

INF3190 - Hjemmeeksamen 1 - Vår 2018

Ruting

Formelt

Denne oppgaven er karaktergivende og skal løses individuelt. Karakteren som gis teller omlag 20% på sluttkarakteren. Oppgaven blir vurdert etter i hvor stor grad kravene i avsnittet "Oppgave" er oppfylt.

Utleverte filer

Programmet testes med mininet topologien som vist [her](#), og python-scriptet som kan lastes ned [her](#).

Det må være mulig å kjøre pingmålinger mellom **MIP-adresse 10** og **MIP-adresse 100**.

Oppgave

Målet med denne oppgaven er å implementere en rutingserver som tilbyr rutingfunksjonalitet til nettverkslaget. Med disse funksjonene kan nettverkslaget utføre ende-til-ende overføring av pakker mellom alle endesystemer som er knyttet til nettverket. Selv om det ikke finnes noen direkte Ethernet-forbindelse mellom to endesystemer skal disse kunne kommunisere med hverandre med hjelp fra mellomsystemene (intermediate systems). Systemer som bare tar imot pakker og sender dem videre på vei til mottakeren skal virke som rutere. Alle noder må være i stand til å jobbe både som endesystem og som ruter **samtidig**. Rutingmekanismen som skal implementeres er Distance Vector Routing (DVR) with Split Horizon (distansevektorruting med delt horisont).

Kjernen i oppgaven er implementasjonen av DVR med Split Horizon. Du må implementere en rutingprokoll på nettverkslaget som oppdaterer rutingtabeller slik som det er definert i DVR. Du skal anta at alle direkte sammenkoblede nabonoder har avstand 1.

Spesifikke krav til implementasjonen

Du må implementere en rutingserver. Hver host kjører en slik rutingserver, og tilbyr rutingfunksjonalitet ved å kommunisere med MIP-daemonet på samme host. Grensesnittet til MIP-daemonet mot applikasjonen må ikke forandres fra den obligatoriske oppgaven. Hver node skal kun støtte én applikasjon til enhver tid, fordi MIP-daemonet implementerer nettverkslaget. Nettverkslaget utfører ikke multipleksing og demultipleksing for flere applikasjoner, dette er oppgaven til transportlaget. Til tross for denne begrensningen må MIP-daemonet skille mellom

pakker som rutingsserveren bruker for å kommunisere med andre rutingsservere og pakker til og fra applikasjonen.

MIP-daemonet må forandres litt. Det må kunne få adressen til riktig neste hopp (nabo) fra rutingsserveren slik at det blir mulig å kommunisere mellom endesystemer som ikke er koblet direkte til det samme Ethernet-nettverket. Dette betyr at pingserveren og pingklienten fra den obligatoriske oppgaven burde fungere uten forandring, og det burde være mulig for en klient å "pinge" en server som befinner seg mer enn ett nettverkslagshopp unna.

Rutingsserveren må svare på forespørsler om ruter fra MIP-daemonet for å gi MIP-daemonet muligheten til å utføre videresending av pakker. Den mottar en MIP-destinasjonsadresse fra MIP-daemonet og returnerer MIP-adressen til den riktige nabonoden. På den andre siden må rutingsserveren bruke MIP-daemonet for å kommunisere med rutingsserverne på andre noder, noe som trengs for å oppdatere rutingtabellene. For å skille mellom kommunikasjon angående ruteoppslag for videresending (forwarding) og utveksling av rutingsinformasjon med andre rutingsservere må det brukes to forskjellige sockets mellom rutingsserveren og MIP-daemonet.

Rutingsserveren må oppdatere rutingtabellene regelmessig (periodisk). Rutingsalgoritmen som brukes er Distance Vector Routing with Split Horizon (distansevektorruting med delt horisont). MIP-daemonet skiller routingpakker (meldinger som brukes av rutingsprotokollen for å holde tabellene oppdaterte) fra MIP-ARP og fra datapakker ved å sette **TRA**-bittene til **010** (dvs. ikke transport, ikke ARP, men routing). Rutingsprosessen begynner så snart alle nødvendige prosesser er startet på et system. MIP-daemonet må videresende transportpakker (TRA = 100) i henhold til den aktuelle tilstanden i rutingtabellen, eller sende dem til applikasjonen når destinasjonsadressen i pakken er adressen til hosten selv. Som i den obligatoriske oppgaven skal pakker droppes hvis ingen applikasjon lytter etter pakker.

Hvis MIP-daemonet videresender en pakke i sin rolle som ruter, så skal TTL (Time To Live)-feltet i pakken reduseres med 1. Hvis TTL-en når -1, må MIP-daemonet droppe pakken istedenfor å videresende den. Dette er nødvendig for å unngå at pakker går i evig løkke hvis det oppstår feil i rutingtabellene.

Det gis bonuspoeng til løsninger der hverken MIP-ARP-oppslag eller rutingsoppslag for videresending blokkerer videresending av andre pakker som ankommer senere (dvs. hvis oppslagene skjer asynkront).

Du skal levere følgende:

1. Et designdokument som inneholder:
 - En forside med kandidatnummer, oppgavetittel, kurs og semester. **Vi skal ikke ha navn eller brukernavn.**
 - En beskrivelse av din implementasjon av Distance Vector Routing (DVR) with Split Horizon.
 - Hvordan programmet er designet. Gjerne med en tegning (flytdiagram) som viser i hvilken rekkefølge de forskjellige funksjonene blir kalt.
 - Dokumentasjon av hvordan programmet skal startes og evt. avsluttes.

- Hvilke filer programmet består av (C-filer, headerfiler osv.).
 - Eventuelle andre særegenheter.
2. Kildekoden, hvor koden er behørig kommentert (der det er nødvendig), og en README fil som inneholder kort informasjon om hvordan programmene kjøres. Dokumenter alle variabler og definisjoner. For hver funksjon i programmet skal følgende dokumenteres:
- Hva funksjonen gjør
 - Hva inn- og utparametre betyr og brukes til
 - Hvilke globale variabler funksjonen endrer
 - Hva funksjonen returnerer

Om poengfordelingen:

- En ruter har to oppgaver: Ruting og forwarding. Forwardingdelen er enklere enn DVR implementasjonen (inkl. avstandsmåling) og er derfor mindre viktig.
- Det gis ikke poeng for delene av obligen som må gjenbrukes (f.eks. pingapplikasjonen).
- Korrekt bruk av DVR med Split Horizon for å regne rutingtabellene vil kunne gi ekstrapoeng.
- Ekstra poeng:
 - Etter at ARP brukes for første gang, bør det ikke oppstå noe ekstra forsinkelse ved senere utsendelser over samme sti. Kan sjekkes vha. ping.

Spesifikke krav til koden

Koden din skal kompileres på og vil bli testet på den samme virtuelle maskinen som ble brukt i den obligatoriske oppgaven.

Orakeltjeneste og administrative spørsmål

Får du problemer under implementasjonen anbefaler vi deg å sjekke FAQ på orakelsiden som ligger under INF3190 sin hjemmeside. Du kan også stille spørsmål på Piazza. Benytt de ukentlige orakeltimene godt.

For administrative spørsmål rundt hjemmeeksamen, ta kontakt med kristahi [at] ifi.uio.no

Besvarelse

Designdokumentet skal redigeres vha. et egnet verktøy, f.eks. LaTeX, OpenOffice, Word, osv. Dokumentet skal inneholde besvarelsen med alle etterspurte detaljer, samt ha en forside hvor følgende opplysninger er angitt: kandidatnummer, oppgavetittel, kurs og semester.

Før levering skal dokumentet konverteres til **PDF**-format.

Omfanget av dokumentet behøver ikke nødvendigvis være så stort, men det må inneholde tilstrekkelig informasjon til å oppfylle kravene som beskrevet under avsnittet *oppgave*. Det som er viktig er å kunne dokumentere forståelse for de temaene oppgaven berører, i tillegg til selve gjennomføringen. Vi stiller krav til ryddighet og struktur.

Elektronisk innlevering

Eksamen er **anonym**. Bruk [idchecker.sh](#) for å undersøke om navnet ditt er i koden (`./idchecker.sh`

`dittBrukernavn` og `./idchecker.sh enLitenDelAvDittNavn`). Scriptet undersøker om navnet ditt eksisterer i .tex, .c, .h og .txt-filer i mappa (husk å legge til eventuelle andre filformater du bruker). Det er ditt ansvar at din innlevering ikke inneholder personlig informasjon.

Alt skal leveres elektronisk hvor alle filer (Makefile, *.c, *.h, design.pdf, README.txt, etc.) er samlet i **én** katalog med **kandidatnummeret** som navn. Av denne katalogen lager du en komprimert tar-ball. Den elektroniske innleveringen skal leveres via web. Linken [Innlevering av oppgaven](#) finnes på kursets hjemmeside.

Etter innlevering anbefales det at du laster ned arkivet du leverte inn, pakker det ut og undersøker at alt fungerer (inkludert at README og design er med). Det anbefales også å levere inn utkast av eksamen i god tid før fristen i tilfelle du får tekniske problemer eller gjør en alvorlig feil rett før innlevering.

Innleveringsfrist: fredag 6. april 2018 kl 23:59.

Merk at denne tidsfristen er **HARD**, oppgaver levert etter fristen vurderes med karakteren "F", altså stryk.