Gedächnisprotokoll zu VVS1, VVS2, DatKom (jeweils als V4)
Prüfer: Prof.Freiling
Prüfling: Markus de Brün
Note: 1,0

Das Protokoll ist vermutlich nicht vollständig, aber das wichtigste sollte drin sein.

Freiling: Womit wollen sie denn anfangen?

FLP

FLP sagt ja folgendes aus ...blabla... Der Beweis dazu ist ja sehr schwer, aber den haben wir ja gar nicht gemacht, sondern?

Völzer!
Der betrachtet nicht Consensus sondern Pseudo-Consensus. Das ist aber nicht schlimm, weil jeder Consens-Algorithmus auch Pseudo-Consensus löst und wir deswegen einen Rückschluss ziehen können.

Wo ist denn der Unterschied?

Die Terminierungs-Eigenschaft. Consensus fordert dass der Algorithmus immer terminiert, Pseudo-Consensus nur, dass Terminierung immer möglich ist.

Was wollen wir denn zeigen?

1.) Es gibt eine nicht uniforme Start-Konfiguration
2.) Wir kommen von jeder nicht uniformen Konfiguration wieder in eine nicht uniforme Konfiguration

zu 1:
c_0 ist 0-uniform, c_n ist 1-uniform (wegen der Validity-Eigenschaft von Consensus) Dazwischen gibt es eine Konfiguration die nicht uniform ist. Betrachte letzte 0-uniforme c_{k-1} und zeige dass c_k nicht uniform ist. Prozess p_k schlägt in c_{k-1} 0 vor und in c_k 1. Wenn wir p_k stumm schalten können die anderen Prozesse die Konfigurationen nicht unterscheiden. Also muss in c_k auch 0 entscheidbar sein. Da c_k aber nicht 0-uniform ist, muss c_k nicht uniform sein.

zu 2:

Ausführlich erklärt! In der Vorlesung ist der Beweis nicht so gut rüber gekommen. Ich kann aber das paper von Völzer sehr empfehlen! Wichtig ist zu erwähnen, wo man die Consensus Eigenschaften braucht:
Pseudo-Terminierung heisst $val(p, c) \neq \emptyset$
Integrity und Agreement: Wenn einmal v entschieden wurde, dann bleibt das auch so.

Was bedeutet das denn jetzt?

Wir brauchen einen Fehlerdetektor (FD) für Consensus.

Welchen denn?

Bei einer Mehrheit korrekter Prozesse Ω.

Und was wäre, wenn wir annehmen, dass alle Prozesse zu Beginn ausfallen?

Dann lösen wir mit den verbleibenden einfach Consensus - ohne FD

Und wie ist das bei Terminating reliable broadcast (TRB)?

Da brauchen wir P.

Wie zeigt man das?

hinreichend: Löse TRB mit P
notwendig: Emuliere P durch TRB

Wie ist denn diese Emulation.

Alle Prozesse senden fortlaufend Nachichten. Bekome ich von jemandem ein *nil*, dann verdächtige ich ihn.

Dann kam noch ein bisschen zu Synchronen Systemen und Message Delays im Internet und dann waren wir auch schon bei der Praxis...

Was für Angriffsmöglichkeiten haben wir kennengelernt?

Bufferoverflow, Race conditions, Passwörter (durch Wörterbuch bzw hybride Angriffe)

Wie funktionieren denn race conditions?

Zwei Prozesse greifen auf gemeinsame Ressource zu und verändern sie getrennt. (Beispiel: zwei Prozesse, eine Variable)

Und wie kann man das ausnutzen?

Es gab mal diesen Ptrace-Bug. Aufrufe von ptrace werden nicht geblockt. Man kann ein Modul mit ptrace manipulieren bevor der Kernel seine EUID/GUID auf 0 setzt.

Wozu braucht man denn TCP bzw IP? Wo unterscheiden sich denn die header?

IP zur Adressierung von Rechnern, TCP für Prozesse (über Port-Nummer). Hier habe ich ihm noch die anderen Felder des IP-Headers genannt.

Wie funktioniert denn die Adressierung, wenn wir für ein Netzwerk nur eine IP haben?

Äh...

Ist klar worauf ich hinaus will?

Vielleicht auf Network Address Translation (NAT)!?

Ja!

Okay. Wenn wir eine Seite anfragen ersetzt die NAT-Box die richtige IP durch die Internet-IP und den richtigen Port durch einen Verweis auf eine Tabelle. In der Tabelle stehen dann die richtige IP und der richtige Port. Kommt eine Anfrage an die Internet-IP so schlagen wir die "Portnummer" in der Tabelle nach und setzen die richtige IP und den richtigen Port wieder ein.

Es gibt da dieses ICMP Protokoll...

...für Kontrollnachichten zwischen routern und für ping.

Und wie macht man damit ein tracerout?

Sende ein Paket mit TTL=1 - das kommt dann vermutlich zurück, weil die

TTL abgelaufen ist.
Sende ein Paket mit TTL=2 und so weiter. So bekommen wir schliesslich den ganzen Pfad.

Und wenn firewalls ICMP Pakete filtern? Was ist denn wenn wir das in ein TCP Paket verpacken?

Dann kommt das durch die firewall aber ggf nicht wieder zurück. Ich glaube nicht, dass das funktioniert.

Da gibt es angeblich ein tool für, * oder so.**

Kenne ich nicht.

Schade. Ich weiss auch nicht wie das funktioniert...
Letztes Thema: Was ist BGP? Ich meine nicht das Byzantine Generals Problem.

Ein exterior routing Protokoll. Hier kann man Konditionen angeben, wie zum Beispiel das Pakete aus den USA nicht durch den Irak laufen oder ähnliches.

Und was benutzt man intern?

Früher RIP oder RIPv2, heute eher OSPF.

Okay, dann gehen sie soch mal kurz raus.

Die Prüfung war sehr angenehm und ich hatte den Eindruck dass Prof.Freiling auch bereit war sich von einem Thema zum anderen lenken zu lassen. Beispiele waren meist ausreichend, Detailwissen war eher selten gefragt.