

Transportlag

I min implementasjon av MIPTP og dens transportprotokoll for pålitelig dataoverføring har jeg brukt go-back-n for å sikre at pakker som forsvinner blir sendt på nytt. Jeg har satt maks grense at miptp kan ha 10 klient tilkoblinger. det vil si, 10 klientmotakere, 10 klientersendere eller en blanding. bruker en multi-dimensjonal array for å holde styr på hvilke sockets som tilhører hvilken port. Antar at 2 tilkoblinger ikke vil ha samme port.

Programmet starter ved at en transferclient åpner en fil og leser den inn. jeg bruker stat(2) funksjonen til å holde styr på lengden, legger på en header fra klienten med source, destination, port og størrelse før jeg sender den til miptp. miptp leser inn pakken og deler opp om nødvendig. pakkene blir lagt inn i en struct. Sender de 10 første pakkene (window size), og venter på ack. når ack motas sendes den neste ikke ack'ede pakken. De 10 neste pakkene fra sist ack'ede sendes ved hver timeout helt til alle pakker er sendt og acket. Ackede pakker sendes direkte til transferserver.c hver gang en ny pakke ackes og skrives til fil.

Ettersom jeg har en ekstra header under miptp sender jeg noe mindre pakker enn 1492. den største pakkestørrelsen jeg fikk til å bruke uten feil er 1460.

Port 30: hvis to filer sendes gjennom port 30 er det vanskelig å holde kontroll på hvilken fil som er hvilken. Problemet kan løses ved å legge på headere som holder styr på pakkene.
port: port 80 webserveren lager en egen unik id slik at den enkelt kan holde styr på de forskjellige forbindelsene sine.

Regne ut timeout kan gjøres ved å først sette en verdi, som man senere endrer ut i fra hvor ofte ack'ede pakker kommer inn.

Innleveringen er noe uryddig da jeg fikk veldig dårlig tid mot slutten, og filer fra hel som er brukt i headerfil er uten at det er i bruk i programmet er ikke tatt bort. Håper ikke dette er noe proble,

Filer som er i bruk:

makefile: Forskjellige muligheter for kompilering av programmet
functions.h: headerfil med alle metoder og litt beskrivelser av metoder.

transferclient.c: leser en fil og sender over unix domain socket

transferserver.c: motar pakker fra miptp og skriver til fil

miptp.c: ansvarlig for go-back-n

local.c: funksjoner for oppdeling av fil til pakker.

daemon.c: select server som motar melding fra unix-socket/raw-socket

send&recv.c metoder for å lese og skrive til/fra socket

cashe.c: miparpcache struct og metoder for å hente ut/inn ny cach

raw.c: det som omhandler interface, mac adresser, arp.