

Diplomprüfungsprotokoll Rechnernetze I - Softwaretechnik I

Steffen Ruthotto

21. April 2008

1 Rechnernetze

Als Eröffnungsfrage wollte Herr Steigner wissen, was ein Kommunikationsprotokoll auszeichnet. Habe dazu u.a. was von horizontaler Kommunikation erzählt, was wohl das war, was er hören wollte. Er fragte noch, warum eigentlich horizontale Kommunikation. Nebenbei hatte ich etwas von Schichten erzählt (im Zusammenhang mit vertikalem Datenfluss), also kam er als nächstes zum TCP/IP-Modell, das ich aufmalen sollte. Daraus entstand eine ca. fünfminütige Diskussion zwischen Herrn Steigner und Herrn Ebert, weil Herr Ebert wissen wollte, ob das alte OSI-Modell überholt ist. Als nächstes wollte Herr Steigner wissen, wie man an eine MAC-Adresse kommt, wenn man nur die IP-Adresse kennt. Habe kurz ARP erklärt. Wichtig war hier der Zusammenhang, dass man die bekannte Broadcast-Adresse verwendet, die jeder Teilnehmer mithört. Er fragte nach den Problemen, die es mit Broadcast geben kann, worauf ich den Broadcast-Sturm und Schleifen und STP erwähnte, was Herr Steigner aber alles nicht genauer wissen wollte. Er fragte, wie man schleifenfrei, aber ausfallsicher (also mit redundanten Verbindungen) Switches verbinden kann. Ich hatte keine Ahnung, worauf er hinaus wollte, aber die Antwort war, dass man zwei Switches über mehrere Ports miteinander verbinden kann und so die Bandbreite erhöht. Da ich zwischendurch was von VLAN erwähnt hatte, fragte er dann, wie ein Switch lernt (im allgemeinen Fall) und was bei einem VLAN der Administrator bereitstellen muss, nämlich die VLAN-Zugehörigkeit. Er sagte noch, dass der Name der VLAN-Zugehörigkeit der Tag ist, was ich etwas komisch fand. Welche Tabellen ein Switch braucht für VLAN wollte er dann noch wissen, das sind einmal die normale Tabelle, die er eh hat und dazu die VLAN-Tabelle (genaue Bezeichnung weiß ich nicht). Hier steht drin, für welches VLAN welcher Port benutzbar ist (da pro VLAN ein Spanning Tree existiert, kann ein Link ja für ein VLAN aktiv und für ein anderes geblockt sein) und ob die Pakete darüber getaggt oder ungetaggt versendet werden. Dann kamen wir noch zum Unterschied zwischen Port-based und MAC-based Forwarding und als letztes zur Flusskontrolle. Hier sollte ich frei erzählen, was mir einfällt. Habe dann den Unterschied erklärt zwischen Kontrolle auf Data Link Layer (Sliding Window) und bei TCP (Advertised Window). Bei Data Link fragte Herr Steigner noch nach der Bestimmung der Fenstergröße. Habe irgendwas von RTT gemurmelt, was er wohl hören wollte, damit war Rechnernetze auch schon zu Ende.

2 Softwaretechnik

Herr Ebert fing gleich mit UML an, die üblichen Fragen zum Softwarelebenslauf ließ er aus zeitlichen Gründen weg. Er wollte wissen, welche Diagrammarten der UML wir denn in der Vorlesung kennengelernt haben. Hab ich aufgezählt und dann wollte er wissen, ob ich noch mehr kenne, die nicht in der Vorlesung dran waren. Hab irgendwas geraten, was es aber leider gar nicht gab.

Dann kamen wir zu State Charts, wie die denn aussehen etc. und ich sollte mal beispielhaft einen aufzeichnen. Hab dann an eine Kante Guard, Action, Trigger hingeschrieben, weil sie mir in der Reihenfolge einfielen. Auf Nachfrage habe ich die in die richtige Reihenfolge gebracht und erklärt, dass sie optional sind. Er wies noch auf den Schrägstrich bei Action hin. Als nächstes kam das Metamodell dazu. Es war leider das, das ich selber am meisten gefürchtet hatte. Das Problem waren XOR- und AND-Blobs und deren Beziehungen, aber ich kann leider nicht wirklich beschreiben, was das Problem war. Wenn ich mich richtig erinnere, ist es wichtig, dass And-Blobs aus XOR-Blobs bestehen und XOR-Blobs aus Zuständen.

Als letztes kam dann das Testen dran. Herr Ebert fragte, was ein Testfall ist, wie man an Testfälle kommt und so weiter. Habe dann Blackbox- und Whiteboxtesten weiter erklärt. Wie kommt man an gute Testfälle (Randfälle betrachten), was für Überdeckungen gibt es (Bei BB: Eingabe-, Ausgabe- und Funktionüberdeckung, bei WB: Anweisungs- und Zweigüberdeckung). Er wollte wissen, ob ich schon mal mit Tools getestet habe. Habe dann JUnit aus Programmierung erwähnt. Herr Ebert fragte noch nach Mutationstesten, wies aber auch darauf hin, dass er das nicht immer behandelt hat und deswegen wurde die Frage aus dem Protokoll gestrichen.

3 Allgemeines

Alles in Allem war die Prüfung sehr angenehm. Beide Prüfer waren sehr nett und ruhig und es war immer klar, um was es geht also keine schwammigen Fragestellungen oder unklare Formulierungen. Ich konnte mir aussuchen, womit ich anfangen will und die Benotung war auch angemessen. Die Kombination Steigner / Ebert kann ich also weiterempfehlen.

Ich habe insgesamt ca. 3 Wochen gelernt, was schon relativ knapp ist bei der Stoffmenge. Zur Vorbereitung kann man bei ST gut das Skript von Andreas Harder verwenden und für die UML-Diagramme auch mal genauer in einem Buch nachschlagen. Für Z hatte ich mir noch ein paar Ebooks besorgt. Der Fragenkatalog ist auch sehr hilfreich. Man sollte die Metamodelle alle mal gemacht und auch mit anderen drüber diskutiert haben.

Für RN kann man sehr gut aus Herrn Steigners Folien lernen. In Büchern kriegt ihr den Stoff auch nicht viel ausführlicher und vor allem nicht so gut an Beispielen durchgerechnet. Die englische Wikipedia gibt auch sehr viel her. Man kann sich da gut einen ersten Überblick verschaffen, um was es geht und dann in den Folien noch mal genauer nachschauen.

Weil es nicht immer leicht ist in den Folien ein Thema wiederzufinden, habe ich mir eine Themenliste zu den Folien erstellt, vielleicht hilft sie ja jemandem.

4. Switch, Router, Adressen
5. Allgemeines, Sockets
6. Sockets
7. Sockets
8. Sockets, HTTP
9. HTTP
10. Nyquist, Fourier
11. Shannon, Kabel
12. Fiber-Optics, Wireless
13. Multiplexen, Modulation, ISDN, DSL
14. DSL, SONET
15. SONET, ATM, Kodierung
16. Krypto, Parität, Hamming
17. CRC, JPEG
18. Protokolldesign, x-Kernel, OSI
19. Stop And Wait, Sliding Window, Ethernet, STP, VLAN
20. VLAN
21. VLAN-Tagging, RIP
22. RIP, OSPF, BGP
23. IPv4, CIDR, ARP
24. ICMP, UDP
25. TCP
26. Staukontrolle

Viel Erfolg bei euren Prüfungen!