt sind. Die Bedienverteilung ist ebenfalls exponentialverteilt und es gibt...

Spaniol: Wofür steht das K?

Ich: Das gibt die Anzahl der Plätze im Warteraum an.

Spaniol: Also…

Ich: Ja okay, es gibt $K - 1$ Plätze im Warteraum und 1 Platz im Bediener.

Spaniol: Können Sie die Zustandswahrscheinlichkeit aufschreiben?

Ich: Ja kann ich. (Habe die Formel aufgeschrieben.)

Spaniol: Dieses System benutzt man auch zur Modellierung von Telefonsystemen. Dabei nimmt man an, dass $K = 1$ ist. Können Sie für diesen Fall die Zustandswahrscheinlichkeit angeben?

Ich: Ja. (Ich habe also die Gleichung aufgeschrieben, für K den Wert 1 eingesetzt und das Ganze noch gekürzt… 3. Binomische Formel. Das war dann auch das letzte Mal während der gesamten Prüfung, dass ich etwas schreiben musste.)

Spaniol: Wenn man nun das $M/M/1/K$-System zur Modellierung von Telefonsystemen benutzt, was meinen Sie, welches M eher falsch ist? Der erste oder das zweite?

Ich: Das zweite.

Spaniol: Ja richtig, denn die Dauer von gewöhnlichen Telefonanrufen ist nicht exponentialverteilt. Wofür benutzt man Telefonleitungen denn noch?

Ich: Zur Datenübertragung, für Faxe, …

Spaniol: Was meinen Sie denn, warum Faxe in diesem Modell besonders stören?

Ich: Weil Sie nur eine sehr kurze Übertragungsdauer haben…

Spaniol: Und…?

Ich: Weil sie eine konstante Übertragungsdauer haben.

Spaniol: Sagt Ihnen die Erlang-n-Verteilung etwas?

Ich: Ja, die kenne ich. (Aber soviel wusste ich dazu jetzt nicht zu sagen und habe deswegen schnell etwas anderes eingeworfen…) Da gibt es aber auch die Erlang-C-Formel. Die benutzt man bei $M/M/S$-Systemen, um die Wahrscheinlichkeit zu berechnen, dass ein neuer Kunde warten muss. Das ist vor allem beim Entwurf von Telefonsystemen interessant. Soll ich die Wahrscheinlichkeit aufschreiben?

Spaniol: Nein, ich glaube Ihnen schon, dass Sie die wissen. Nun betrachtet man Wartesysteme ja nicht nur einzeln, sondern auch in Wartenetzen. Was wäre denn so die einfachste Form eines Wartenetzes?

Ich: Das einfachste Wartenetz? Es gibt zum Beispiel das Jackson-Netz. Das ist einfach.

Spaniol: Genau, das wäre die einfachste Form eines Wartenetzes. Nun kann die Berechnung der Zustandswahrscheinlichkeit dort ja recht komplex sein. Wie macht man das denn?

Ich: Das Jackson-Netz hat ja Produktform…

Spaniol: Produktform, genau das wollte ich hören. (!)

Ich: Ja und da kann man dann einfach das Produkt der Zustandswahrscheinlichkeiten der einzelnen Knoten bilden. So erhält man die Zustandswahrscheinlichkeit für das gesamte Jackson-Netz.

Spaniol: Und wie erhält man letztlich die Zustandswahrscheinlichkeit?

Ich: Wie gesagt… über die Produktform.

Spaniol: (Er hat nochmals nachgefragt, aber ich habe nicht verstanden, was er meint. Die Antwort, die er hören wollte war schließlich „Man erhält die Lösung durch Lösen eines Gleichungssystems“. Hmmm...)

Ich: Aha, okay.

Spaniol: Wenn ein Kunde einen Knoten wechselt wird die Zeit für diesen Wechsel im Allgemeinen nicht berücksichtigt. Wieso macht man das nicht?

Ich: Naja, es gäbe die Möglichkeit die Übergangszeit zur Bedienzeit des letzten Knoten hinzu zu addieren. Das würde aber das Übergangsverhalten stören. Außerdem kann man Zwischenknoten einführen, was allerdings zu einer Zustandsexplosion führt.

Spaniol: Wenn wir uns jetzt einen Supermarkt vorstellen. Was wird bei der Modellierung eines solchen durch ein Jackson-Netz nicht berücksichtigt?

Ich: Ähm… dass die Kunden nicht direkt zu einem anderen Knoten wechseln können.

Spaniol: Ja, die Kunden würden bei dieser Modellierung unendlich schnell die Theken wechseln. Wenn wir nun aber Zwischenknoten einführen würden, welche die Übergangszeiten modellieren würden, welche Verteilung würde für diese Knoten gelten.

Ich: Die Gleichverteilung.

Spaniol: Na, die Übergangszeiten wären konstant.

Ich: Das war, was ich eigentlich meinte.

Data Communication and Internet Technology

Spaniol: Kommen wir mal zu Data Communication. Im Bereich LAN gibt es ja zum Beispiel Ethernet. Was kennen Sie da noch?

Ich: FDDI

Spaniol: Was ist das?

Ich: Das ist ein Token Ring.

Spaniol: EIN Token Ring?

Ich: Genauer gesagt sind es zwei Ringe. Ein Primary-Ring und ein…

Spaniol: Gibt es einen Token oder mehrere?

Ich: (Habe ich nicht sofort gewusst und zu schnell geraten „einen“.)

Spaniol: Nein, es gibt mehrere. (Falsch geraten)

Ich: (Da fiel es mir wieder ein.) Ach ja richtig, es gibt…

Spaniol: Wie wird bei FDDI denn auf Bit-Ebene codiert?

Ich: (Puh, wie war das noch? Das erste was mir einfiel war „Manchester Code“ und das hab ich dann auch gesagt.)

Spaniol: 4B/5B (Wieder falsch geraten grrr…)

Ich: Genau, man kodiert 5 Byte in 4 Byte… ähm… ich meine 4 Byte in 5 Byte.

Spaniol: Byte? Bit! (Irgendwie war ich gerade raus.) Man kann bei 4B/5B garantieren, dass nur eine gewisse maximale Anzahl von Einsen bzw. Nullen hintereinander folgt. Für wie viele aufeinander folgende identische Bits ist das möglich?

Ich: Alsooo… (Im Moment hatte ich das Gefühl, dass nichts mehr geht.)

Spaniol: Für 1 sicher nicht. Für 2 auch nicht. Für 3 kann man es garantieren.

Ich: Öh, ja.

Spaniol: Wieso möchte man verhindern, dass viele gleichartiger Bits aufeinander folgen?

Ich: Man möchte Gleichspannung verhindern. Außerdem muss man sehen können, wann bei einer Folge von beispielsweise Nullen die nächste Null anfängt.

Distributed Systems

Spaniol: Gehen wir zu den Verteilten Systemen. Wir hatten ja Zeitsynchronisation. Da gibt es verschiedene Algorithmen für. Ich frage ganz gerne danach. (!)

Ich: Es gibt Cristian’s Algorithm, Berkeley Algorithm, …

Spaniol: Suchen Sie sich einen aus und erklären mir, wie er funktioniert.

Ich: Dann nehme ich mal Cristian’s Algorithm. Es gibt dort einen Server, der die UTC kennt. Ein Client möchte seine eigene Uhrzeit mit der UTC dieses Servers synchronisieren. Dazu sendet er eine Anfrage an den Server und merkt sich diesen Zeitpunkt t_{send}. Der Server antwortet mit der UTC t_{UTC}. Sobald der Client die Antwort erhält, merkt er sich wieder diesen Zeitpunkt $t_{receive}$. Nun setzt der Client seine Uhrzeit $t_{sync} = t_{UTC} + \frac{t_{receive} - t_{send} - t_{response}}{2}$.

Spaniol: Was nimmt man hier an, wenn man durch 2 dividiert?

Ich: Man nimmt an, dass die Sendezeiten vom Client zum Server und vom Server zum Client etwa gleichlang sind.

Security in Communication Networks

Spaniol: Wir müssen noch zu Sicherheit in Kommunikationsnetzen kommen. Wenn Sie eine Freundin in Thailand haben und möchten mit ihr verschlüsselt kommunizieren. Wie tauschen Sie dann einen sichern Schlüssel aus.

Ich: Mit Diffie-Hellmann.

Spaniol: Können Sie kurz erklären, wie das funktioniert? Nur ganz kurz.

Ich: Man braucht zunächst Zwei Zahlen. Eine Primzahl p und eine Zahl $g < p$. Jetzt denke ich mir eine weitere Zahl S_A aus und berechne $T_A = g^{S_A} \bmod p$. Das sende ich an meine Freundin.

Spaniol: Und Ihre Freundin berechnet…

Ich: Sie berechnet $T_B = g^{S_B} \bmod p$ und sendet dies an mich. Ich berechne anschließend $K_{AB} = {T_B}^{S_A} \bmod p$ und sie $K_{AB} = {T_A}^{S_B} \bmod p$, womit wir beide einen gemeinsamen Schlüssel hätten.

Spaniol: Nun kann man diesen Schlüssel nicht einfach als DES Schlüssel nehmen. Was macht man da?

Ich: Da ein DES schlüssel effektiv nur aus 56 Bit besteht, vereinbart man, vom Schlüssel K_{AB} nur die ersten 56 Bit zu nehmen.

Spaniol: Ja, man vereinbart 56 Bit des Schlüssels K_{AB} zu verwenden. Diese kann man beliebig aus K_{AB} wählen, man muss dies nur festlegen. Nun kann aber ein Angreifer, der die Kommunikation abhört einen Schlüssel mit beiden Seiten austauschen…

Ich: Richtig, das ist die Bucket-Brigade Attacke. Ein Angreifer kann so unbemerkt die Kommunikation abhören und manipulieren.

Spaniol: Wie kann man das verhindern?

Ich: Eine Seite kann die Nachrichten vor dem Senden signieren.

Spaniol: Womit macht man das?

Ich: Mit Public-Key.

Spaniol: Mit einem Public-Key geht das bestimmt nicht.

Ich: Jaaa, ich meine natürlich mit einem Public-Key-Verfahren. Die Signierung erfolgt dann mit dem eigenen Private-Key.

Spaniol: Sie kennen ja sicher auch Lamports Hash. Können Sie das erklären?

Ich: Es gibt einen Client und die Workstation des Clients. Außerdem gibt es einen Server, bei dem sich der Client authentifizieren möchte. Zunächst schickt der Client ein Passwort an seine Workstation. Diese wählt eine Zahl n und berechnet $hash^n(Passwort)$. Diesen Hash und n sendet sie an den Server. Möchte sich nun der Client gegenüber dem Server authentifizieren, dann sendet er wieder sein Passwort an die Workstation. Dieses schickt eine Anfrage an den Server, worauf dieser mit n antwortet. Die Workstation berechnet nun $hash^{n-1}(Passwort)$ und sendet das Ergebnis an den Server. Der berechnet davon noch einmal den Hash und vergleicht das Ergebnis mit seinem gespeicherten Wert. Er speichert jetzt den neuen Hash und $n - 1$.

Spaniol: Gibt es hier Möglichkeiten für einen Angriff?

Ich: Ja, die sogenannte Small- n-Attack. Die funktioniert dann, wenn der Angreifer es schafft, sich als Server auszugeben. Bei der nächsten Anmeldung des Clients bzw. seiner Workstation sendet der Angreifer dieser einen kleinen Wert n. So erhält der Angreifer $hash^{n-1}(Passwort)$. Diesen Wert kann er dafür benutzen, um sich beim richtigen Server zu authentifizieren. Wenn der wirkliche Server ein kleineres n gespeichert hat, dann hat der Angriff Erfolg.

Spaniol: Wie oft kann sich der Angreifer dann authentifizieren?

Ich: So häufig, wie die Differenz aus den beiden n's.

Spaniol: Kann man diesen Angriff verhindern?

Ich: Jaaa… (Ich habe kurz nachgedacht.)

Spaniol: Über Papier kann man einen Schlüssel austauschen… So das war's dann für mich. Wenn Sie bitte kurz rausgehen würden.

Ich: (Ich gehe also raus)

Spaniol: (Holt mich nach einer Minute wieder rein) Das war ja alles sehr erfreulich. Ich gebe Ihnen eine 1.0.

Anmerkungen

Ich selbst hätte nach der Prüfung mit einer 1.3 gerechnet, weil ich einen kleinen Aussetzer bei FDDI hatte. Prof. Spaniol fand darin aber scheinbar keinen Grund, darauf zu verzichten, mich ordentlich für meine Leistung zu belohnen :o) Mir war das natürlich mehr als recht. Insgesamt war die Prüfung sehr locker und angenehm.

Ich habe vor meiner Prüfung immer wieder Gerüchte gehört, dass Prof. Spaniol oft launisch sein soll und dann auch unfaire Noten vergibt. Ich weiß nicht, wer da einen schlechten Tag hatte, Prof. Spaniol, oder die Person, die diese Gerüchte verbreitet hat, aber ich bewerte diese Aussagen als nicht haltbar. Ich habe Prof. Spaniol jedenfalls als sehr freundlich und umgänglich empfunden.

Generell ging die Prüfung eher in die Breite als in die Tiefe. Ich musste nur einmal eine Formel aufschreiben und dort einen Wert einsetzen. Alles andere lief mündlich ab. Längere Beweise oder sonstige Rechnungen braucht man hier nicht zu können. Für Modeling empfehle ich insbesondere, alle Zustandswahrscheinlichkeiten zu lernen und ALOHA erklären und modellieren zu können. Bei den Verteilten Systemen fragt Prof. Spaniol gerne Algorithmen zur Zeitsynchronisation ab. Selbstverständlich sollte man generell ein gutes allgemeines Verständnis von der Materie haben.

Prof. Spaniol redet relativ viel und lässt einem nur wenig Zeit nachzudenken. Überlegt man zu lange, gibt er schnell selber die Antwort. Teilweise nimmt er einem auch das Wort ab. Ich fand diese Art gar nicht mal so unangenehm, da es so nicht sonderlich auffällt, wenn man mal etwas nicht weiß.

Ich kann Prof. Spaniol als Prüfer und sein Vertiefungsgebiet nur empfehlen.

Viel Erfolg bei der Prüfung!