

gedächtnissprotokoll

datum: 19.02.2009

modul<sic!>: dkr

prüfer: be wolfinger

beisitzer: heidtmann

note: 3,3

wolfinger hatte wieder seinen umschlag mit den fragen dabei. er sieht aber garnicht so dick aus. entweder hat er noch irgendwo nachschub, oder er mischt in der mittagspause einmal durch. wie auch immer. die pakete sind aber schon strukturiert. auf dem tisch lagen schon einige leere zettel und ein bleistift.

hint:

überleg dir für jedes thema ein paar sätze auf die "frage was ist ~?". gerne auch mit details die er sonst ohnehin fragen wird.

ich hab erstmal nur die fragen aufgeschrieben, weiter hinten dann antworten die mir so einfielen oder ich nach der prüfung aus dem skript rausgesucht hab. manche fragen enthalten teile der antworten der vorhergehenden frage.

1. grundlagen der datenübertragung

wie ist der zusammenhang zwischen der frequenz und der wellenlänge?

2. crc

was ist crc?

kann man sich sicher sein, dass die daten korrekt übertragen wurden?

wie muss das generatorpolynom aussehen damit alle 1-bit-fehler auf jeden fall garantiert werden können?

was ist das fehlerpolynom?

wie ist der zusammenhang zwischen fehlerpolynom und generatorpolynom?

wie lange bursts werden bei welcher polynomlänge erkannt?

3. hypercube

was ist ein hypercube?

wo werden hypercubes verwendet?

wie sieht ein hypercube der dimension 1 aus?

wieviele knoten hat ein hypercube?

wieviele verbindungen hat ein hypercube pro knoten?

was sind die vorteile von hypercubes?

4. echtzeitkommunikation im internet

ist das internet generell für echtzeitkommunikation geeignet?

was kann man gegen die problematik des best effort tun?

funktioniert die priorisierung immer?

wie kann man verhindern, dass zuviele aufträge hoher priorität entstehen?

5a. last

was ist last?

nennen sie ein beispiel für so ein system.

was genau wird an der schnittstelle gemessen?

5b. last-modellierung

wie kann man last modellieren?

1. Grundlagen der Datenübertragung

Wie ist der Zusammenhang zwischen der Frequenz und der Wellenlänge?

Er wollte schlicht $\lambda = c/f$ hören, klar. Hätte bestimmt noch mehr gefragt, wenn ich nicht vor dem Berg gestanden hätte wie der sprichwörtliche Ochse.

Grundlegende Bruchrechnung sollte man auch beherrschen, wenn nicht kann es aber trotzdem für eine Dreiminus reichen (insgesamt, hier 5,0)

2. CRC

Was ist CRC?

Cyclic Redundancy Check ist ein Verfahren zur Prüfsummenbildung.

An die Daten werden Nullen angehängt, und zwar eins weniger als das Generatorpolynom lang ist. Das so verlängerte Polynom wird durch das Generatorpolynom geteilt. Der Rest wird vorne mit Nullen aufgefüllt, bis die Zahl so viele Stellen hat wie die vorher hinzugefügten Nullen.

Der Empfänger teilt die empfangenen Daten wieder durch das Generatorpolynom, wenn

kein Rest rauskommt, wurden die Daten wahrscheinlich nicht verfälscht.

Kann man sich sicher sein, dass die Daten korrekt übertragen wurden?

Nein, sie könnten so verfälscht worden sein, dass bei der Polynomdivision wieder Null rauskommt. (Fehlerpolynom!)

Wie muss das Generatorpolynom aussehen, damit alle 1-bit-Fehler auf jeden Fall garantiert werden können?

Es muss mindestens zwei Terme enthalten.

Was ist das Fehlerpolynom?

Für jedes umgekippte Bit im gesendeten Polynom ist es 1. (auch innerhalb der Prüfsumme)

Wie ist der Zusammenhang zwischen Fehlerpolynom und Generatorpolynom?

Ist das Fehlerpolynom ein Vielfaches des Generatorpolynoms, wird der Fehler nicht erkannt.

Wie lange Bursts werden bei welcher Polynomlänge erkannt?

$x^{32} \dots 1 \Rightarrow$ alle 32-bit-Bursts werden erkannt.

3. Hypercube

Was ist ein Hypercube?

Ein Hypercube ist eine Topologieform. Tja, und wie fasst man die Topologie in einfache Worte? Vielleicht einfach aufzeichnen.

Wo werden Hypercubes verwendet?

Nicht im Backbone-Bereich, sondern rechnerintern.

Wie sieht ein Hypercube der Dimension 1 aus?

Es sind zwei Knoten mit einer Linie. (Nicht etwa ein Knoten ohne Linie, das wäre Dimension 0)

Wieviele Knoten hat ein Hypercube?

2^n

Wieviele Verbindungen hat ein Hypercube pro Knoten?

n

Was sind die Vorteile von Hypercubes?

Skript 08/09 S. 65:

- relativ einfache Wegeermittlung
- relativ gute Zuverlässigkeit durch Alternativpfade
- geringe Anzahl von Hops
- relativ geringe Anzahl benötigter Verbindungen

4. Echtzeitkommunikation im Internet

Ist das Internet generell für Echtzeitkommunikation geeignet?

schwierig, weil man benötigt für echtzeitkommunikation:

skript 08/09 s. 126:

- limitierte paketverlustrate
- limitierte verzögerungsschwankung
- garantierter mindestdurchsatz

internet, also ip, bietet aber nur best effort.

was kann man gegen die problematik des best effort tun?

(weißt du nicht? nimm diese hinweise: dumb fat pipe, diffserv, selected-irgendwas)

skript 08/09 s. 126:

- überdimensionierung ("dumb fat pipe")
- priorisierungen (diffserv)
- reservierungen (isa, rsvp)

funktioniert die priorisierung immer?

nein, da zu viele priorisierte aufträge da sein können.

wie kann man verhindern, dass zuviele aufträge hoher priorität entstehen?

beschränkungen für bestimmte nutzer oder systeme einführen.

5a. last

was ist last?

skript 08/09 s. 166:

$L=L(U,S,IF,T)$

(nein, last ist kein tupel, es ist eine das ergebniss einer funktion mit vier parametern, steht extra in den folien und wird in der vorlesung erwähnt)

last ist die sequenz von aufträgen, die durch eine umgebung u an ein bediensystem s über eine wohldefinierte schnittstelle if während eines beobachtungsintervalls t übergeben werden.

nennen sie ein beispiel für so ein system.

tcp. aus der umgebung kommen keine tcp-pakete, vielmehr kommen daten, die vom bediensystem in tcp-paketen verpackt werden sollen.

vielleicht nicht so das beste beispiel.

was genau wird an der schnittstelle gemessen?

aufträge. (nicht etwa daten, pakete, bytes...)

5b. last-modellierung

wie kann man last modellieren?

mit dem benuterverhaltensautomat (bva)

es gibt d und r knoten.

d wie delay, erzeugt eine wartezeit abhängig von einer wahrscheinlichkeitsverteilung.

r wie request, erzeugt einen auftrag, hat einen last-typ und attribute (nicht parameter oder so).

wichtig ist auch, dass die last-typen nicht in den r-knoten definiert sind, sondern dort nur die `<s>parameter</s>` attribute festgelegt werden.

übergänge zwischen den knoten mit wahrscheinlichkeiten.