



Webprotokollers ytelse over mobile pakkesvitsjede bærertjenester

Hovedoppgave ved sivilingeniørutdanningen i
Informasjons- og Kommunikasjonsteknologi
ved Høgskolen i Agder

Av
Jannicke Knutsen
Våren 2002

1 Forord

Arbeidet med hovedoppgaven har vært en lærerik tid. I tillegg til alt jeg har lært om TCP, RLC, LLC, GPRS, UMTS osv, har jeg erfart både fordeler og ulemper med å arbeide alene på et prosjekt. Jeg har lært å arbeide mer selvstendig, men synes det har vært en tung oppgave å skrive rapport alene.

Jeg vil rette stor takk til alle som har hjulpet med og støttet meg denne perioden. Tusen takk til Trond Botheim, som har vært veileder og som jeg har arbeidet sammen med hos NetCom. Takk til Frithjof Fjellstad, Ragnar Johnsen og Per Henrik Hogstad, som har kommet med råd og vink til hvordan ulike oppgaver kunne løses.

Tusen Takk til alle på Nordre Viken.

En stor Tusen Takk til mine beste skolekamerater Kjetil Ribe og Anders Henriksen, som jeg ikke hadde klart meg uten!

Og til slutt en TusenTusen Takk til Tim for moralsk støtte og en utrolig tålmodighet!!

Jannicke Knutsen

Grimstad, Mai 2002

2 Sammendrag

Motivasjonen for studier av webtrafikk over mobil pakkesvitsjede bærertjenester er at utnyttelsesgraden, spesielt med transportprotokollen TCP (Transmission Control Protocol) over radio, generelt blir ganske dårlig. Effektive datarater er betydelig lavere enn det som forventes teoretisk. Målet med oppgaven var å undersøke utvalgte web- og Internett-protokollers ytelse over GPRS og UMTS ved å modellere, optimalisere og eksperimentelt verifisere protokollene.

Rapporten er bygget opp rundt ulike protokoller og pakke-data-tjenester. De viktigste egenskaper og punkter fra litteraturen er tatt med for å gi forståelse av hvordan de virker, og videre fortelle hvordan de påvirkes av ulike web-protokoller, da i hovedsak TCP. Fokus er ytelsen i opplink og nedlink i form av overføringshastighet. Her ligger problemet i at ytelsen ikke er stabil i begge retninger. Et av målene har vært å undersøke hvorfor det er ustabil, ved en kvantitativ modell for forsinkelse og ytelse i opplink og nedlink. Utgangspunkt for modelleringen baserte seg på å finne sannsynligheter for at forsinkelser vil oppstå.

Nedlinkens ytelsen er en viktig måleenhet fordi brukere, da spesielt mobilbrukere, i hovedsak vil bruke Internett til å produsere trafikklast på nedlinken. Datapakker som ikke kan overføres øyeblikkelig, blir satt i kø ved kilden. Når køen blir full, kan det bli behov for retransmisjoner. Retransmisjon av pakker gir ekstra forsinkelse. Forsinkelse gir igjen reduksjon i ytelse. For hvert datasegment som mottas, genereres umiddelbart en kvittering, en ack. Ack'er fra nedlink blir buntet sammen mens de venter i køen bak datapakkene i opplink, og vice versa. Denne effekten kalles *ack compression* og reduserer den helhetlige ytelsen til TCP. Vindusstørrelsen i opplink og nedlink må være store nok til å fylle round trip pipe for at ack compression skal kunne oppstå. Vindusstørrelsen er avhengig av Internett-protokollens maksimale overføringsstørrelse (IP MTU).

Kommunikasjonssystemet ble modellert, og resultatene ble eksperimentelt verifisert. TEMS™ Investigation ble brukt til å se på RLC og LLC's oppførsel, bl.a. dataflyten i opplink og nedlink, i sanntid.

Området har vært viktig å undersøke fordi man ikke vet hvordan RLC og LLC oppfører seg over GPRS når radiokvaliteten målt i C/I er dårlig. I stabil tilstand er det lite eller ingen feil, og alt går knirkefritt. Normalt vil man ligge i stabilt område, men fenomenene med ulike C/I-verdier er interessante å studere på grunn av slow start og congestion, for da stemmer ikke forsinkelsene mer.

Det er et faktum at webtrafikk går dårligere i situasjoner med retransmisjon av IP. Resultatene etter å ha studert RLC, LLC og TCP analytisk i forhold til det som ble observert på målinger, viste seg å bero på andre forhold enn det som var forventet i utgangspunktet. Det viser seg at flytkontrollen på TCP nivå har mye større innvirkning enn antatt. Den antatte variasjonen i overføringstiden for LLC rammer, ser ut til å ha liten innvirkning på sannsynligheten for oppførselen til flytkontrollmekanismene i TCP. Enveis TCP-sesjoner gir ytelse nær det maksimale for aksessforbindelse. Men; dersom det pågår toveis trafikk, som vil være vanlig ved web-applikasjoner, reduseres ytelsen til TCP betraktelig gjennom langsom slow start- og congestion recovery-mekanismer

Innholdsfortegnelse

1	Forord.....	II
2	Sammendrag	III
3	Innledning.....	VI
3.1	Bakgrunn for oppgaven.....	VI
3.2	Mål	VI
3.3	Avgrensninger.....	VII
3.4	Veiledning til leseren	VII
4	Protokoller og pakke­datatjenester.....	1
4.1	GPRS – General Packet Radio Service.....	2
4.1.1	Radiogrensesnittet i GPRS.....	3
4.1.2	Protokollstakken til GPRS	3
4.2	UMTS - the Universal Mobile Telephony System	8
4.2.1	UMTS arkitektur.....	10
4.2.2	UMTS bærertjenester.....	11
4.2.3	UMTS Terrestrial Radio Access Network – UTRAN	12
4.2.4	WCDMA – Wideband Code Division Multiple Access.....	13
4.3	Transportprotokoller	14
4.3.1	TCP – Transmission Control Protocol.....	14
4.3.2	UDP.....	18
5	Modellering av kommunikasjonssystemet.....	19
5.1	Systembeskrivelse.....	19
5.2	Matematisk modell for RLC og LLC.....	20
5.3	Matematisk modell for TCP.....	27
6	Eksperimentell verifisering av resultater	30
6.1	Måling over GPRS med MTU = 1457 oktetter.....	32
6.1.1	Observasjoner ved målinger med C/I = 12 dB.....	33
6.1.2	Observasjoner ved målinger med C/I = 10 dB.....	34
6.1.3	Observasjoner ved målinger med C/I = 8 dB.....	35
6.2	Målinger over GPRS med MTU=576 oktetter.....	36
6.2.1	Observasjoner ved målinger med C/I = 12 dB.....	36
6.2.2	Observasjoner ved målinger med C/I = 10 dB.....	36
6.2.3	Observasjoner ved målinger med C/I = 8 dB.....	37
6.3	Kommentar til målinger.....	39
6.4	Eksperimentelt resultat av BLER som funksjon av C/I.....	39
6.5	Simulering av web-trafikk	40
7	UMTS	41
8	Diskusjon	43
9	Konklusjon.....	43
9.1	Videreføring.....	44
10	Referanser	45
11	Ordliste/forkortelser	46
12	Vedlegg.....	48

Figurliste

Figur 1 – Nettverkskonfigurasjon	2
Figur 2 – Protokollstakken mellom SGSN og MS.....	3
Figur 3 – Overhead på de ulike kommunikasjonslagene	7
Figur 4 – Domenemodell av UMTS arkitektur	10
Figur 5 - Arkitekturen til UMTS bærertjeneste	11
Figur 6 - UTRAN arkitektur	12
Figur 7 – Signaleringslag	16
Figur 8 - GPRS arkitektur	19
Figur 9 - Forventet oppholdstid for en LLC ramme i systemet	24
Figur 10 - RTT for GPRS	26
Figur 11 – Laboratorieoppsett.....	30
Figur 12 – Utsnitt fra loggfil med $C/I = 12\text{dB}$	33
Figur 13 - Utsnitt fra loggfil med $C/I = 10\text{ dB}$	34
Figur 14 - $C/I = 8\text{dB}$	35
Figur 15 - $C/I = 12\text{ dB}$	36
Figur 16 - $C/I = 10\text{dB}$	37
Figur 17 - $C/I = 8\text{dB}$	38
Figur 18 - BLER som funksjon av C/I	40
Figur 19 – RTSP, real-time streaming protocol over UDP over UMTS	41
Figur 20 - Statistikk for UMTS.....	42
Figur 21 - RTT i UMTS.....	42

Tabelliste

Tabell 1 - Coding Schemes	6
Tabell 2 - QoS klasser i UMTS.....	9
Tabell 3 – Teoretiske verdier	21
Tabell 4 - Utregnede, teoretiske verdier	22
Tabell 5 - RTT avhengig av LLC størrelse.....	26
Tabell 6 - Parametre for $MTU = 576\text{ byte}$	27
Tabell 7 - Parametre for $MTU = 1457$	28
Tabell 8- Betingelser for hvilke deler av F_u som gjelder	28
Tabell 9 – Målingselementer	31
Tabell 10 - Ytelse ved simulert http-trafikk	40

3 Innledning

Denne prosjektrapporten er skrevet som avsluttende hovedoppgave ved sivilingeniørutdanningen i Informasjons- og kommunikasjonsteknologi, IKT, ved Høgskolen i Agder, Grimstad. Oppgaven har gått ut på å se på webprotokollers ytelse over mobile pakkesvitsjede bærertjenester.

De to hovedmåleenhetene for nettverk er overføringshastighet og forsinkelse. Vi ønsker selvsagt minst mulig forsinkelse og mest mulig ytelse. En av oppgaven var å undersøke transportprotokollen *Transmission Control Protocol's* overføringshastighet og filoverføringsforsinkelse over en trådløs ende-til-ende forbindelse. Det er viktig å sette fokus på ytelsen i opplink og nedlink. Her ligger problemet i at ytelsen ikke er stabil i begge retninger. Det kan virke som at å øke ytelsen vil redusere forsinkelsen, men det er ikke tilfelle. Et av målene har vært å undersøke hvorfor det er på denne måten.

Et utvalg av relevante protokoller har blitt identifisert og beskrevet. Kommunikasjonssystemet ble modellert med hensyn på spesifikke, variable størrelser knyttet til utvalgte protokoller og kanal-karakteristikker.

Eksperimentene ble utført både i et kontrollert miljø i NetComs produksjonsnett og i live-nett. Den eksperimentelle delen vil konsentrere seg om radionettet, som antas å representere de største bidrag til forsinkelse og hastighet.

3.1 Bakgrunn for oppgaven

Motivasjonen for studier av webtrafikk over mobil pakkesvitsjede bærertjenester er at utnyttelsesgraden, spesielt med TCP over radio, generelt blir ganske dårlig. Effektive datarater er betydelig lavere enn det som forventes teoretisk. UMTS er på stadig frammarsj, og NetCom har sammen med Siemens startet opp et prosjekt for å optimalisere radiogrensesnittet i UMTS.

3.2 Mål

Målet med oppgaven var å undersøke utvalgte web- og internettprotokollers ytelse over GPRS og UMTS ved å modellere, optimalisere og eksperimentelt verifisere protokollene.

3.3 Avgrensninger

I midten av prosjektperioden ble det klart at samarbeidsprosjektet mellom NetCom og Siemens, som var antatt å starte i månedsskiftet april/mai 2002, ble utsatt med minst en måned, noe som fikk konsekvenser for å utføre undersøkelser i UMTS.

På grunn av forsinkede leveranser av blant annet software-releaser etter åpen 3GPP-standard for UTRAN fra NetComs leverandører, var det ikke mulig å skaffe til veie instrumenter og software for målinger, observasjoner og konfigurasjon av UTRAN, som ville tillate å sammenholde observasjonene med benyttede parametre og protokollenes interne oppførsel. Derfor ble det denne delen av den planlagte oppgaven nedprioritert til fordel for GPRS.

Når man gjør fullstendige analytisk beregninger, kan man senere se på de samme fenomenene ved hjelp av målinger. Utgangspunktet for teoretiske modeller og beregninger på lag to er [8]. På høyere lag vil det være altfor ressurskrevende og også meget vanskelig å lage fullstendige, matematiske modeller. Dette er pga. de høyere lags kompleksitet.

3.4 Veiledning til leseren

For å forstå webprotokollenes virkemåte, og hva, hvordan og på hvilken måte disse virker over mobile nett, har oppgaven ført til mye litteraturstudier. Rapporten er derfor bygget opp rundt ulike protokoller og pakke-data-tjenester. De viktigste egenskaper og punkter fra litteraturen er tatt med for å gi en forståelse av hvordan de virker og videre fortelle hvordan de påvirkes av ulike web-protokoller, da i hovedsak TCP. Videre følger en modelleringsdel, der matematiske modeller er beskrevet. Påfølgende kapittel omhandler den eksperimentelle delen, med verifisering av matematiske resultater.

Rapporten avsluttes med et diskusjonskapittel, der konklusjon og forslag til videreføring er fremlagt. Referanser og Ordliste/Forkortelser finnes bakerst i rapporten.

4 Protokoller og pakke­datatjenester

Den hurtigvoksende bruken av Internett har framkalt behov for trådløs tilgang til Internett. Det er realistisk å tro at den dominerende Internett tilkoblingen i framtiden vil være trådløs. Hver eneste mobiltelefon og hver eneste lille *ting* som kan kommunisere vil ha en IP protokollstakk og et trådløst grensesnitt.

Historien til mobiltelefoni dateres tilbake til begynnelsen av 1980-årene, da systemer som *Nordic Mobile Telephony* (NMT) og *Total Access Communication Systems* (TACS) ble introdusert. Dette refereres til som første generasjons mobilsystem, og kjennetegnes med et analogt radiogrensesnitt.

2. generasjons systemer ble introdusert i begynnelsen av 1990-årene. Eksempler på disse systemene er det verdensomspennende systemet *Global System for Mobile Communication* (GSM) og det japanske systemet *Personal Digital Cellular* (PDC). Kjennetegnet til andre generasjons systemer er at radiogrensesnittet er digitalt. Nye tjenester, som SMS og pakkesvitsjede datatjenester, ble tilbudt i tillegg til tale. Internett-revolusjonen stilte, og stiller fortsatt, nye krav til det mobile telenettet. GSM tilbyr en fast mengde datatjenester, fører de til ineffektiv bruk av båndbredde for databrukere på grunn av den bristende datatrafikken. Det ble behov for å ta i bruk pakkesvitsjede tjenester for å håndtere datatrafikken på en mer effektiv måte, og en ny databærer i GSM nettet ble utviklet. *General Packet Radio Service* (GPRS) ble introdusert for å optimalisere Internett/Intranets tilgangskvalifikasjoner, og blir betegnet som generasjon 2,5.

International Telecommunication Unit, ITU[2] utviklet ideen om IMT-2000 (International Mobile Telecommunications-2000) som 3. generasjons mobilsystem. Dette systemet vil gi tilgang, ved bruk av en eller flere radiolinker, til et bredt spekter av telekommunikasjonstjenester som støtter det stasjonære telekommunikasjonsnettverket (f.eks. PSTN/ISDN/IP), og til andre tjenester som er tilrettelagt mobile brukere. F.eks. vil krav om høyere bitrater og en mer fleksibel tjenesteplattform ivaretas. Visjonen til tredje generasjons mobilsystem er å oppfylle drømmene om å gjøre hvor-som-helst- og når-som-helst-kommunikasjon til en realitet. IMT-2000 er resultatet av samarbeid mellom mange enheter både innenfor og utenfor ITU. Utfallet av dette førte til utformingen av *Universal Mobile Telephony System* (UMTS), som er den europeiske og japanske varianten av ITUs IMT-2000. En kort gjennomgang av GPRS og UMTS følger nedenfor.

4.1 GPRS – General Packet Radio Service

Som nevnt ble GPRS implementert i GSM som en bærer for å håndtere datatrafikken på en mer effektiv måte, og er dynamisk kanal-allokering for datapakker. GPRS er et pakkesvitsjet dataoverføringssystem og er ikke beregnet for overføring av sanntidsinformasjon, som tale. I utgangspunktet er GPRS basert på prinsippet med hybrid svitsjing [1], som tar utgangspunkt i to typer trafikk: (i) taleanrop og (ii) pakke-data. I praksis, og i samsvar med nåværende design og trafikkstyring, har taleanrop prioritet over datapakker.

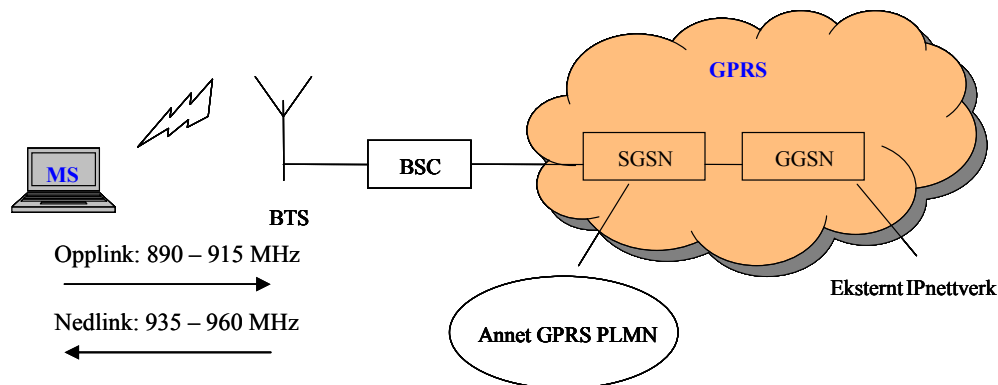
GPRS tilbyr i utgangspunktet to hovedtyper av tjenester: punkt-til-punkt (PTP) og punkt-til-multipunkt (PTM).

Det er tatt i betraktning at det alltid vil være kø av datapakker. Datapakker som ikke kan overføres øyeblikkelig, blir satt i kø ved kilden. I virkeligheten vil det ikke alltid være sånn. Når køen blir full, kan det bli behov for retransmisjoner. Retransmisjon av pakker gir ekstra forsinkelse. Forsinkelse gir igjen reduksjon i ytelse. Hvordan TCP oppfører seg i GPRS, blir beskrevet i 4.3.1.

Hovedfordelene med GPRS er flere:

- ✓ Tilgang til standardisert, trådløst, pakkesvitsjet Internett/Intranett overalt hvor GSM er tilgjengelig.
- ✓ Variabel maksimal datahastighet, fra noen få bit per sek. opp til 171,2 kbit/s.
- ✓ Mulighet for volumbasert betaling gir brukeren mulighet til å være tilkoblet hele tiden og kun betale for overførte data.
- ✓ Mulighet til å anvende både eksisterende tjenester så vel som og nye applikasjoner.
- ✓ IP-pakker over radiogrensesnittet, samt optimal deling av radioressurser
- ✓ Pakkesvitsjet teknologi i nettverkets infrastruktur, samt optimal deling av nettverksressurser.
- ✓ Muligheten til utvidelse med fremtidige radioprotokoller.

Denne rapporten bygger på nettverkskonfigurasjonen i Figur 1, med overføring av data fra MS til SGSN/GGSN i både opplink og nedlink.



Figur 1 – Nettverkskonfigurasjon

4.1.1 Radiogrensensnittet i GPRS

GPRS benytter samme 200kHz radiokanaler og radiospektrum som GSM. Grensesnittet er en kombinasjon av frekvens- (FDMA) og tids- (TDMA) multipleksing.

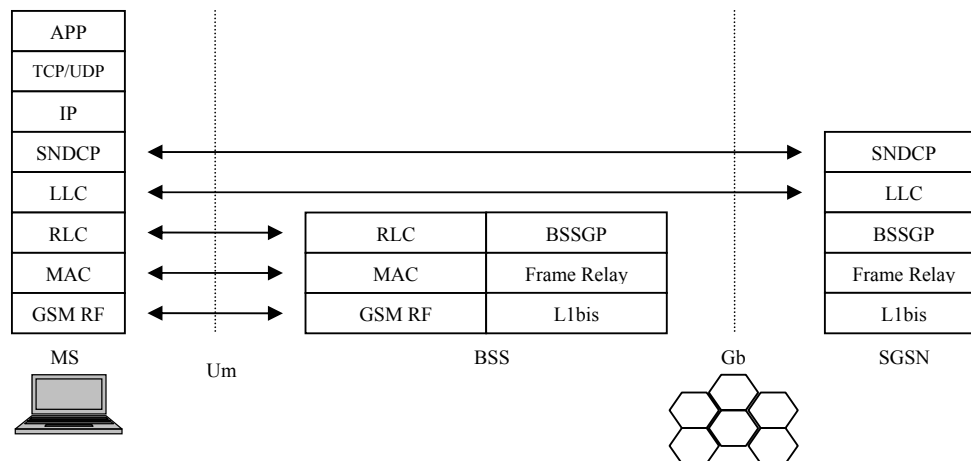
GPRS gir mulighet til å kombinere opptil 8 tidsluker i hvert tidsintervall for IP-basert pakke data, med hastighet opp til maksimal, teoretisk verdi på 171.2 kbit/s. TDMA deler bærerfrekvensen opp i 8 kanaler. En TDMA-ramme inneholder en tidsramme fra hver av de 8 kanalene, og er 4.615 ms lang. Tidslukene fra hver kanal er 0.577ms lange. Men; en typisk GPRS-oppkobling vil ikke benytte seg av alle 8 tidslukene. En mulig konfigurering kan være en tidsluke – max 20 kbit/s til typisk 14.4 kbit/s – for opplink og fire tidsluker – max 80 kbit/s til typisk 57.6 kbit/s – for nedlink[18]. I nettverks- konfigurasjonen denne rapporten er skrevet for, ble det benyttet en tidsluke i opplink og tre tidsluker i nedlink. Siden GPRS benytter ulike tidsluker på opplink og nedlink, støtter den asymmetrisk datatrafikk.

Bærefrekvensen er separert med 200 kHz på hver side. Frekvensene er allokert i par, slik at hvert opp/nedlink par er separert med eksakt 45 MHz.

4.1.2 Protokollstakken til GPRS

Radiogrensensnittet i GPRS består av en lagdelt protokollstruktur som tilbyr ulike kontrollprosedyrer. Siden GPRS benyttes som bærer ved overføring av data, er tanken at en har en IP forbindelse fra ende til ende. Kommunikasjonen internt i GPRS-nettet foregår ved bruk av IP over IP (mellom SGSN og GGSN).

Protokollene som bør vurderes i analyser av radiogrensensnittets ytelse er vist i Figur 2 [4], og benyttes for dataoverføring i GPRS mellom MS og SGSN.



Figur 2 – Protokollstakken mellom SGSN og MS

De ulike delene av protokollstakken er:

- TCP - Transmission Control Protocol
- UDP - User Datagram Protocol
- IP - Internet Protocol
- SNDCP - Subnetwork Dependent Convergence Protocol
- LLC - Logic Link Control
- RLC - Radio Link Control
- MAC - Medium Access Control
- GSM RF - Radiobæreren
- BSSGP - Base Station System GPRS Application Protocol
- L1Bis - Fysisk lag

Um og Gb er to av grensesnittene i GPRS.

I det kommende avsnittet blir det gitt grundigere beskrivelse av de viktigste protokollene. Transportprotokollene TCP og UDP blir beskrevet i 4.3.

SNDCP – Subnetwork Dependent Convergence Protocol

SNDCP mapper nettverkslagets pakkeenheter, PDU'er, til det underliggende LLC laget. Protokollen oversetter fra rammer til IP pakker og motsatt. I tillegg har den en valgfri komprimeringsfunksjon for både TCP/IP header og datainnhold.

Komprimerings-funksjonen blir ikke benyttet.

LLC – Logical Link Protocol

LLC laget gir en pålitelig, kryptert link mellom SGSN og MS, og er uavhengig av de underliggende radiogrensesnitt protokollene. LLC laget leverer kun fra seg korrekte rammer, og sørger dermed for feilfri pakkeoverføring til SNDCP.

LLC kan operere i både *acknowledged* og *unacknowledged* mode. I nettet kjøres kun *acknowledged* mode, som betyr at LLC alltid er feilfri. Rammestørrelsen er varierende. Hvordan blir så LLC rammestørrelse bestemt? LLC rammens størrelser bli tilsvarende like store som IP pakkene. LLC's maksimalverdi er 1506 oktetter.

RLC – Radio Link Control

RLC laget er et nøkkellag i radiogrensesnittet fordi det har ansvar for kommunikasjon over radiolinken mellom MS og BSS. RLC gir pålitelig overføring av data ved å bruke en valgfri ARQ mekanisme, som er en feilkontroll-mekanisme. Feilsjekk foregår på RLC nivå før blokkene sendes videre til LLC. Den komplette LLC rammen må ha blitt overført korrekt over radiogrensesnittet av RLC protokollen, før den blir videresendt til LLC laget. I tillegg foretar RLC segmentering og reassemblering av data til eller fra LLC laget. I segmenteringen inngår det å danne RLC blokker som mappes til fysisk lag. Reassemblering og feilsjekk medfører varierende forsinkelser. I motsetning til LLC, har RLC fast vindusstørrelse, og består av 64 blokker [9].

Hentet fra 3GPP [9] er RLC funksjonen ansvarlig for følgende:

- segmentering av LLC PDUer til RLC datablokker og re-assemblering av RLC datablokker til LLC PDU'er.
- segmentering av RLC/MAC kontroll meldinger til RLC/MAC kontroll blokker og reassemblering av RLC/MAC kontroll meldinger fra RLC/MAC kontroll blokker.

- Backward Error Correction, BEC, gir mulighet for selektiv retransmisjon av RLC datablokker.

MAC – Medium Access Control laget

MAC laget kontrollerer MS' tilgang til radiogrensesnittet, og tilbyr scheduling (tidsplanlegging) til den tilknyttede signaleringen.

GSM RF

GSM RF er det fysiske GSM TDMA grensesnittet. Bit interleaving, modulasjon/demodulasjon og regulering av sendereffekt er eksempler på funksjonaliteter på dette laget.

RLC/MAC protokoller for GPRS

I dette avsnittet blir det gitt en forklaring på hvordan RLC/MAC protokollen virker over GPRS. Teksten bygger på artikkel [8], som også ble benyttet som basis for matematisk utredning av RLC og LLC.

Opplink og nedlink blir i utgangspunktet brukt som uavhengige kanalressurser. I en bestemt TDMA tidsluke, kan PDCH(Packet Data Channel) i opplink bære data fra en MS, og nedlinkdata til en annen MS. Uansett er allokering av opplink og nedlink ressurser nødvendig for å tillate enkle MS'er å overføre pakker i begge retninger samtidig.

Høy fleksibilitet sikres ved å implementere multi-slot RLC/MAC-lag operasjon som gir mer enn en PDCH (eller TS i en TDMA ramme) som skal brukes i parallell med en MS for overføring av pakke data.

RLC-data som tilhører samme LLC-ramme kan overføres i parallell på ulike PDCH'er. Etter å ha mottatt et visst antall RLC-datablokker, blir MS pollet til å svare med ack som nettverket har forhåndsreservert passende opplink-kapasitet til. Ack'en kan inneholde et bitmønster som indikerer ukorrekte RLC-blokker. Disse blokkene må retransmitteres av nettverket.

MAC mekanismen har primært innvirkning på ytelsen i opplink, slik at det for ytelsen i nedlink kun er relevant å betrakte RLC protokollen.

Hver radioblokk består av 4 burster. En radioblokk inneholder en eller to RLC blokker avhengig av coding scheme. RLC retransmitterer datablokker med uopprettelige feil.

RLC blokker kodes i samsvar med en av de fire skjemaene gitt i Tabell 1 - Coding Schemes [8].

Channel Coding Scheme	Fill bits per RLC block	Fill bytes	RLC/MAC Data rate
CS-1	160 bits	20 bytes	9,05 kbit/s
CS-2	240 bits	30 bytes	13,4 kbit/s
CS-3	288 bits	36 bytes	15,6 kbit/s
CS-4	400 bits	50 bytes	21,4 kbit/s

Tabell 1 - Coding Schemes

CS-1 gir høyeste beskyttelse mot feil, mens CS-4 ikke gir beskyttelse i det hele tatt. Konsekvensen av bruken av coding schemes er at brukerdataraten øker med høyere coding scheme på bekostning av kravet til C/I nivået (som er et mål for radiolink-kvaliteten).

Maksimal ytelse per tidsluke er avhengig av hvilke coding scheme som er implementert [12]. Dersom kun CS-1 og CS-2 er implementert, er det klart at CS-2 gir bedre ytelse i de fleste tilfeller. I dette studiet er kun CS-1 og CS-2 blitt vurdert, da disse er de eneste som er implementert i nettverket.

Det er kjent at jo verre radioforholdene er, jo høyere er antall retransmisjoner. CS-1 har lav bitrate, som gir få retransmisjoner. CS-2 har høyere bitrate, som gir flere retransmisjoner. C/I må være dårligere enn 6,5 dB før CS-1 gir høyeste rate [12]. I et miljø uten frekvenshopping vil C/I være enda lavere, og CS-2 vil derfor gi bedre ytelse i alle tilfeller der forbindelsen kan opprettholdes.

Mobilkommunikasjonen foregår på følgende måte:

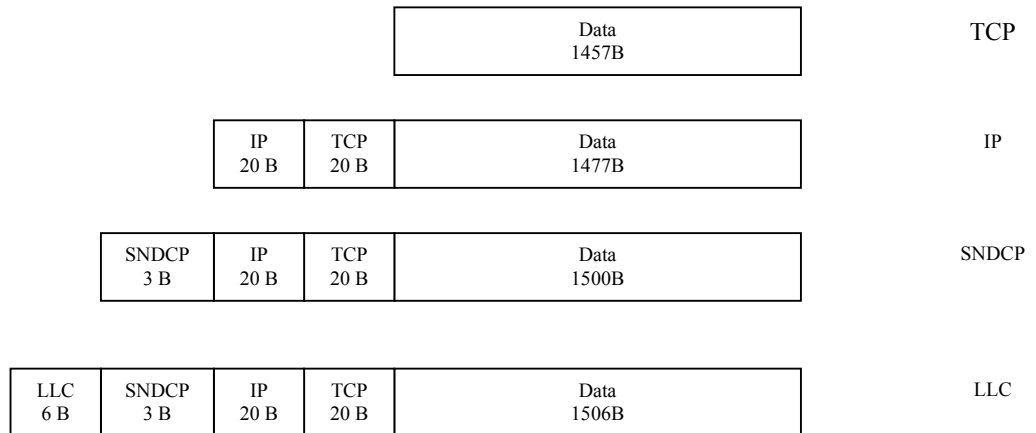
Utnyttelsesgraden har tre modi;

Pakker som overføres fra nettverkslaget, f.eks. IP, overføres over radiogrensesnittet mellom MS og SGSN.

En pakkesvitsj gir direkte grensesnittet til BSC ved å bruke LLC protokollen. I RLC/MAC laget blir LLC rammene segmentert til RLC-datablokker, som igjen formateres til burster på fysisk lag.

Hver blokk komprimerer 4 normal burster i en etterfølgende TDMA ramme. Hvordan en datapakke brytes ned i segmenter og legges til overhead på de ulike lagene, er illustrert i Figur 3 – Overhead på de ulike kommunikasjonslagene. I dette studiet er det tatt utgangspunkt i den største LLC rammen man kan ha før det blir fragmentering.

Data på applikasjonsnivå kan bli delt i flere TCP og IP segmenter, avhengig av datavolum og TCP pakkestørrelse. Typisk mapper hver TCP-pakke en-til-en gjennom SDCP og LLC laget før de splittes opp til et antall RLC blokker. RLC blokkene blir lagt til en header i MAC laget. Deretter sendes de ut på radiogrensesnittet i fire burster, som igjen sendes over en utøvende TDMA rammer med en gjennomsnittlig varighet på 20ms [12].



Figur 3 – Overhead på de ulike kommunikasjonslagene

4.2 UMTS - the Universal Mobile Telephony System

ITU utviklet, som nevnt i innledningen av kapitlet, ideen om IMT-2000 som 3. generasjons mobilsystem, der UMTS er den europeiske og japanske varianten. Det mest kjente trekket ved UMTS er høyere brukerhastigheter.

UMTS tilbyr teletjenester, som tale og SMS, og bærertjenester som er kvalifiserer til informasjonsoverføring mellom aksesspunkter. Det er mulig å forhandle/reforhandle en bærertjenestes karakteristikk ved oppstart av sesjon eller etablering av forbindelse, og gjennom pågående sesjon eller forbindelse. Både forbindelsesorienterte og forbindelsesløse tjenester tilbys for punkt-til-punkt- og punkt-til-multipunkt-kommunikasjon. UMTS data vil være "always on".

UMTS datarater er på delt medium og det blir trafikkmessig deling mellom kundene. På linjesvitjet forbindelse kan 348 kbit/s oppnås, og opp til 2Mbit/s for pakkesvitsjet forbindelse [4]. Bærertjenestene har ulike QoS parametre for maksimal overføringsforsinkelse, forsinkelsesvariasjoner og bitfeilfrekvens. Tre typer aksess er definert med følgende tilbydde hastigheter[10]:

- *Rural outdoor* – makro-celler hvor hastigheter opptil 500 km/t tillates. I første fase skal en her få aksess til 144 kbit/s. Gjelder satellitt og landsdekkende område, full mobilitet
- *Suburban outdoor* – mindre celler hvor hastigheter opp mot 120km/t skal gi 348 kbit/s i fase 1. Gjelder for byområde/begrenset område
- *Indoor/Short range* – mikro eller pico celler hvor kapasiteten skal være minst 2Mb/s. Gjelder innendørs og for korte rekkevidder utendørs. Hastighet opp til 10 km/t tillates

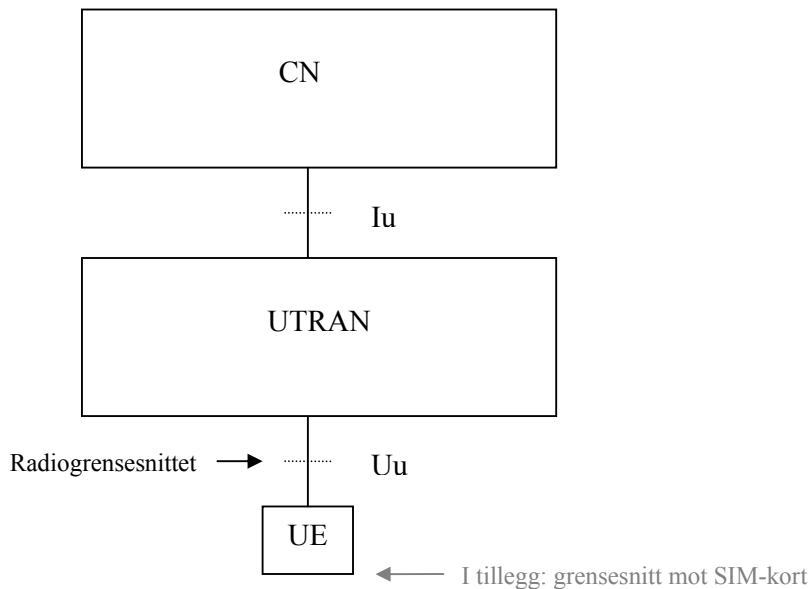
UMTS skal støtte et bredt spekter av tjenester med ulik karakteristikk. I tillegg til kravet om varierende bithastigheter, er det store variasjoner i kravene om forsinkelse og feil for de ulike tjenestene. Tabell 2 - QoS klasser i UMTS viser de fire trafikk-klassene som UMTS tilbyr, og er hentet fra [3]. Hovedforskjellen mellom de ulike trafikk-klassene er kravene til forsinkelse.

Trafikk klasse	Karakteristikk	Eksempel på applikasjon
Conversational	Beskytter tidsrelasjonene til informasjonsflyten Lite forsinkelse	Tale Videokonferanse Voice over IP
Streaming	Beskytter tidsrelasjonene til informasjonsflyten	Streaming av audio/video
Interactive	Beskytter innholdet av pay load Meldingen er ventet innen en bestemt tid ved destinasjonen, men forsinkelseskravene er ikke fastsatt	Web browsing Databasetilgang Nettverksspill
Background	Beskytter innholdet av pay load Destinasjonen forventer ikke data innen en viss tid	Bakgrunnsmodtagelse av e-post og SMS, samt nedlasting av data

Tabell 2 - QoS klasser i UMTS

4.2.1 UMTS arkitektur

Et UMTS nettverk består av tre samarbeidende domener: *Core Network (CN)*, *UMTS Terrestrial Radio Access Network (UTRAN)* og *User Equipment (UE)*, som vist i Figur 4 – Domenemodell av UMTS arkitektur, som tar utgangspunkt i figur 5.1 i [4].



Figur 4 – Domenemodell av UMTS arkitektur

GSM nettverk med GPRS er basis for nettverksarkitekturen. Alt utstyr må modifiseres for UMTS operasjoner og tjenester.

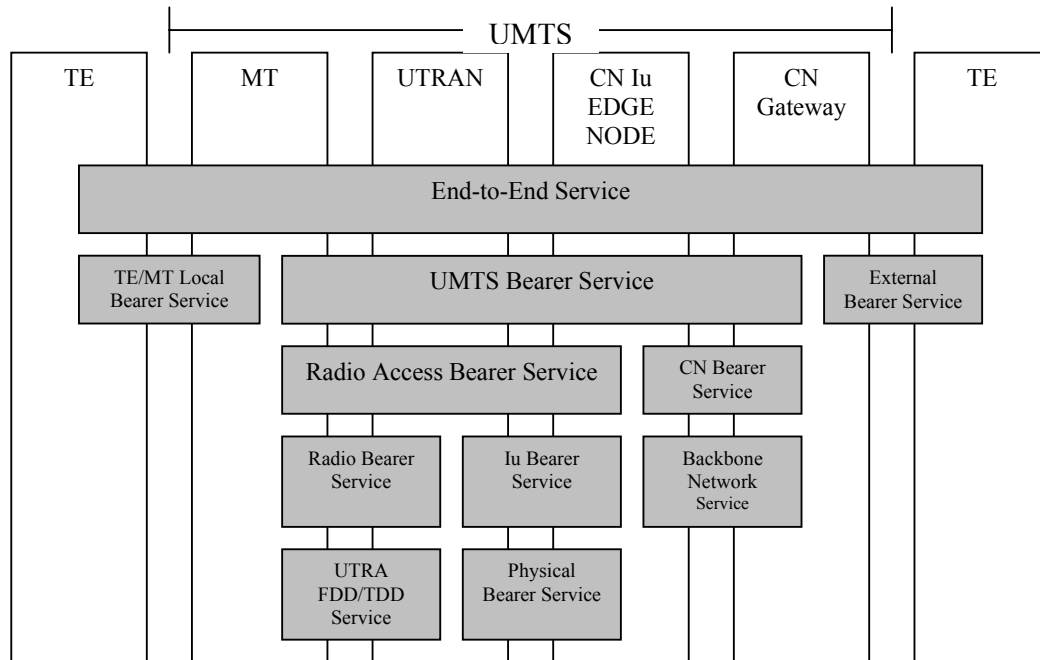
Hovedfunksjonen til CN, kjernenettverket, er å sørge for svitsjing, ruting og gjennomgang av brukertrafikk. Kjernenettet inneholder også databasene og nettverkets styringsfunksjoner. Kjernenettverket kan videre deles i et linjesvitsjet og et pakkesvitsjet domene.

UTRAN gir mobilutstyret aksess til radiogrensesnittet og består av to typer noder; *Radio Network Controllers*, RNCs, og *Node Bs*. En mer utfyllende beskrivelse av UTRAN finnes i avsnitt 4.2.3.

UE består av *Mobile Equipment (ME)* og *UMTS Subscriber Information Module (USIM)*. ME er terminalutstyret som kreves for radiomottagelse og –transmisjon over Uu grensesnittet, mens USIM er smartkortet som inneholder brukerinformasjon og autentiserings- og krypteringsnøkler.

4.2.2 UMTS bærertjenester

Tjenestearkitekturen til UMTS har et veldig fleksibelt og modulært design. Bærertjenestene gir ende-til-ende QoS, og basert på figur 2.2 i [4], er arkitekturen vist i Figur 4.



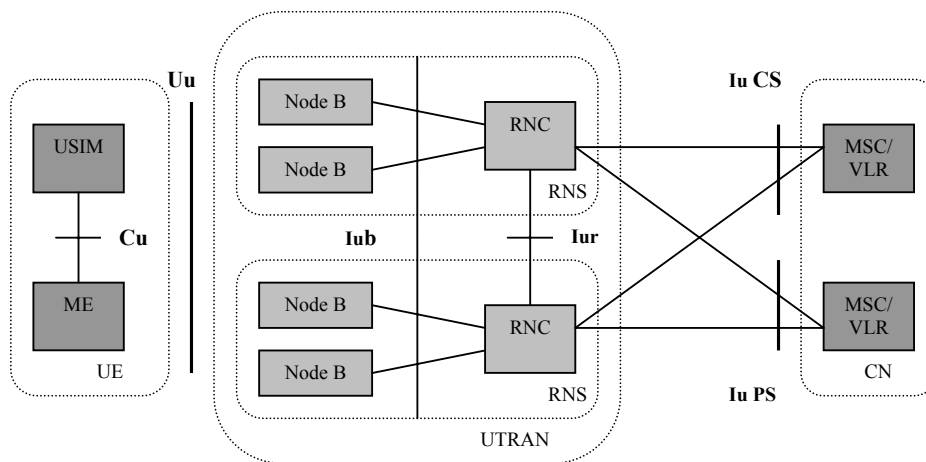
Figur 5 - Arkitekturen til UMTS bærertjeneste

Hver bærertjeneste på et spesifikt lag tilbyr sine individuelle tjenester ved å benytte seg av de som ligger på lagene under.

4.2.3 UMTS Terrestrial Radio Access Network – UTRAN

Radioaksessnettet tar hånd om alle radio-relaterte funksjonaliteter og består av to typer noder; *Node Bs* og *Radio Network* og *Controllers*. Nodene (Node B og RNC) og grensesnittene (Iu, Iur, Iub, Uu) vises i Figur 6 - UTRAN arkitekturog er hentet fra [4].

Noen av egenskapene til nodene er beskrevet under. (En kortfattet forklaring på de ulike grensesnittene finnes i Orddliste/forkortelser)



Figur 6 - UTRAN arkitektur

Node B: Node B terminerer Iub grensesnittet og tilsvarer BTS i GSM terminologi. Node B er ansvarlig for prosessering på fysisk lag, slik som kanalkoding og *interleaving*, hastighetstilpasning, spredning og modulering av signalet. Den inneholder også algoritmer til fysisk lag, som f.eks. *inner loop power control* og ulike målingsalgoritmer.

RNC: RNC korresponderer med BSC i GSM terminologien og er nettverkselementet som sørger for å ha kontroll med radio ressursene i UTRAN. Avhengig av rollen til RNC'en blir den referert til som enten *Controlling RNC* (CRNC), *Serving RNC* (SRNC) eller *Drift RNC* (DRNC).

CRNC kontrollerer en eller flere Node B med hensyn til bruk av ressurser og last i cellene. Dette inkluderer også adgangskontroll.

SRNC er den RNC'en som for en gitt UE terminerer både data og RANAP signaleringslinken for forbindelsen mot kjernenettet (CN).

DRNC er enhver *annen* RNC (enn SRNC) som kontrollerer celler som UE har kontakt med for eksempel i forbindelse med soft handover.

Hovedkarakteristikken til UTRAN har også vært hovedkravene for design av UTRAN arkitekturen, funksjonene og protokollene, og kan i henhold til [4] oppsummeres i følgende punkter:

- Støtte for UTRA og alle relaterte funksjonaliteter. Spesielt har det i designet av UTRAN blitt lagt stor vekt på kravene om å støtte soft handover (en terminal er tilkoblet nettverket via to eller flere aktive celler) og WCDMA-spesifikke *Radio Resource Management* algoritmer.
- Maksimering av felleshåndtering av pakkesvitsjede og linjesvitsjede data, med en unik protokollstakk i radiogrensesnittet og med bruk av samme grensesnitt for tilkoblingen fra UTRAN til både PS og CS domeneene i kjernenettverket.
- Maksimering av fellestrekk med GSM, om mulig
- Bruk av ATM transport som hovedtransportmekanisme i UTRAN.

4.2.4 WCDMA – Wideband Code Division Multiple Access

Wideband CDMA teknologi ble valgt til å gjelde for UTRAN radiogrensesnitt, og dermed for UMTS [4]. Uu grensesnittet er WCDMA radiogrensesnittet, og gjennom dette aksesserer UE den stasjonære delen av systemet. Dette gjør Uu til sannsynligvis det viktigste, åpne grensesnittet i UMTS.

WCDMA skal brukes i Europa og Asia, inkludert Japan og Korea, og benytte frekvensbånd på rundt 2GHz.

EDGE og multicarrier CDMA (CDMA2000) kan brukes i tillegg til WCDMA, men vil ikke bli tatt opp her.

Nedenfor er det listet opp et utdrag, hentet fra [4], over de ulike hovedparameterne som kjennetegner WCDMA.

WCDMA er et *Direct-Sequence CDMA system* (DS-CDMA), der brukerdata spres utover et vidt spekter ved at de multipliseres med pseudo-random bit utledet fra WCDMA *Spreading Codes*. I forhold til kanalisering i UMTS, benyttes koder til synkronisering og *scrambling* (scrambling brukes i tillegg til spredning for å skille signaler fra ulike kilder fra hverandre). WCDMA har to basis modier: frekvensdelt dupleks (FDD) og tidsdelt dupleks (TDD). I FDD brukes separate 5MHz bærerfrekvenser for opplink og nedlink respektivt, mens det i TDD bare er en tidsdelt 5MHz mellom opplink og nedlink. TDD er basert på FDD konseptet og ble lagt for å forbedre basic WCDMA systemet.

Chip hastigheten på 3.84 Mc/s gir en båndbredde på tilnærmet 5Mhz.

WCDMA støtter høyt varierende brukerdata-hastigheter, noe som betyr at konseptet *obtaining Bandwith on Demand (BoD)* støttes. Hver bruker tildeles rammer med 10ms varighet, der brukerhastigheten holdes konstant. Uansett kan datakapasiteten mellom brukerne variere fra ramme til ramme. Denne raske radiokapasitetstildelingen vil typisk bli kontrollert av nettverket for å oppnå optimal ytelse for pakke-datatenester. WCDMA operasjonene foretas i asynkrone basestasjoner, slik at det ikke er behov for globale tidsreferanser slik som for GPRS. Utvikling av innendørs og mikro-basestasjoner er enklere når det ikke er behov for å motta GPS signal.

WCDMA benytter koherent deteksjon på opplink og nedlink basert på pilotkanal.

Bruken av koherent deteksjon på opplink er nytt for offisielle WCDMA systemer og vil resultere i en generell deknings- og kapasitetsøkning på opplink.

Radiogrensesnittet i WCDMA har blitt utviklet på en slik måte at avanserte CDMA-mottakere, som multibrukerdeteksjon og smarte adaptive antenner, kan legges til av

nettverksoperatøren som et valgfritt tillegg i systemet for å øke kapasiteten og/eller dekingen.

Det er støtte for handover mellom GSM og WCDMA, for å være i stand til å ha innflytelse på GSM-dekkingen for WCDMA.

4.3 Transportprotokoller

4.3.1 TCP – Transmission Control Protocol

Transmission Control Protocol, TCP, er en av de to hovedprotokollene på transportlaget (lag 3) i den alternative Internettarkitekturmodellen [5], og den mest utbredte transportprotokollen idag. TCP tilbyr pålitelig, forbindelsesorienterte bytestrøm tjenester, [5]. TCP har med stor suksess blitt brukt i datanettverk for å sikre ytelse ende-til-ende, og brukes heller over Internett og på trådløs link enn over en punkt-til-punkt link. Protokollen er vindusbasert på den måten at mottakeren avterer et vindu til senderen. Vinduet tilsvarer hvor mye bufferplass mottakeren har, og begrenser på den måten hvor mye data senderen kan overføre til en gitt tid. Dette betyr at vinduet støtter flytkontroll. TCP deler pakker opp i segmenter og hver retning bærer et segment av byte-strømmen.

En avansert *congestion-control* mekanisme er implementert i TCP. Congestion kan ses på som en slags trafikkork, eller en overbelastning, som oppstår når for mange pakker kjemper om samme link, køen flyter over og pakker må droppes. Congestion control mekanismen, sammen med resource allocation er to sider av samme sak, som begge forsøker å ta seg av trafikkork-problemet.

Hver gang TCP sender et segment, starter en timer og venter på kvittering, heretter kalt ack. Dersom timeren utløper før dataene i segmentet har blitt kvittert, antar TCP at segmentet er tapt eller korrupt, og retransmitterer segmentet. TCP benytter *adaptive retransmisjon* for å tilpasse seg varierende forsinkelser. Den adaptive retransmisjonsalgoritmen kontrollerer forsinkelser for hver forbindelse og regulerer timeout-parameterne tilsvarende.

IP kaster pakker etter at *time to live* (TTL) utgår og TCP antar at hver pakke har en maksimal, begrenset levetid.

I hjertet av TCP er sliding window (glidende vindu) algoritmen, som protokollen benytter seg av på en ende-til-ende basis for å oppnå pålitelig og ordnet levering. Mistes pakken, utløper timeren og senderen retransmitterer den. Pakker kan havne i feil rekkefølge når de sendes over Internett. Sliding window ordner pakkene korrekt ved hjelp av pakkenes sekvensnummer. Denne algoritmen blir iverksatt under etablering av en forbindelse, som opprettes ved treveis handshake. Sliding window utnytter nettverkets båndbredde godt ved at algoritmen tillater senderen å overføre multiple pakker mens det ventes på en ack. Pakker sies å være unacknowledged hvis den har blitt overført, men ack er ikke mottatt. Teknisk sett betyr dette at antall pakker som kan vært unacknowledged ved en gitt tid, er begrenset av vindusstørrelsen og begrenset til et lite, fast tall.

Siden en vel innstilt sliding window protokoll holder nettverket fullstendig mettet med pakker, oppnår protokollen høyere ytelse enn en enkel positiv ack protokoll.

TCP bruker sliding window til å for å løse ende-til-ende flytkontroll-problemet, den har ikke en eksplisitt mekanisme for å congestion control.

Karn's algoritme: TCP skal ikke oppdatere round trip estimatet for retransmitterte segmenter. Når et round trip estimat regnes ut, skal sampler som tilsvarer retransmitterte segmenter ignoreres og i stedet skal det brukes en backoff strategi. Retransmisjon vil forverre congestion istedenfor å lette den. Hvis retransmisjonen ikke sjekkes, vil den økende trafikken produsere økende forsinkelse, noe som igjen fører til økende trafikk osv, helt til nettverket blir ubrukelig. Dette kalles congestion collapse, og TCP kan unngå dette fenomenet ved å redusere overføringsraten når congestion oppstår.

For å unngå congestion, kan man for eksempel benytte seg av en lett implementerbar teknikken *slow-start*. Slow-start funksjonen virker på følgende måte: Når det startes trafikk på en ny forbindelse eller trafikken øker etter en periode med congestion, skal congestion vinduet startes med en størrelse tilsvarende det enkelte segment, og øke vinduet med ett segment hver gang en ack ankommer. Altså: Vinduet settes til 1, når ack for denne pakken kommer, legger TCP til 1 til vinduet og sender to pakker. Når de to neste ack'ene ankommer, økes vinduet med 2, en for hver ack, og neste sender fire pakker osv.

For å unngå å øke vinduet for raskt og føre til økning i congestion, legger TCP til en tilleggsrestriksjon; Når congestion vinduet når sin originale størrelse før congestion, legges det til en congestion avoidance fase, som sakter ned økningsraten. Under congestion avoidance økes congestion vinduet med 1 kun hvis alle segmentene i vinduet har blitt ack'et.

Alt i alt forbedrer slow-start og congestion avoidance ytelsen til TCP dramatisk uten å måtte legge til signifikante overhead-beregninger.

TCP i trådløs link[6]

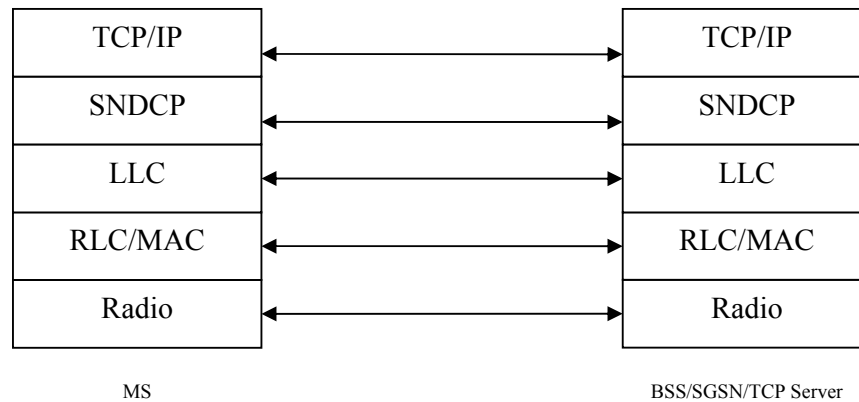
En av oppgaven har vært å undersøke TCP's ytelse og filoverføringsforsinkelse over en trådløs ende-til-ende forbindelse. TCP setter opp en ende til ende forbindelse i GPRS. TCP's ytelse i trådløse linker kan lett bli brutt ned i luftgrensesnittet på grunn av handoff, høye bitfeilrater og lang roundtrip delay.

De to hovedmåleenhetene for nettverk, er ytelse og forsinkelse. Vi ønsker selvsagt minst mulig forsinkelse og mest mulig ytelse. Det kan virke som at å øke ytelse vil redusere forsinkelsen, men det er ikke tilfelle.

TCP feiltolker trådløse tap eller forsinkelser som resultat av congestion, som igjen fører til redusert ytelse i form av TCP timeout og retransmisjon av pakker.

For GPRS og UMTS gjelder følgende: TCP pakken legges først til IP, SNDTCP og LLC-relaterte overhead og blir deretter sendt til RLC og MAC laget. RLC/MAC legger til sin egen overhead og videresender så pakken til fysisk lag.

En enkel modell, basert på figur 1 i [6], for protokollstakken til pakke-data-kommunikasjon er som Figur 7 viser.



Figur 7 – Signaleringslag

Ytelse forbedres ettersom meldinger blir større. Hver melding inkluderer en bestemt andel overhead, og en større melding betyr at overheaden er avskrevet over flere bytes.

TCP laget og RLC laget påvirker hverandre gjensidig[12]. At TCP avhenger av RLC, kan ha alvorlig innvirkning på TCP's ytelse. Dette medfører bruk av RLC ack mode der TCP brukes. I tillegg kan høye RLC blokkfeil-rater føre til signifikant forsinkelse i LLC, noe som igjen kan føre til at TCP congestion avoidance algoritmen utløses.

TCP's ytelse over GPRS

Artikkel [7] ble utgangspunktet for å undersøke TCP's ytelse og virkemåte analytisk på høyere lag. Den omhandler TCP's ytelse over toveis trafikk over asymmetriske linker. Effekten kalt *ack compression* reduserer den helhetlige ytelsen til TCP.

Segmenter på opplink og ack'er på nedlink deler en felles FIFO kø på IP laget. Køen betjenes av overføringsraten til opplink, og er nøkkelen til at ack compression inntreffer. For hvert datasegment som mottas, genereres umiddelbart en ack. Ack'er fra nedlink blir buntet sammen mens de venter i køen bak datapakken i opplink, og vice versa. GPRS har en asymmetrisk virkemåte, dvs kapasitet på opplink er lavere enn på nedlink. Vindusstørrelsen i opplink og nedlink må være store nok til å fylle round trip pipe for at ack compression skal kunne oppstå.

En felles prosess håndterer all TCP prosessering innen MS/BSS. TCP påkalles hver gang en ack ankommer MS, mottar datasegmenter fra nettverket, og leverer dem til den mottakene applikasjonen. Prosesseringstid i ende nodene kan overses, da denne er tilnærmet lik null i henhold til [7].

Prosessering av ack resulterer i at en ny datapakke legges til i TCP/IP køen for overføring. I tillegg, hvis MS er i slow start fasen, økes vindusstørrelsen med en, og en ekstra pakke legges til. (Denne sistnevnte pakken legges til i køen direkte bak den første i IP køen, og resultere i at to pakker blir overført uten en mellomliggende ack fra den motgående nedlinken). Datapakken legges til i køen, som svar på en burst av ack'er. Bursten ankommer den motsatte noden, og blir videresendt i en bunt uten noen mellomliggende kvitteringer.

Funksjonaliteten antatt for IP laget er enkel; for innkommende trafikk er IP laget ansvarlig for å videresende data fra lavere lag til den lokale TCP prosessen. Siden TCP's prosesserings tid i ende-nodene antas å være lav, kreves det ingen kø for innkommende trafikk på IP laget. For utgående trafikk kan det bli dannet en kø i IP laget. Køen avventer overføring på linken, som et resultat av at ack'er bunter seg sammen. I verste fall kan pakkeverdien til et helt vindu bli lagt til den utgående køen i rask rekkefølge etter hverandre, på grunn av at en bunt av ack'er ankommer fra den motsatte retningen.

Påliteligheten til TCP er basert på kvitterings- og retransmisjonsmekanismer. Hver oktett data som sendes får et sekvensnummer, og en ack på at pakken er mottatt kreves fra mottakene side. Dersom sender ikke mottar ack innen et gitt tidsintervall, retransmitteres data.

Det finnes tre ulike regimer for hvordan problemet med ack compression kan unngås:

- I. Gi prioritet til ack'er over datapakker
- II. Kontrollere ytelsen i begge retninger ved å benytte backpressure
- III. Benytte forbindelsesorientert båndbredde allokering

Å gi prioritet til ack'er i en kø, er en åpenbar løsning for å redusere ack compression. Prioritering av ack'er oppnås på bekostning av å ytelsen i opplink. I tillegg vil opplinken bare kunne bruke den delen av linken som ikke blir brukt til å overføre ack'er på nedlink. Nedlinken må justere sin ytelse til kapasiteten som er igjen på opplink etter å ha overført ack'er fra nedlink. I ekstreme tilfeller kan opplinken utelukkende bli brukt til å overføre ack'er og dermed mette linken helt. Effektiviteten til opplink er svært sensitiv til forholdet mellom datapakkers overføringstid over nedlink og kvitteringers overføringstid over opplink. Dersom man øker ack størrelsen, har det ingen betydelig innvirkning på nedlink, men ytelsen til opplink blir sterkt degradert. Ytelsen til nedlink forbedres, men dette kan føre til at ytelsen på opplink blir betraktelig dårligere.

Dersom man ønsker en løsning som kontrollerer ytelsen i begge retninger, kan man benytte *Backpressure* på TCP laget. Ved å bruke denne mekanismen, har man mulighet til å kontrollere ytelsen til forbindelsene i begge retninger ved å variere maksimalt antall datapakker i den utgående IP køen. Ytelsen kan kontrolleres ved at den overførende TCP forbindelsen i hver retning har tillatelse til å sende datapakker til IP laget så lenge inngangen til backpressure ikke overskrides. Når dette skjer, blir den overførende TCP forbindelsen avbrutt inntil det er tilgjengelig plass i IP køen. Backpressure kan forbedre effektiviteten til nedlink [7].

Forbindelsesorientert båndbredde allokering er en ideell løsning for å bukt med ack compression problemet, som vil tillate kontroll med ytelsen i både opplink og nedlink. Et slikt system skulle opprettholde betingelsen om at opplinken alltid er fullt utnyttet, men tillate kontroll med fraksjonene av datapakker og ack'er som overføres på opplink. Det grunnleggende målet er å gi en garantert minimumseffekt på opplink, mens den maksimale ytelsen til nedlink opprettholdes under disse betingelsene. Betingelsene oppfylles ved å begrense maksimalt antall ack'er som kan overføres før en datapakke overføres, når en eller flere datapakker venter i kø [7]. Dette kontrollerer tiden opplinken bruker for å overføre ack'er når datapakker venter, og vice versa. For å implementere dette, kan man benytte to køer; en for ack'er og en for data. Videre må en scheduler benyttes for å fordele hvilken av de to køene som skal overføres.

4.3.2 UDP

UDP er den andre av de to hovedprotokollene på transportlaget (lag 4) i internettarkitekturen [5]. Bruk av UDP istedenfor TCP som transportprotokoll, gir en mer transparent kategori av tjenester. I forhold til TCP som tilbyr en pålitelig bytestrøm kanal, er UDP en upålitelig, forbindelsesløs datagram-leveringskanal.

Protokollen garanterer ikke pålitelig og kontrollert levering av data. Dersom pakketap godtas, tillater RLC og LLC å operere i unacknowledged mode.

Som TCP, er UDP også en ende-til-ende-protokoll, men UDP tilbyr enkle multipleksingstjenester på toppen av IP. Essensielt tilbyr UDP innpakking og merking slik at data når riktig prosess. Men; UDP er ingen gunstig protokoll til overføring av data over GPRS og UMTS, da den er forbindelsesløs.

Dersom man benytter seg av NetCom Data Akselerator, opplever man ikke reduksjon i ytelse. NetCom Data Akselerator [14] er en klient-server basert programvare som gir en økt opplevd overføringshastighet for datatrafikk over mobilnettverket.

Programvaren utfører følgende operasjoner for å redusere nedlastingstiden:

- 1) Optimering av trafikken
- 2) Komprimering av innholdet
- 3) Mulighet for å fjerne uønsket filelementer

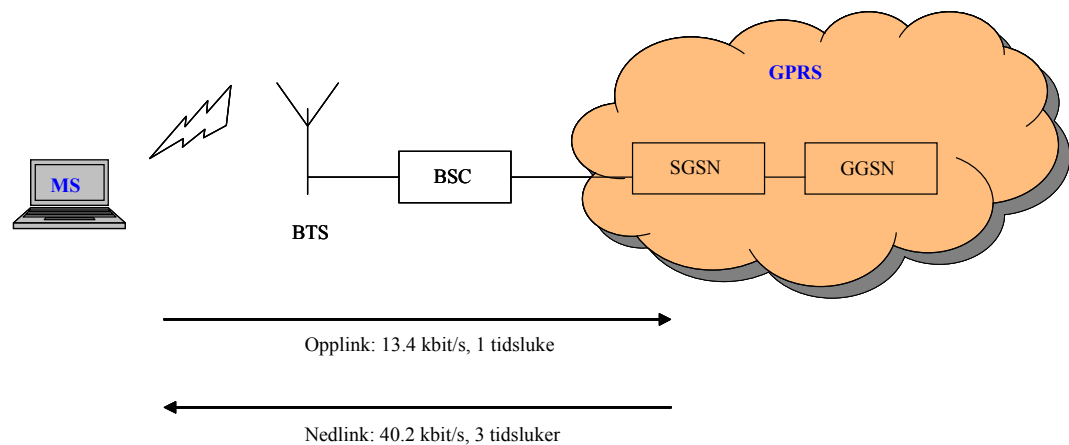
Her er TCP erstattet med UDP og en overliggende proprietær transportprotokoll.

5 Modellering av kommunikasjonssystemet

I dette kapitlet blir modelleringen av kommunikasjonssystemet beskrevet. I første del blir det gitt en beskrivelse av systemet. De påfølgende delene tar for seg modellering og matematisk utregning av verdier på hhv. Nettverkslaget og Transportlaget. Modelleringen på nettverkslaget seg på RLC og LLC protokollene, mens TCP blir omtalt matematisk på transportlaget.

5.1 Systembeskrivelse

Fokus er ytelsen i opplink og nedlink i form av overføringshastighet. Her ligger problemet i at ytelsen ikke er stabil i begge retninger. Et av målene har vært å undersøke hvorfor det er ustabil.



Figur 8 - GPRS arkitektur

Basiskonfigurasjonen er som vist i Figur 8 - GPRS arkitektur, og består av en mobil terminal som kommuniserer via BSS til SGSN/GGSN. Kommunikasjonen garanterer levering av pakker i rekkefølge og gir en situasjonsbestemt båndbredde for TCP forbindelsen mellom ende-nodene i hver retning. En tidsluke er reservert for opplink og tre tidsluker for nedlink.

I den grad det var mulig, var det ønskelig å lage en kvantitativ modell for å se på forsinkelse og ytelse i opplink og nedlink. For å løse dette på en tilfredsstillende måte, var det nødvendig å begynne fra bunnen av protokollstakken (jfr Figur 2), og finne ut hva som skjer på hvert signaleringslag (jfr Figur 7 – Signaleringslag). For at modelleringen ikke skulle bli for omfattende, ble det gjort abstraksjon på høyere nivå, siden disse lagene er meget komplekse. Utgangspunkt for modelleringen var å finne sannsynligheter for at forsinkelser vil oppstå. Det ble også lagt vekt på at modelleringen skal være mest mulig korrekt, så modelleringen startet med å se på hva som skjer på lag 2, med RLC og LLC;

1. Hva er sannsynligheten for retransmisjoner her?
2. Er det mye forsinkelse?

5.2 Matematisk modell for RLC og LLC

Utledningen av den matematiske modellen på lag 2 var et komplisert arbeid, da det finnes svært lite publisert materiell på området. Utgangspunktet var artikkelen "Link Layer Analysis of the General Packet Radio Service for GSM" [8]. I henhold til [8] er det enkelte punkter det er viktig å merke seg når det gjelder ytelsen til GPRS på opp- og nedlink;

Nedlinkens ytelsen er en viktig måleenhet fordi brukere, da spesielt mobilbrukere, i hovedsak vil bruke Internett til å produsere trafikklast på nedlinken. For å evaluere ytelsen i GPRS' nedlink, ble overføringsforsinkelsen til en LLC ramme analysert. Forsinkelsen i overføringen av LLC rammer kan brytes ned i mange tidskomponenter [8figur 1]. LLC rammen opplever overføringsforsinkelse for hver fysiske link den passerer.

Verdiene LLC opplever som forplantingsforsinkelse ved hver fysiske link, er så små at de kan overses. Sterkt avhengig av trafikksituasjonen, og også av den gjeldende radiolink-kvaliteten uttrykt i BLER (Block Erasure Rate), vil det være en kø av LLC rammer i BSC. I situasjoner med lav trafikk, vil en LLC ramme ha en prosesseringsforsinkelse i BSC og BTS. Denne forsinkelsen oppstår på grunn av avløsende funksjoner i BSC, og koding/interleaving i BTS.

Ved høy trafikk eksisterer ikke denne forsinkelsen. Årsaken er at relevante funksjoner kan bli utført i parallell med overføringen av LLC rammer som allerede venter i kø. Overføringsforsinkelsen av LLC rammer slutter når alle de tilhørende RLC blokkene har blitt korrekt mottatt av MS og *ikke* når RLC PDU i BSC har mottatt en positiv ack korrekt.

Følgende LLC rammestørrelser, inkludert LLC overhead på 6 oktetter, har blitt vurdert:

- 576 oktetter, som er standard IP størrelse, og
- 1506 oktetter, som er standard i LAN og maksimal størrelse for en LLC ramme

Rammestørrelsene gjelder dermed for $E[F_{Length}]$ i Formel 5.6.

Den første forsinkelsesfaktoren er en gjennomsnittsverdi for forsinkelsen, som presenteres av RLC lagets selektive ARQ mekanisme, $T_{RLCAckHandling}$, i Formel 5.1. Gjennomsnittsverdien er avhengig av to faktorer: BLER og størrelsen på en (partiell) LLC ramme uttrykt som en multipl av RLC blokker.

Block Erasure Rate, BLER, er sannsynligheten for at en RLC blokk må retransmitteres antatt at det ikke er samsvar mellom påfølgende RLC blokker. BLER kan tolkes som sannsynligheten for at dekoding av en RLC blokk feiler, altså prosentandelen av ukorrekte mottatte RLC blokker. ARQ protokollen gir retransmisjon av hver blokk som mottas feil. BLER er derfor også forholdet mellom antall retransmisjoner og antall overføringer totalt [12], og er avhengig av coding scheme og C/I. BLER til en *Temporary Block Flow*, TBF, kan finnes ved å registrere hvor mange RLC blokker som krever retransmisjon. Dette brukes som basis for å gjøre linktilpasninger og bytte fra et coding scheme til et annet. Teoretiske BLER-verdier er gitt i Tabell 3[8].

C/I [dB]	BLER	
	CS-1	CS-2
8	0.08	0.25
12	0.007	0.05
16	0.0002	0.0065

Tabell 3 – Teoretiske verdier

Formel 5.1 $T_{RLCAckHandling}(n) = T_{PartialLLC_TransDelay}(n) + T_{NackHandling}(n)$,

der n er rammestørrelsen uttrykt som en multiplum av RLC blokker. n ble satt til 49 bytes, som tilsvarer en LLC ramme med størrelse 100 bytes. Gjennomsnittsverdien for forsinkelse er avhengig av to faktorer; $T_{PartialLLC_TransDelay}(n)$ og $T_{NackHandling}(n)$. $T_{PartialLLC_TransDelay}(n)$, Formel 5.2, gir den kumulative retransmisjonsforsinkelsen til alle påfølgende partielle LLC rammetransmisjoner.

Formel 5.2 $T_{LLCFrames_TransDelay}(n) = (n \times (E[R] - 1) \times T_{RLCBlock_TransDelay})$,

der $T_{RLCBlock_TransDelay} = \frac{240}{13} ms$ og tilsvarer tiden det tar å sende 4 TDMA-rammer.

R er antall retransmisjonsforsøk per RLC blokk. Fordelingen til R er kjent som den modifiserte geometriske fordelingen med gjennomsnittsverdi og andre moment som vist i Tabell 4.

$$E[R] = \frac{1}{(1 - BLER)}$$

Formel 5.3

$$E[R^2] = \frac{1 + BLER}{(1 - BLER)^2}$$

Formel 5.4, $T_{NackHandling}(n)$, gir den kumulative tiden som trengs for å generere de negative ack'ene i MS, samt deres overførings- og prosesseringsforsinkelse i BSC. Tidsforsinkelsen inntreffer $E[M(n)]$ ganger per ramme (Formel 5.5).

Formel 5.4 $T_{NackHandling}(n) = (T_{MS_NackProc} + T_{Nack_TransDelay} + T_{BSC_NacProc}) \times E[M(n)]$,

der

- $T_{MS_NackProc}$ er tiden MS bruker på å svare på en negativ ack; 10ms
- $T_{Nack_TransDelay}$ er overføringstiden for en negativ ack og tilsvarer tiden for 2 TDMA rammer; 10ms
- $T_{BSC_NacProc}$ er tiden RLC PCU bruker i BSC for å forberede en retransmisjon; 10ms

Tabell 4 viser de kalkulerte verdiene for Formel 5.1, Formel 5.2 og Formel 5.4. $E[R]$ ble utregnet på bakgrunn av BLER-verdiene i Tabell 3.

E[R]		$T_{\text{PartialLLC_TransDelay}}(49)$	
CS-1	CS-2	CS-1	CS-2
1.09	1.33	0.0815	0.2991
1.007	1.053	0.0063	0.0480
1.0002	1.0065	0.0002	0.0059

Tabell 4 - Utregnede, teoretiske verdier

Forventningen $E[M(n)]$, Formel 5.5, gir gjennomsnittsverdien for antall negative ACK som sendes etter en LLC ramme, som initielt består av n RLC blokker.

$M(n)$ = "antall negative ACK som sendes etter en LLC ramme som initielt består av n RLC blokker"

Formel 5.5
$$M(n) = E[M(n)] = \sum_{m=1}^{\infty} m \times \Pr(M(n) = m)$$

$$\Pr(M(n) = 0) = (1 - BLER)^n$$

$$\Pr(M(n) = 1) = \sum_{k=1}^n \left(\binom{n}{k} BLER^k (1 - BLER)^{n-k} \right) \Pr(M(k) = 0)$$

$$\Pr(M(n) = m) = \sum_{k=1}^n \left(\binom{n}{k} BLER^k (1 - BLER)^{n-k} \right) \Pr(M(k) = (m - 1))$$

Forventningen forteller hvor lang tid en gitt ramme, f.eks. en IP pakke, bruker før den er ferdig med overføringen.

På bakgrunn av disse verdiene var det mulig å regne ut en gjennomsnittlig verdi for forsinkelsen på RLC laget.

Køforsinkelse i BSC

Køforsinkelsen er viktig fordi den endrer seg dynamisk med ankomstraten til LLC rammene ved BSC. Ankomstraten har en betydelig virkning på den helhetlige overføringsforsinkelsen av en LLC ramme mellom SGSN og MS. Derfor er det vekselvirkning mellom en LLC ramme med kort overføringsforsinkelse, og høy utnyttelse av nedlinken.

Så lenge det eksisterer en kø, vil LLC rammer som ankommer ikke oppleve prosesseringsforsinkelsen. Isteden vil de respektive RLC blokkene være kodet allerede ved det tidspunktet de kan bli overført.

Overføringsforsinkelsen for en LLC-ramme med lengden $E[F_{Length}]$ er gitt av Formel 5.6.

Formel 5.6 $T_{MeanFrame} = E[F_{Length}] \times T_{RLCBlock_TransDelay}$ ms,

der $E[F_{Length}]$ er en multipl av RLC-blokker som uttrykker gjennomsnittlig lengde av en LLC-ramme med MTU størrelse 576 og 1457 byte og receiver window størrelse lik 4. $T_{RLCBlock_TransDelay}$ er som nevnt i Formel 5.2.

Kalkulerte gjennomsnittstider for en ramme, $T_{MeanFrame}$, avhengig av pakkestørrelsen, er

- 39.58s, og
- 104.0s.

På bakgrunn av disse formlene var det mulig å regne ut gjennomsnittlig antall LLC-rammer, $E[N_{Frames}]$, dvs antall LLC-rammer som overføres og venter på videre overføring. Antall RLC blokker er, som nevnt i 4.1.2, konstant. $E[F_{Length}]$ har ovennevnte verdier.

Pollaczek-Khintchine formelen for M/G/1 køer brukes [13], og resulterer i Formel 5.7. Formel 5.7 er normalisert til $T_{MeanFrame}$, det vil si at ankomstraten λ er gitt i LLC-rammer per $T_{MeanFrame}$, og er en variable med verdier 0 – 1. Utregnede verdier for de ulike pakkestørrelsene, som ble brukt i denne oppgaven finnes i Tabell 4.

Formel 5.7
$$E[N_{Frames}] = \lambda E[R] + \frac{\lambda^2 E[R^2]}{2(1 - \lambda E[R])}$$

E [R ²]	
CS-1	CS-2
1.28	2.22
1.021	1.163
1.0006	1.0197

Ved å benytte Little's Result [13] får man gjennomsnittlig oppholdstid for en LLC-ramme i BSC, $E[T_{System}]$, gitt i

Formel 5.8.

Formel 5.8
$$E[T_{System}] = \frac{1}{\lambda} E[N_{Frame}] \times T_{MeanFrame}$$
 ms

Formel 5.8 ble programert i MatLab. Programkoden for utregningen av forventet tid for en LLC ramme er som vist under.

```
%gjennomsnittlig oppholdstid for en LLC ramme i BSC

function [svar]= hoved(lam,BLER)

T_block=(240/13)*10^-3;
F_length=40;
```

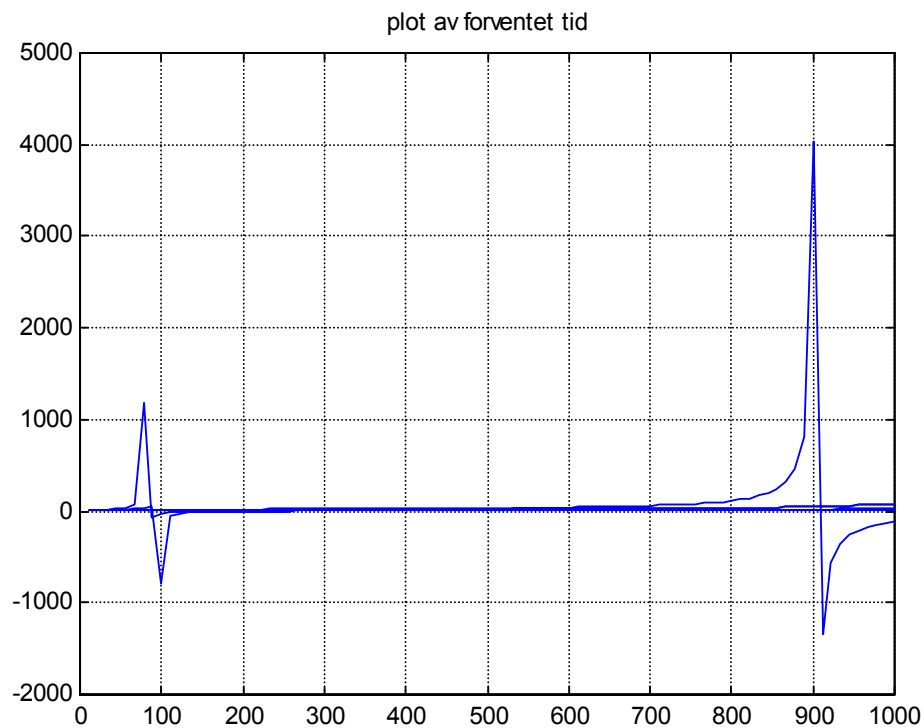
```

if lam/((1-BLER)*100)>=1
    svar=0;

else
    E_R=1/(1-BLER);
    E_R2=(1+BLER)/((1-BLER)^2);
    E_Sblock=E_R*T_block;
    T_meanframe=F_length*T_block;
    E_Sframe=F_length*E_Sblock;
    E_Nframes=(lam*E_R)+((lam^2*E_R2)/(2*(1-lam*E_R)));
    E_Tsystem=(1/lam)*E_Nframes*T_meanframe;
    svar=E_Tsystem;

end
    
```

Da jeg hadde problemer med matlab-programeringen, klarte jeg ikke å plote en tilfredsstillende graf her. I stedet plottet jeg en graf for en fast λ på 0.9 med rammestørrelse 576 og CS -2, som vist i ved å kjøre dette programmet i MatLab, får man grafen som vist i Figur 9 - Forventet oppholdstid for en LLC ramme i systemet nedenfor. I utgangspunktet har λ en verdi som varierer i område fra 0.1 – 0.9.



Figur 9 - Forventet oppholdstid for en LLC ramme i systemet

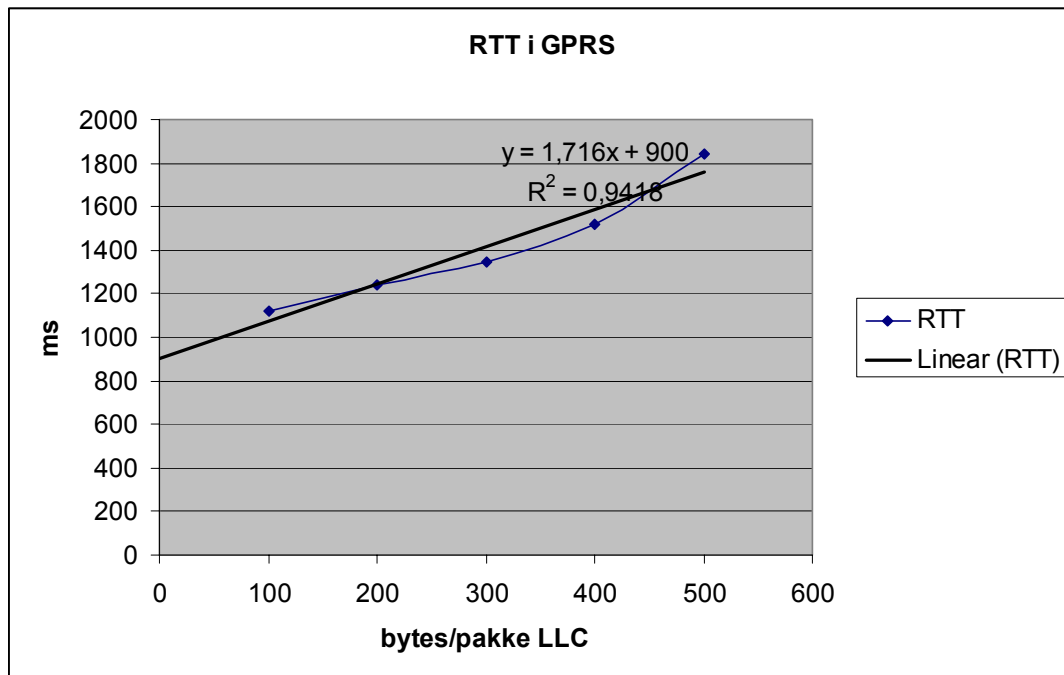
Da jeg fikk problemer med matlab-programeringen, klarte jeg ikke å plote en tilfredsstillende graf her. I stedet plottet jeg en graf for en fast λ på 0.9 og CS-2; BLER = 0,25 og 0.05, Flength = 576 bytes

Etter å ha regnet på ulike størrelser og verdier, var det mulig å svare på spørsmålene som ble stilt i innledningen av kapitlet:

1. Hva er sannsynlighetene for retransmisjoner?
Sannsynlighetene for retransmisjoner er avhengig av C/I og coding scheme, og ble regnet ut i formel 5.3, $E[R]$
2. Er det mye forsinkelse?
Den totale forsinkelsen ble regnet ut fra formel 5.8

RTT i GPRS

For å få målt opp tider i GPRS, ble det valgt å pinge ulike LLC størrelser et gitt antall ganger mot DNS. Å pinge er den høyeste man kan gå uten innblanding fra TCP. Ved å pinge går den ikke innom TCP med direkte fra IP til ICMP. Når man pinger sender man en LLC-ramme som går helt for seg selv. Round trip time, RTT, i GPRS ble funnet ved at LLC-rammer med ulike størrelser ble pinget 100 ganger. Resultatet av pingingen kan leses i Tabell 5. Ut fra denne tabellen ble det generert en graf. Siden det ikke er mulig å pinge med 0 pakker, ble det lagt til en trendlinje for å finne RTT's skjæringspunkt med y-aksen. Figur 10 viser tiden det tar å sende ulike LLC-rammer med ulike størrelser.



Figur 10 - RTT for GPRS

LLC size	RTT
100	1121
200	1244
300	1348
400	1520
500	1841

Tabell 5 - RTT avhengig av LLC størrelse

Skjæringspunktet med y-aksen er 900ms. 900ms ser ut til å være summen av alle konstante ledd for både opp- og nedlink.

5.3 Matematisk modell for TCP

Oppgaven med å undersøke RLC og LLC's forsinkelser matematisk var utført. Artikkel [7] var utgangspunktet for å undersøke TCP's virkemåte og ytelse analytisk på høyere lag. Den omhandler TCP's ytelse over toveis trafikk over asymmetriske linker. Effekten kalt *ack compression* reduserer den helhetlige ytelsen til TCP. Segmenter på opplink og ack'er på nedlink deler en felles FIFO kø på IP laget. Køen betjenes av overføringsraten til opplink, og er nøkkelen til at ack compression inntreffer. For hvert datasegment som mottas, genereres umiddelbart en ack. Ack'er fra nedlink blir buntet sammen mens de venter i køen bak datapakke i opplink, og vice versa. TCP setter opp en ende til ende forbindelse i GPRS. Samtidig har GPRS en asymmetrisk virkemåte, dvs lavere kapasitet på opplink enn på nedlink. Vindusstørrelsen i opplink og nedlink må være store nok til å fylle round trip pipe for at ack compression skal kunne oppstå. W_u og W_d er vindusstørrelsen til opplink og nedlink gitt i byte. Vindusstørrelsen er avhengig av IP MTU størrelse. Default receiver window er 4. De ulike parameterne ble regnet ut for de to valgte MTU størrelsene, og coding schemes, og er listet opp i Tabell 6 og Tabell 7. Betingelsen i Formel 5.9 antas å alltid være oppfylt, jfr. (2.1) i [7].

For å finne forsinkelsen fra MS til SGSN og fra SGSN til MS, $D_{MS_SGSN} + D_{SGSN_MS}$, ble en pakke tilsvarende 100 byte LLC pinget 1000 ganger mot DNS(?). Gjennomsnittstiden var 1133 ms. Denne verdien ble benyttet for $D_{MS_SGSN} + D_{SGSN_MS}$ i utregningene.

Parametrene fra Tabell 5 og Tabell 6 ble satt inn og regnet ut for Formel 5.9. Det viste seg at betingelsene var oppfylt for såvel de ulike MTU størrelsene, som for valg av coding scheme.

Formel 5.9
$$\frac{W_u}{\rho_u} + \frac{W_d}{\rho_d} > D_{MS_SGSN} + D_{SGSN_MS}$$

MTU = 576 byte W = 2144 byte			
CS-1		CS-2	
ρ_u	9.05 kbit/s	ρ_u	27.15 kbit/s
ρ_d	13.4 kbit/s	ρ_d	40.2 kbit/s
W_u/ρ_u	1.895	W_u/ρ_u	1.280
W_d/ρ_d	0.632	W_d/ρ_d	0.427

Tabell 6 - Parametre for MTU = 576 byte

MTU = 1457 byte W = 5668 byte			
CS-1		CS-2	
ρ_u	9.05 kbit/s	ρ_u	27.15 kbit/s
ρ_d	13.4 kbit/s	ρ_d	40.2 kbit/s
W_u/ρ_u	5.010	W_u/ρ_u	3.384
W_d/ρ_d	1.670	W_d/ρ_d	1.128

Tabell 7 - Parametre for MTU = 1457

For å finne hvor godt linken utnyttes, dvs TCP's ytelse, ble det tatt utgangspunkt Teorem 1 i [7]. Teoremet forteller at "gjennomstrømmingseffekten for opplink i en toveis konfigurasjon over asymmetriske linker er gitt av F_u ". F_u estimerer forholdet mellom ytelsen i en link og kapasiteten til den korresponderende link under ack compresssion. Dette betyr at F_u estimerer linkens ytelse, og er gitt av Formel 5.10. Utregningene er basert på parameterne i Tabell 6 og Tabell 7.

Formel 5.10

$$F_u = \begin{cases} 1, & \text{if } (W_u/\rho_u) > W_d/\rho_d + (D_{MS_SGSN} + D_{SGSN_MS}) \\ \frac{2(W_u/\rho_u)}{(W_u/\rho_u + W_d/\rho_d) + (D_{MS_SGSN} + D_{SGSN_MS})}, & \text{if } W_d/\rho_d - (D_{MS_SGSN} + D_{SGSN_MS}) \leq W_u/\rho_u \leq W_d/\rho_d + (D_{MS_SGSN} + D_{SGSN_MS}) \\ \frac{(W_u/\rho_u)}{(W_u/\rho_u)}, & \text{otherwise} \end{cases}$$

Betingelsene for hver av de ulike delene av funksjonen ble regnet ut. Resultatet av utregningen ble som vist i Tabell 8.

	MTU= 576		MTU= 1457	
	CS-1	CS-2	CS-1	CS-2
$(W_u/\rho_u) > W_d/\rho_d + (D_{MS_SGSN} + D_{SGSN_MS})$	TRUE	FALSE	TRUE	TRUE
$W_d/\rho_d - (D_{MS_SGSN} + D_{SGSN_MS}) \leq W_u/\rho_u \leq W_d/\rho_d + (D_{MS_SGSN} + D_{SGSN_MS})$	FALSE	TRUE	FALSE	FALSE

Tabell 8- Betingelser for hvilke deler av F_u som gjelder

Tolkning av resultater:

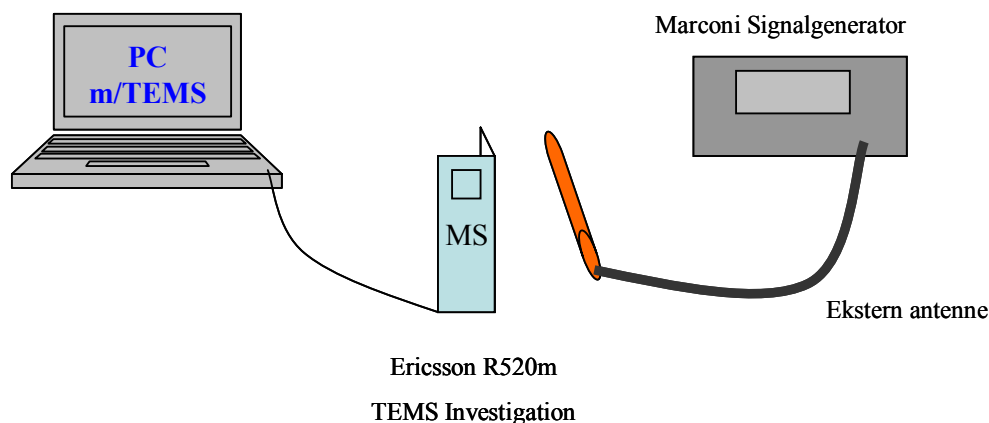
Fu estimerer linkens ytelse. Dersom forholdet er 1, betyr det at linken utnyttes 100%
Med MTU=576 er effektiviteten lik 1 dersom CS-1 benyttes, dvs opplinken utnyttes 100%, og ack compression fører til at linken blir full.

For CS-2, derimot, er ytelsen 0.901, noe som betyr at linken utnyttes med 90.1%. CS-2 gir bedre ytelse i de fleste tilfeller fordi den har høyere bitrate, som fører til flere retransmisjoner.

Med MTU=1457 er effektiviteten lik 1 både for CS-1 og CS-2. Dette kan tolkes dithen at opplinken er 100% full

6 Eksperimentell verifisering av resultater

De kalkulerte verdiene ble eksperimentelt verifisert. Oppsettet for verifisering var som vist i Figur 11 – Laboratorieoppsett nedenfor. En bærbar pc ble tilkoblet GPRS via mobiltelefonen, en Ericsson R520m med TEMS™ Investigation, og en TCP forbindelse for å laste ned og laste opp i kontrollert miljø ble dermed etablert. Programmene TEMS, CommView og ftp Explorer ble startet. Målingene ble foretatt under kontrollerte forhold, og frekvens- hoppingen i nettet var avslått. For å skape trafikk på linken, ble FTP Explorer brukt for å laste et dokument i nedlink og samtidig sende en fil i opplink. Trafikken ga kø i begge retninger. C/I –verdiene ble regulert av en ekstern antenne tilkoblet en signalgenerator. Den eksterne antennen ble lagt ved siden av mobiltelefonen, for å skape støy på mobilens antenne. Normalt vil man befinne seg i stabilt område, men fenomenene med ulike C/I-verdier er interessante å studere på grunn av slow start og congestion, for da stemmer ikke forsinkelsene mer. I stabil tilstand er det lite eller ingen feil, og alt går knirkefritt. Støyen ble lagt på for å gjøre signalkvaliteten dårligere. Konsekvensene var at ytelsen ble svekket i hvert tilfelle. Området har vært viktig å undersøke fordi man ikke vet hvordan RLC og LLC oppfører seg over GPRS når det er dårlig C/I.



Figur 11 – Laboratorieoppsett

TEMS ga mulighet til å låse mobiltelefonen fast til en celle. I tilfelle med MTU størrelse på 1457 oktetter, var kanalnummer 90 den aktuelle. For å finne riktig bærefrekvens, ble Formel 6.1 benyttet

Formel 6.1 $F(n) = 890 + 45 + 0.2 \times n$, der n er kanalnummer og svaret er gitt i MHz

$$F(90) = 890 + 45 + 0.2 \times 90 = 953 \text{ MHz}$$

Tabell 9 gir en kortfattet oversikt over de ulike enhetene i laboratorieoppsettet.

Utstyr/Programvare	Beskrivelse
Ericsson R520m med TEMS™ Investigation	
TEMS™ Investigation GSM 3.2	Programvare: testverktøy for luftgrensesnitt over GSM og GPRS nettverk. Programmet gir mulighet til å monitorere såvel talekanaler som dataoverføring over GPRS. Oppførselen til datasesjoner kan studeres. Data presenteres i real time ytelse. Alle data kan dessuten lagres i loggfiler for etterarbeid. http://www.ericsson.com/tems
CommView Version 3.3, Evaluation Version	Program som lister opp nettverksforbindelser og nødvendig IP statistikk, og som gjennomgår individuelle pakker. Pakkene dekodes ned til det laveste laget med full analyse av de mest utbredte protokollene, og de kan logges for videre analyse. Et fleksibelt system gjør det mulig å droppe de pakkene man ikke ønsker eller man kan selv velge hvilke man ønsker å se på. http://www.tamos.com
Rob Vonk's EasyMTU 3.0, Internet connection speedup toolkit	Program som gjør det mulig å sette IP MTU til den størrelsen man ønsker. EasyMTU søker etter TCP/IP protokollen som er knyttet til din dial-up adapter og endrer kun disse innstillingene slik at andre elementer i registeret forblir uendret. http://members.tripod.com/~EasyMTU/
FTP explorer Version 1.00	ftp applikasjon http://www.ftpx.com
Marconi Instrument 10kHz – 1GHz Signalgenerator 2022D	Signalgenerator
KATHREIN Typ Nr. K751167 f/MHz 806 - 960	Ekstern antenne

Tabell 9 – Målingselementer

TEMS ble brukt til å se på RLC og LLC's oppførsel; bl.a. ytelse i opp- og nedlink, decode errors, og anvendelse av CS-1 og CS-2. Med TEMS var det mulig å se på kurver, som viste dataflyten i opplink og nedlink, i sanntid.

Programmet comm view for å se på TCP/IP; antall pakker, vindusstørrelse, opp og nedlink. Programmet EasyMTU ble brukt for å sette MTU-størrelsen til ønskede verdier.

Både TEMS og CommView ble brukt til å logge henholdsvis RLC/LLC- og TCP/IP-logger for de ulike C/I-veridene på 8dB, 10dB og 12dB. Disse C/I-verdiene gjenspeiler akseptable C/I-områder. TEMS-loggene ble eksportert til tekstfiler fra TEMS og åpnet i Excel. Ut fra disse filene var det mulig å finne ulike verdier for RLC og LLC.

I de kommende underkapitlene er blir ulike fenomener som inntreffer avhengig av C/I, vist med figurer fra loggfilene i TEMS. For alle figurene gjelder at den øverste kurven representerer nedlink og den nederste opplink i de respektive vinduene.

6.1 Måling over GPRS med MTU = 1457 oktetter

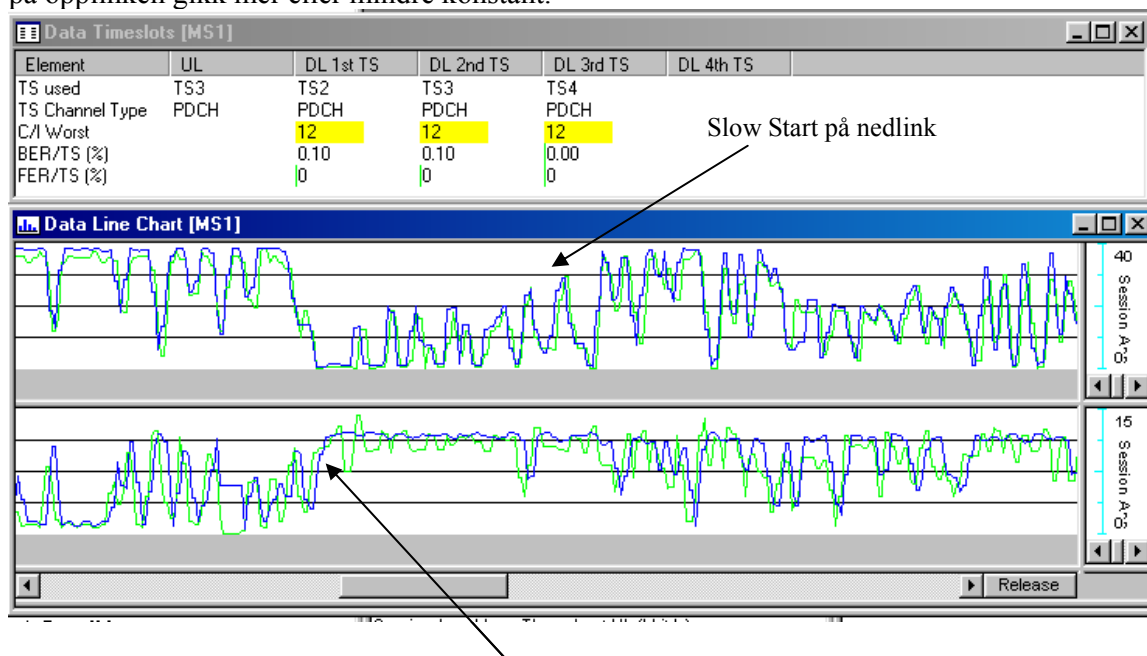
MTU = 1457 ble valg fordi den tilsvarer den største rammen LLC kan ha før den blir segmentert. Bærefrekvensen, utregnet på bakgrunn av Formel 6.1, var på 953.00MHz, modulert med $f^M 20.0\text{kHz}$. Støy ble lagt på ved å variere RF nivået i området 282 - 545 μV , slik at C/I holdt seg mer eller mindre konstant på en valgt verdiene. Årsaken til at støy ble lagt på under målingene var at det i stabil tilstand er lite eller ingen feil, og alt går knirkefritt.

Opplinkbufferet holder seg lite med mindre det blir feil der. Dersom det blir feil, oppstår forsinkelse, jfr problemet med at ack'er bunter seg sammen i køer, som beskrevet i kapittel 4.1.2. Forbindelsen det ble målt på er feilfri, men forsinkelsen varierer.

Ut fra loggfilene ble det hentet ut verdier for LLC ytelse i opplink og nedlink.

6.1.1 Observasjoner ved målinger med $C/I = 12$ dB

Overføringen av data var forholdsvis jevn mellom opplink og nedlink. Å laste ned data tok kortere tid enn å sende data. På Figur 12 nedenfor blir et utsnitt av overføringen vist, der man kan se et eksempel på at slow-start inntraff da overføringen på opplinken gikk mer eller mindre konstant.



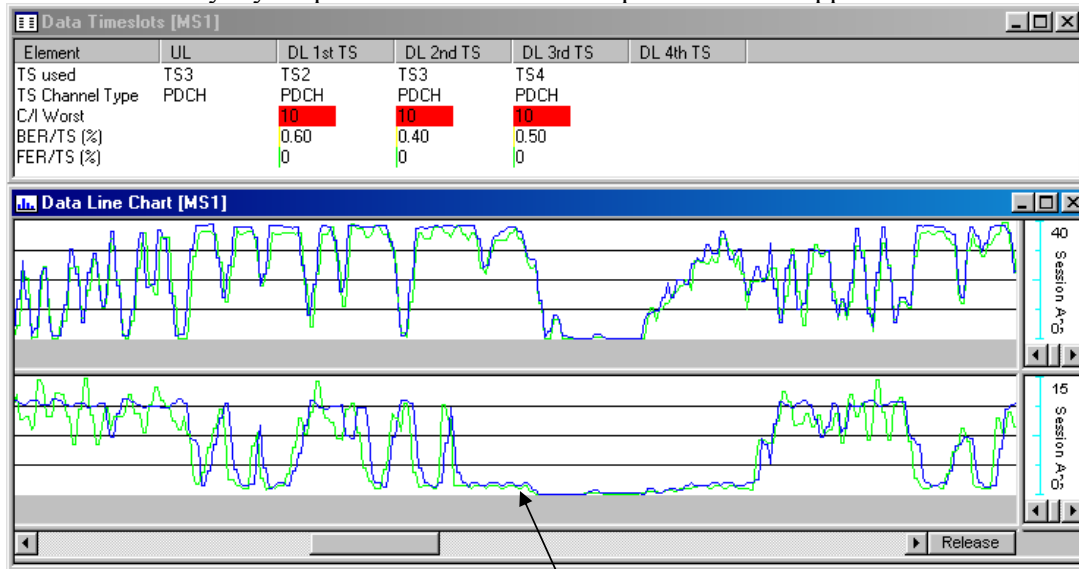
Konstant overføring på opplink

Figur 12 – Utsnitt fra loggfil med $C/I = 12$ dB

En C/I verdi på 12 dB er en "god" verdi. Mobiltelefonen mestrer å ha trafikk i begge retninger. Selvom det, som nevnt, tok noe lenger tid å sende data på opplink enn det tok å laste ned på den raskere nedlinken.

6.1.2 Observasjoner ved målinger med C/I = 10 dB

De samme dokumentene ble lastet i opplink og i nedlink, men denne gangen med C/I = 10 dB. Dårligere C/I resulterte i at det tok dobbelt så lang tid å sende data i forhold til å laste ned. Mye tyder på at bufferet var fullt en periode slik at opplinken timet ut.



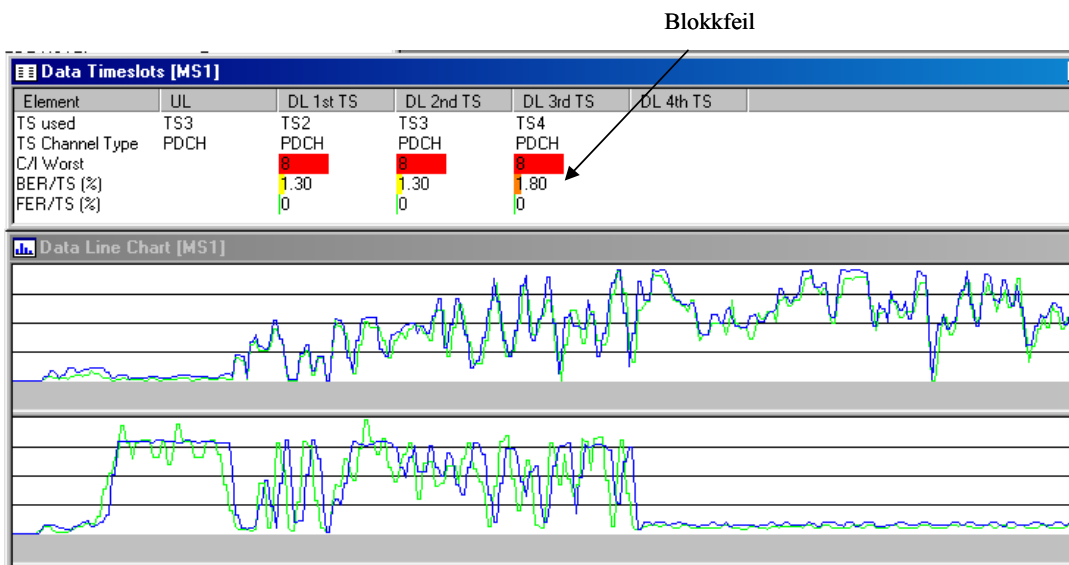
Timeout-periode

Figur 13 - Utsnitt fra loggfil med C/I = 10 dB

Det var også en del 'fjelltopper' i overføringen. De små toppene må være RLC retransmisjoner. Figur 13 viser et utsnitt fra loggfilen, der opplink timer ut en periode. Dette betyr at bufferet er fullt. Fra figuren kan man også observere at BER er større enn for C/I=12dB.

6.1.3 Observasjoner ved målinger med $C/I = 8$ dB

Med $C/I = 8$ dB tar det ikke lang tid før opplinken timer ut. Konsentrasjonen av 'fjelltopper' er høy, og i tillegg er det mye blokk feil, BER, dette kan ses i Figur 14 - $C/I = 8$ dB Opplinken timet ut i lange perioder, og hadde store problemer med å fortsette med opplasting av filen. Opplinken klarte ikke å sende og blir stående uten å laste noe som helst i en periode. BER er betydelig høyere enn i de foregående målingene, og fra figuren kan man også tydelig se starten på en lang timeout periode. Etter denne perioden, kan slow start tydelig ses:



Figur 14 - $C/I = 8$ dB

Overføringen gikk tregt, og da nedlink har nådd 79% og opplink 40%, stoppet overføringen helt opp. Siden det ikke skjedde noe, ble måleutstyret sjekket, og det viste seg at telefonen hadde slått seg av.

BCCH blir dårlig ved lave C/I verdier (fra 8dB og lavere). MS vil ikke kunne aksessere nettet lenger fordi broadcast kanalen blir uleselig.

Dersom TCP venter, vil ikke LLC rammer umiddelbart etterfølge hverandre, og TBF'en kobles ned. Problemet på 8dB var at GPRS ikke klarte å sette opp ressurser.

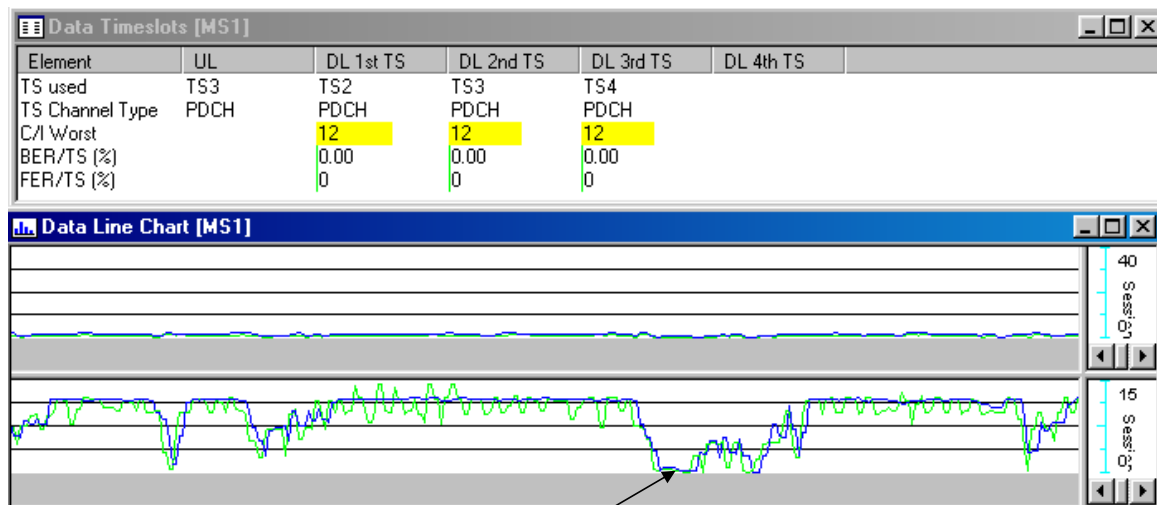
6.2 Målinger over GPRS med MTU=576 oktetter

For MTU = 576, som tilsvarer typisk verdi på IP pakke, ble de samme målingene som i 6.1 gjort. C/I verdiene var 8, 10 og 12 dB. Bærefrekvensen ble utregnet på bakgrunn av Formel 6.1, og var på 958.8MHz, modulert med $^{FM}20.0\text{kHz}$. Støy ble lagt på ved å variere RF nivået på samme måte som for målingene med større MTU.

I de kommende underkapitlene blir ulike fenomener som inntreffer avhengig av C/I, vist med figurer fra loggfilene i TEMS.

6.2.1 Observasjoner ved målinger med C/I = 12 dB

Det ble lastet og sendt en fil i opplink og nedlink. Overføringen i nedlink gikk raskt i forhold til opplastingen i opplink. Også her ble timeout og slow start observert:

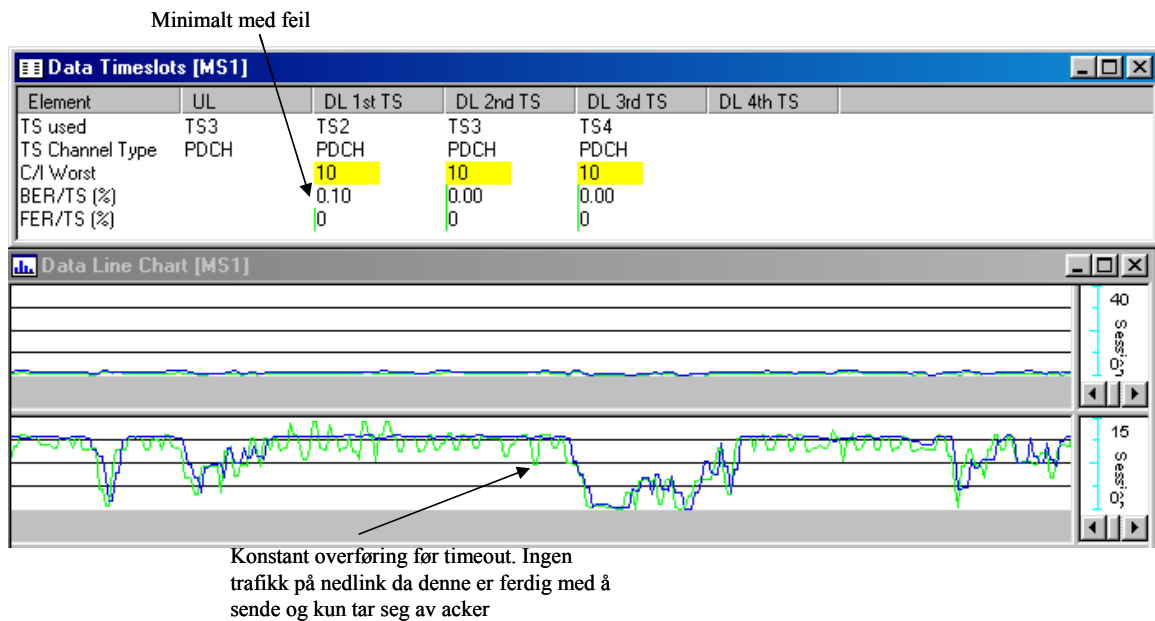


Nedlinken blir raskere ferdig enn opplink. Allikevel oppstår timeout etter en periode med konstant overføring. Slow start kan tydelig ses.

Figur 15 - C/I= 12 dB

6.2.2 Observasjoner ved målinger med C/I = 10 dB

Samme fil som for 12 dB ble lastet og sendt. Det er lite blokkfeil å observere, og, som for MTU=1457, tilsvarer de små toppene sannsynligvis RLC-retransmisjoner:

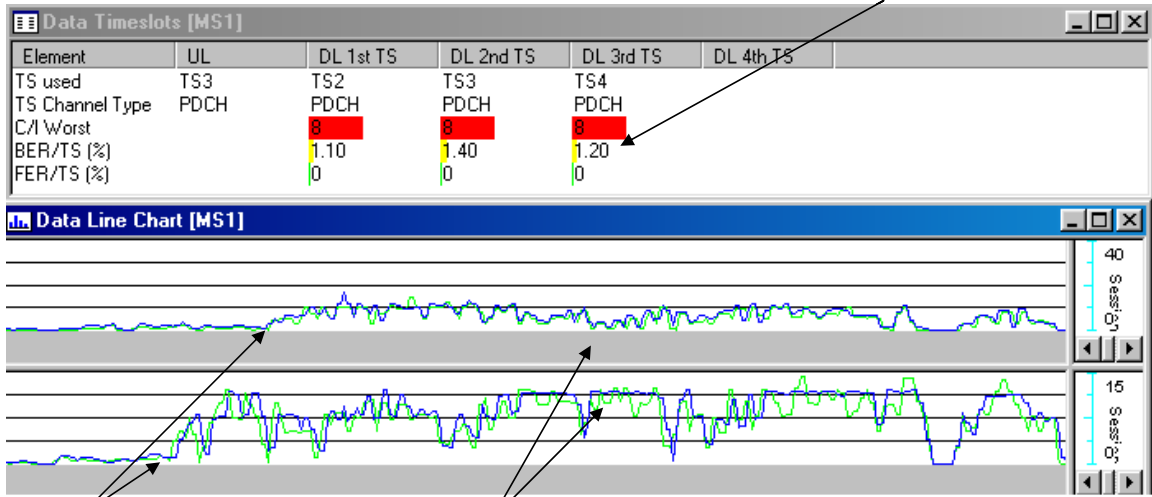


Figur 16 - C/I=10dB

6.2.3 Observasjoner ved målinger med C/I = 8 dB

Filene ble lastet ned og sendt i hver sin retning. Sammenlignet med observasjoner gjort for større MTU, ble det observert fra kurvene i opplink og nedlink at nivået i opplink var lavere enn for større MTU. Med C/I = 8 dB, var blokkfeilraten høyere enn for de foregående C/I-verdiene:

Bitfeilene er omtrent på samme nivå som for MTU=1457



Slow start

Sammenhengen mellom slow start og konstant overføring: nedlink er i slowstart-fase mens opplink har konstant nivå.

Figur 17 - C/I = 8dB

6.3 Kommentar til målinger

Generelt sett tar det dobbelt så lang tid å laste opp(sende) et dokument enn å motta et når $MTU = 576$ bytes.

For hver alle de foretatte målingene oppstår timeout og slow start.

Med $C/I = 8\text{dB}$ og lavere, ble BCCH også dårlig, slik at MS ikke kan aksessere nettet lenger. Når LLC rammen er slutt, forsvinner TBF'en og bufferet blir tomt. Ny TBF blir satt opp via felles signaleringskanal, som er forstyrret med støy. Det samme inntreffer når forsinkelsen har blitt så lang at TBF'en ikke orker å holde på mer. Dette resulterer i timeout. Under slow start er vinduet lite, og kan dermed time ut, men det antas at vinduene/pakkene kun blir forsinket og ikke mistes fysisk. Det er kun en kø; en TCP kø. TBF'en tar hensyn til at pakken er mindre på opplink enn på nedlink. Sannsynligvis skyldes de ustabile forholdene prosesseringsforsinkelse.

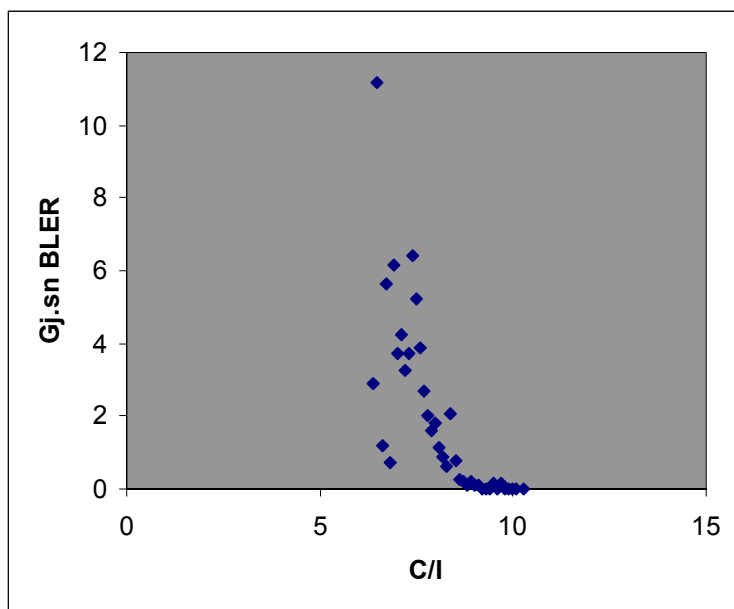
Det er et faktum at webtrafikk går dårligere i situasjoner med retransmisjon av IP pakker. RLC i GPRS har innebygd at den prioriterer ack'er(kapittel 4.1.2). Det later til at denne mekanismen ikke er implementert i versjonen av TCP som finnes i nettet, siden linken timer ut. RLC gir bitfeilraten. Eventuelle rammefeil kommer fra LLC. Normalt tilfredsstillende linken kriteriene gitt i F_u , Formel 5.10.

6.4 Eksperimentelt resultat av BLER som funksjon av C/I

Tre signifikante C/I verdier ble valgt; 8dB, 10dB og 12dB. Resultatene fra målingene vises i figuren nedenfor. Sammenlignet med tabell 2 i [8] ser vi at resultatene avviker noe. Dette skyldes 'de store talls lov'; dersom man skal oppnå målte verdier tilnærmet lik teoretiske, vil man måtte foreta en stor mengde målinger over lengre tid. På grunn av at utledningen av de matematiske formlene tok lenger tid enn antatt, var det ingen mulighet for å rekke dette. CS-2 er mer sensitiv til radioforhold enn CS-1. Dette forklarer at utnyttelsesgraden er høyere for gjennomsnittlig $C/I = 12\text{ dB}$ enn for 8 dB, CS-1 har bedre utnyttelsesgrad enn CS-2, og forskjellen mellom utnyttelsesprosentene til CS-1 og CS-2 er mye større for $C/I = 8\text{ dB}$ enn for 12 dB. CS-1 utnytter systemets tilgjengelige kapasitet bedre enn CS-2. Dersom CS-2 benyttes istedenfor CS-1, er ende-til-ende ytelsen mye mer sensitiv med hensyn på radiolink kvaliteten.

For små pakker benyttes kun CS-1, da den ikke rekker å tune seg inn til noe annet.

Ut fra loggene har gjennomsnittlig BLER som funksjon av C/I blitt generert, og kan ses i Figur 18.



Figur 18 - BLER som funksjon av C/I

Sammenlignet med grafen i [8] stemmer den ikke helt. Dette kan skyldes :

- teoretiske verdier antar mobilitet, og at
- radiomiljøet i laboratorium er konstante.

6.5 Simulering av web-trafikk

Http-trafikk er vanskelig å standardisere og modellerer. Trafikk over web/http blir aldri stabilt fordi det går med mye tid til å finne 'ting og tang'. Dersom det er store pakker, blir det mye 'fram og tilbake', og dette har ikke noe med GPRS å gjøre. Ytelsen skal være den samme.

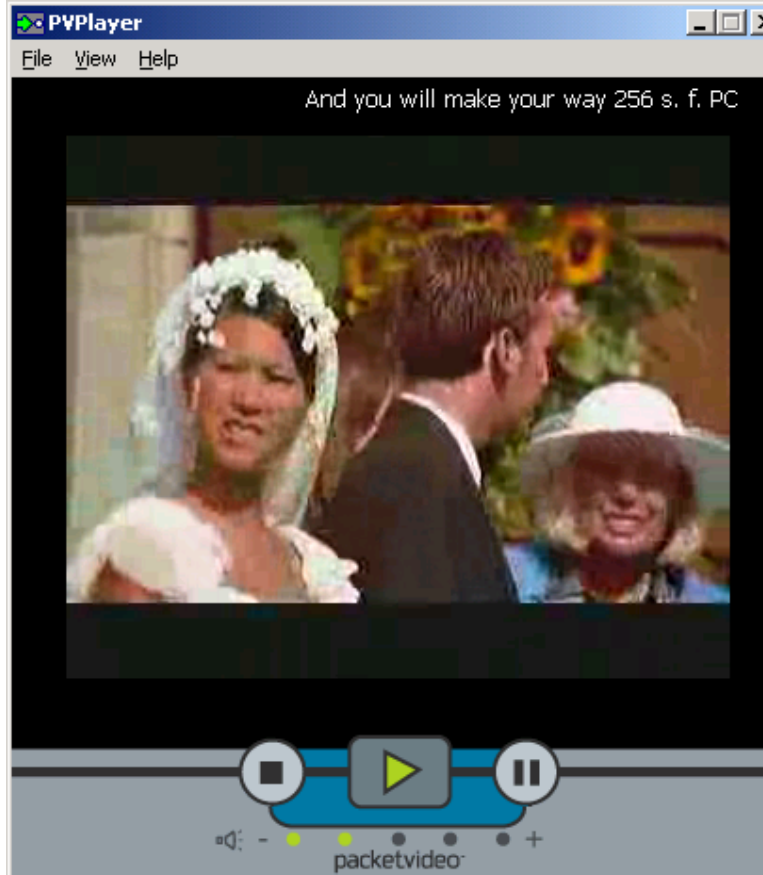
Metoden som ble brukt for å gjøre dette var som følger: En fil med kjent størrelse, 80kB, ble lastet ned, og så hvor lang tids dette tok. Deretter ble den samme filen lastet, men denne gangen ble http-trafikk simulert i opplink ved å pinge dns i evig løkke med en størrelse på 100byte. Filen ble lastet ned to ganger til, med hhv. 2 og 3 ping-vinduer. Ved å pinge på denne måten simulerer man blant annet get og dns-oppslag. Da det kun var trafikk i en retning, gikk nedlastingen greit, men med en gang det ble lagt på simulert http-trafikk ble færre bytes/s overført. De observerte resultatene kan ses i Tabell 10

	Bytes	Tid(s)	Bytes/sek
Ingen ping	81920	77	1064
Ping: LLC = 100byte i evig løkke	81920	97	845
2xPing: LLC = 100byte i evig løkke	81920	110	745
3xPing: LLC = 100byte i evig løkke	81920	133	616

Tabell 10 - Ytelse ved simulert http-trafikk

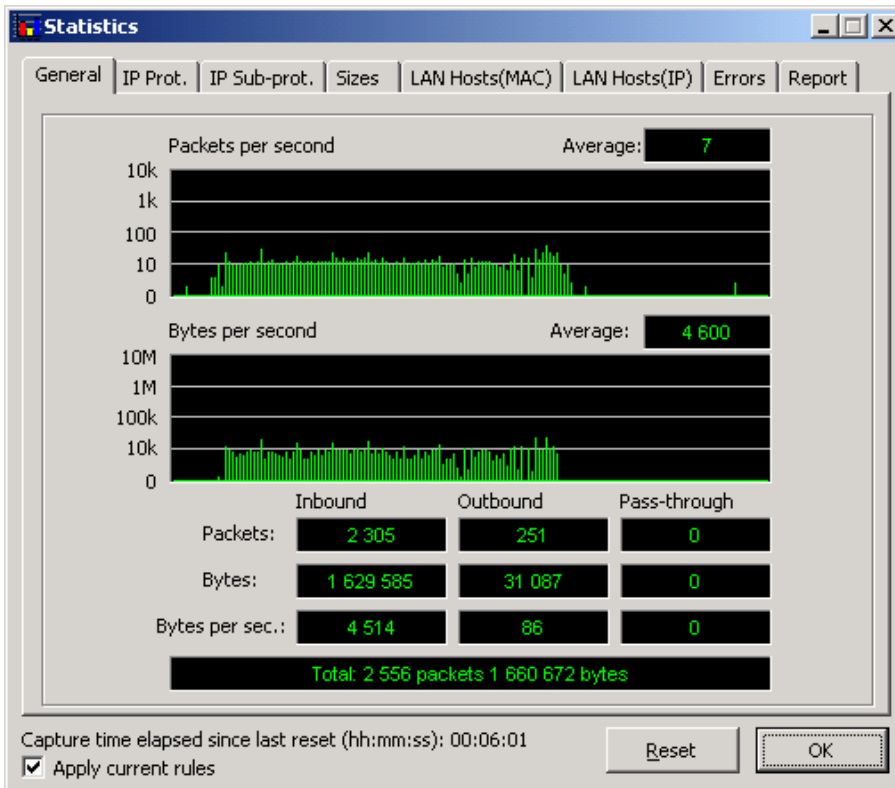
7 UMTS

Dessverre ble det ikke gjort mye i UMTS, selv der det var mulig å oppnå resultater. For eksempel kunne det vært mulig å se på TCP/IP over UMTS. Årsaken til at dette ikke ble gjennomført var for dårlige måleinstrumenter og forsinket software release fra NetCom's leverandører. I tillegg var det tidsmessig for sent å begynne med dette. Det ble utført et lite eksperiment der UMTS-telefonen ble tilkoblelet NetCom's UMTS testnett og det ble streamet en video. Figur 19 er et 'screen-shot' fra streamingen som foregikk over RTSP, Real-Time Streaming Protocol, over UDP.



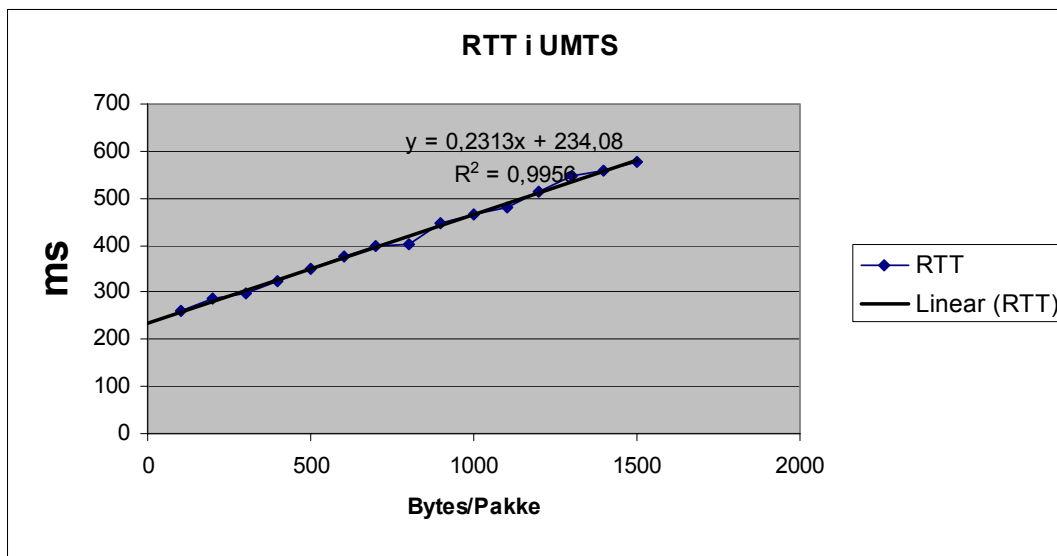
Figur 19 – RTSP, real-time streaming protocol over UDP over UMTS

CommView ble brukt til å se på statistikk over antall pakker og antall bytes per sekund under streamingen. Figur 20



Figur 20 - Statistikk for UMTS

Det ble foretatt en ping test over UMTS for å finne RTT. Fra de målte resultatene ble det generert en graf, Figur 21. Grafens skjæringspunkt med y-aksen er 234,08 sekunder.



Figur 21 - RTT i UMTS

8 Diskusjon

Resultatene etter å ha studert RLC, LLC og TCP analytisk i forhold til det som ble sett på målinger, viste seg å bero på andre forhold enn det som var forventet. Det viser seg at flytkontrollen på TCP nivå har mye større innvirkning enn antatt. Den antatt store innvirkningen i variasjonen i overføringstiden for LLC rammer, ser ut til å ha liten innvirkning på sannsynligheten for oppførselen til flytkontrollmekanismene i TCP. TCP håndterer køen og implementerer flytkontroll per LLC ramme.

Enveis TCP-sesjoner gir ytelse nær det maksimale for aksessforbindelse. Men; dersom det pågår toveis trafikk, som vil være vanlig ved web-applikasjoner, reduseres ytelsen til TCP betraktelig gjennom langsom slow start- og congestion recovery-mekanismer. Det ble utledet at GPRS kan tillate operasjon av asymmetrisk TCP-forbindelse i stabilt regime med høy utnyttelse, slik eksperimentene også viser. Det viste seg under utførelsen av eksperimentene at bl.a. direkte innsyn i bufferstørrelser i TCP/IP ville vært nødvendig for å eksplisitt kunne forklare og kvantifisere degradering i ytelse, som følge av slow start og fast recovery.

Det er vist at en typisk radioforbindelse for asymmetrisk web-aksess gjennom UMTS (384/64 kbit/s) også vil kunne operere ved stabilt høy ytelse, men mangel på korrelerbare observasjoner tillot ikke at dette ble nærmere undersøkt. Imidlertid kan det antas at med kortere RTT, vil effekten av både slow start og congestion recovery subjektivt oppfattes som mindre degraderinger av brukere av web.

Andre protokoller enn TCP er tilgjengelig for streaming-tjenester (f. eks RTSP). Disse vil operere over UDP og ha andre flytkontroll-egenskaper enn TCP. Også "akselleratorer", som tilbys av nettoperatorene vha proxyer og proprietære protokoller for installasjon på klientene, vil kunne avhjelpe problemer som er undersøkt i denne oppgaven.

Utviklingen av mobilkommunikasjon går i retning av å integrere mekanismer for tjenestekvalitet i infrastrukturen. Disse vil gjøre pakke-data-forbindelser mer pålitelige mhp forsinkelse og kapasitet, og dermed avhjelpe noen av fenomenene som reduserer ytelsen for TCP/IP. Imidlertid har vi sett at flytkontrollen i TCP er dårlig egnet for "lange smale" forbindelser selv om variansen i forsinkelse er liten. Det kan derfor ikke regnes for sikkert at slike nyvinninger vil kunne være tilstrekkelige løsninger for ytelsen.

9 Konklusjon

Ytelsen for web-protokoller over mobile pakkesvitsjede bærersystemer er dårligere enn teoretisk maksimum. Studier av TCP/IP over GPRS viste at det ikke er GPRS, men TCP som står for de mest betydelige reduksjonene i ytelsen.

GPRS tilbyr robust overføring over radioforbindelsen. Dette betyr at sammenhengen mellom radioforhold som C/I og BER har liten innflytelse på ytelsen på applikasjonsnivå. Likevel vil hendelser knyttet til radionettet, som cell reselection og deling av ressurser, kunne gi direkte virkning på TCP i form av henholdsvis tapt IP-pakke og forsinkelse, med påfølgende time-out. Disse fenomener er holdt utenfor omfanget av denne oppgaven.

9.1 Videreføring

En naturlig videreføring av prosjektet vil være å se på ytelse og forsinkelse i UMTS. Samtidig er det flere deler av TCP som kan være interessante å se på.

10 Referanser

- [1] M. Zuckerman, "Circuit allocation and overload control in a hybrid switching system", Computer Networks and ISDN systems, vol 16, no. 4, 1989, pp 281-298.
- [2] ITU - International Telecommunication Unit, <http://www.itu.int/>
- [3] UTRAN Advanced Textbook, Windermind AB 2000
- [4] "WCDMA for UMTS", Harri Holma, Antti Toskala, ISBN: 0-470-84467-1
- [5] "Computer Networks, A Systems Approach", second edition, Larry L. Peterson and Bruce S. Davie, ISBN 1-55860-514-2(cloth.), ISBN 1-55860-577-0(paper).
- [6] "TCP over packet radio link with adaptive channel coding", Dongjie Huang and James J. Shi, Ericsson Inc., 0-7803-5718-3/00/\$10.00 ©2000 IEEE
- [7] Improving TCP Ytelse over Two-Way Asymmetric Links: Analysis and Solutions
- [8] Link Layer Analysis of the General Packet Radio Service for GSM, Reiner Ludwig and Dalibor Turina
- [9] 3GPP TS 04.60 V7.9.0 (2001-04): RLC procedures in packet transfer mode
- [10] <http://www.item.ntnu.no/fag/SIE5010/2001/wu-6p.pdf>
- [11] <http://www.umtsworld.com/technology/overview.htm> 4.2., 4.2.4,
- [12] "Radio Network Planning and Optimisation for UMTS", J.Laiho, A Wack'er, T. Novosad, Nokia Finland, ISBN 0 471 48653 1
- [13] The Pollaczek-Khintchine formula for M/G/1 queues
<http://www.fabtime.com/p-k.htm>, og
<http://www.mcs.vuw.ac.nz/courses/OPRE352/Includes/quelectures/PK/node3.html>
- [14] <http://www.netcom.no>
- [15] Telematikknett, Tore Riksaasen, Universitetsforlaget 1995 ISBN: 82-00-41489-2
- [16] <http://www.umts-forum.org/>
- [17] <http://www.umts.no/UtIVerden.htm>
- [18] <http://www.gsmworld.com/technology/gprs/intro.shtml>
- [19] "DataCommunications, Computer Networks and Open Systems", Fred Halsall, Fourth Edition, ISBN 0-201-42293-X

11 Ordliste/forkortelser

ARQ	Automatic Repeat Request
BCCH	Broadcast Control CHannel
BER	Bit Error Rate
BLER	Block Erasure Rate
BSC	Base Station Controller
BTS	Base Tranceiver Station
C/I	<i>Carrier/-to-interference ratio, C/I</i> , er et mål for radiolink-kvaliteten i GSM. C/I er forholdet mellom styrken på det mottatte ønskede signalet og styrken på det mottatte uønskede signalet. Forholdet påvirkes av faktorer som miljøbestemte hindre, antenne karakteristikker og lokale kilder til interferens. Dette betyr at C/I endres ettersom MS beveger seg.
CDMA	Code Division Multiple Access
CN	Core Network, kjernenettverket
CS	CS nettverk tilbyr linjesvitsjede forbindelser, som ISDN og PSTN
Cu	elektriske grensesnitt mellom USIM og ME. Grensesnittet følger en standard format for smartkort
FIFO	First in, First out
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
ICMP	Internet Control Message Protocol
Interleaving	Brukes for å oppnå tidsforskjeller i et digitalt kommunikasjonssystem uten å legge til noe overhead. Interleaving har blitt en meget nyttig teknikk i alle andre generasjons digitale cellulære systemer, på grunn av den raske forøkningen av digitale talekodere som transformerer analog tale til rasjonelle digitale meldinger som transmitteres over trådløs link.
IP	Internet Protocol
ITU	International Telecommunication Union
Iu	Åpent grensesnitt som knytter UTRAN til CN. (RNC - MSC/VLR, SGSN grensesnitt).
Iub	Åpent grensesnitt som knytter sammen RNC og Node B.
Iur	Åpent grensesnitt som tillater soft handover mellom RNC'er fra ulike produsenter og derfor gjør det åpne Iu grensesnittet komplett.
LLC	Logical Link Control
ME	Mobile Equipment
MTU	Maximum Transfer Unit
MTU	Maximum Transmission Unit
PCU	Packet Control Unit
PDCH	Packet Data Channel
PDU	protocol data unit
PS	PS nettverk tilbyr forbindelser for pakkesvitsjede datatjenester, som Internett
QoS	Quality of Service
RF	Radio Frequency
RLC	Radio Link Control

RNC	Radio Network Controller
RNS	Radio Network Sub-systems
RTSP	Real-time Streaming Protocol
SGSN	Serving GPRS Support Node
SMS	Short Message Service
SNDCP	Subnetwork Dependent Convergence Protocol
Soft handover	En terminal knyttet til nettverket via to eller flere aktive celler
TBF	Temporary Block Flow
TBF	Temporary Block Flow
TCP	Transmission Control Protocol
UE	User Equipment
Um	Interface mellom MS og GSM-delen av nettverket
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
Uu	WCDMA radiogrensesnittet. Gjennom dette grensesnittet aksesserer UE "the fixed part" av systemet og er derfor sannsynligvis det viktigste åpne grensesnittet i UMTS
WCDMA	Wideband Carrier Division Multiple Access

12 Vedlegg

Vedlegg 1 – Oppgavebeskrivelse

Vedlegg 1 - Oppgavetekst

”Motivasjonen for studier av webtrafikk over mobil pakkesvitsjede bærertjenester er at utnyttelsesgraden, spesielt med TCP over radio, generelt blir ganske dårlig. Effektive datarater er betydelig lavere enn det som forventes teoretisk. UMTS er på stadig frammarsj, og NetCom har sammen med Siemens startet opp et prosjekt for å optimalisere radiogrensensnittet i UMTS.

På bakgrunn av dette ønsker NetCom å undersøke utvalgte web- og Internettprotokollers ytelse over GPRS og UMTS ved å modellere, optimalisere og eksperimentelt verifisere protokollene.

Den eksperimentelle delen vil konsentrere seg om radionettet, som antas å representere de største bidrag til forsinkelse og hastighet.

Et utvalg av relevante protokoller vil bli identifisert og beskrevet. Kommunikasjonssystemet skal modelleres med hensyn på spesifikke, variable størrelser knyttet til utvalgte protokoller og kanal-karakteristikker. Eksperimentene vil bli utført både i test-nett og i live-nett. ”