EDI i elektronisk samhandling

Versjon 3.0

Publikasjonsserie fra NorStella

Hefte 4

Kommunikasjons-løsninger for utveksling av EDI-meldinger

Januar 2000

NorStella
C. J. Hambros plass 2C
0164 Oslo

Tel. 22 99 61 00
Fax. 22 99 60 10
Internet: www.norstella.no

Forord

Norsk veiledning i bruk av EDIFACT ble første gang utgitt februar 1991. Allerede i november samme år ble versjon 2 publisert. I 1994 ble det så utgitt en oppdateringsversjon. Siden den gang har det skjedd mye når det gjelder både datakommunikasjon spesielt og EDI (Elektronisk datautveksling) generelt. Internett er blitt "allemannseie", og det vokser frem nye formater for dataoverføring. Samtidig er "*electronic commerce*" blitt et hett tema internasjonalt. I sin videste form innbefatter det alt som kan ha med forretning å gjøre: mekling, varebestilling, varelister, kontraktsforhandlinger, betalingsformidling, osv. Det foregår en diskusjon om hva man skal kalle dette på norsk, men vi velger ganske enkelt å kalle det for *elektronisk handel*. Til elektronisk handel vil man ikke bare bruke EDI, men flere teknologier: www for å søke informasjon, elektronisk post til forespørsler og overføring av avtaler, elektronisk skjema til enkle bestillinger og mer tradisjonell EDI til de tunge forretningstransaksjonene. Fortsatt vil EDI kreve at man har et standard format som kan forstås av applikasjoner i begge ender, og Internett vil nok etter hvert bli en vanlig kanal for overføring av EDI-meldinger. Det siste vil avhenge av at man har tilstrekkelig sikkerhet.

I tillegg til elektronisk handel er det også behov for elektronisk overføring til administrative formål som i helsevesenet, til innrapportering fra næringslivet til det offentlige og til overføringer internt i det offentlige. Dette er ikke *handel* i tradisjonell forstand og vi benytter derfor begrepet *elektronisk samhandling* som et samlebegrep for ulike former for elektronisk interaksjon mellom to eller flere parter ved hjelp av elektroniske virkemidler som EDI, www, mm. Begrepet dekker elektronisk handel, offentlig administrasjon, helse, mm., og partene kan være forvaltning-forvaltning, forvaltning-næringsliv og næringsliv-næringsliv samt innbyggere-forvaltning og forbrukere-næringsliv.

I denne tredje versjonen av veiledningen har vi endret både format, innhold og navn på publikasjonen. Navnet er endret til ***EDI i elektronisk samhandling***. Det er flere grunner til dette, men de viktigste er.

- Etter vår oppfatning har EDI en sentral plass i det som omfattes av begrepet elektronisk samhandling. Vi ønsker å markere dette, samtidig som vi erkjenner at det finnes andre typer løsninger som er velegnet for sine formål.

- Elektronisk samhandling er et stort og omfattende område i stadig endring. For stort til at vi tør å begi oss ut på en utdypende behandling av alle typer løsninger. I denne veiledningen fokuserer vi på bruk av EDI. Vi ønsker på denne måten å videreføre tradisjonen fra de tidligere versjoner av veiledningen og tilrettelegge for en forenklet innføring og bruk av EDI, samt bidra til en ensartet fortolkning og bruk av standarden EDIFACT.

Når det gjelder format har vi endret dette slik at det forhåpentligvis er lettere tilgjengelig for brukerne. I stedet for en stor og kompakt perm har vi valgt å utvikle veiledningen som et sett av temabaserte hefter. Denne versjonen av veiledningen fremstår derfor mer som en publikasjonsserie.

Pr. januar 2000 er det utgitt følgende hefter:

1. ***Grunnleggende begreper og teknologier***
   Gir en generell oversikt over alternative teknologier for å realisere løsninger for elektronisk samhandling, samt plasserer EDI og EDIFACT i forhold til øvrige teknologier og gir hjelp til å velge mellom ulike alternative løsninger og teknologier.

2. ***Implementering av EDI-løsninger***
   Gir en oppsummering av erfaringene med realisering av EDI-løsninger, samt identifiserer en del viktige problemstillinger og fallgruver man bør være oppmerksom på ved innføring av EDI.

3. ***Bruk av modellering ved realisering av EDI-løsninger***
   Gir en generell oversikt over fagfeltet modellering sett i sammenheng med EDI-løsninger. Det gis blant annet en grundig genomgang av hvordan virksomhetsmodellering kan bidra til å bedre kvaliteten i en EDI-løsning.

4. ***Kommunikasjonsløsninger for utveksling av EDI-meldinger***
   Gir en oversikt over de krav som bør stilles til kommunikasjonsløsninger for utveksling av EDI-meldinger.

5. ***Avtaler i et EDI-forhold***
   Gir en oversikt over de avtalemessige forhold som bør være tilstede for å sikre en utveksling av EDI-meldinger.

6. ***EDIFACT for satsvis EDI***
   Gir en detaljert innføring i hvordan UN/EDIFACT bør utnyttes for å realisere løsninger for satsvis EDI.

7. ***EDIFACT for interaktiv EDI***
   Gir en detaljert innføring i hvordan UN/EDIFACT bør utnyttes for å realisere løsninger for interaktiv EDI.

8. ***Meldingssikkerhet i EDIFACT-løsninger***
   Gir en generell oversikt over sikkerhet i tilknytning til EDI-løsninger, samt gir en detaljert innføring i hvordan sikkerhetsmekanismene i UN/EDIFACT kan utnyttes for dette formål.

9. ***Overføring av assosierte data i en EDIFACT-løsning***
   Gir en generell oversikt over alternative teknologier for å overføre data i andre formater i tilknytning til EDI-løsninger, samt gir en detaljert innføring i hvordan UN/EDIFACT bør utnyttes for dette formål.

Hvert av de ovenstående heftene er et selvstendig verk som ikke forutsetter at de studeres i sammenheng. Unntaket er det første heftet - ***Grunnleggende begreper og teknologier*** - som gir en samlet oversikt over fagområdet elektronisk handel og administrasjon. Det anbefales derfor at dette heftet studeres innledningsvis, uansett hvilke av de andre delene av veiledningen som er av interesse.

Hovedmålsettingen med veiledningen er å etablere en oppdatert nasjonal veiledning som skal bidra til en forenklet innføring og bruk av EDI, samt en ensartet fortolkning og bruk av

EDIFACT-standarden. Veiledningen skal også gjenspeile dagens situasjon og forventet utvikling innenfor EDI og annen elektronisk samhandling, herunder klargjøre hva som ligger i konseptet elektronisk samhandling. Gjennom dette ønsker vi at veiledningen skal:

- rettlede en utvidelse av eksisterende EDI-plattformer,
- forenkle utviklingen av EDI-anvendelser og dermed bidra til en raskere utbredelse og harmonisering av eksisterende prosjekter i offentlig og privat sektor, herunder relevante prosjekter i NIN og i IT-programmet for næringslivet.
- effektivisere utviklingen av EDI ved at man unngår dobbeltarbeid og derved redusere kostnadene,
- redusere potensielle problemer når EDI tas i bruk på tvers av bransjer og sektorer,
- bidra til en rasjonell og effektiv utnyttelse av den nye EDIFACT-standarden innenfor nye områder,
- harmonisere bruken av EDIFACT i Norge,
- gjøre de ulike deler av EDI-utviklingen mer tilgjengelig og interessant for ulike målgrupper,
- tjene som undervisnings- og opplæringsmateriell.

De enkelte heftene i veiledningen er utarbeidet av ulike fagmiljøer på oppdrag av Norsk EDIPRO. Det faglige ansvaret for veiledningen er tillagt en prosjektgruppe bestående av følgende personer

Jostein Frømyr, EdiSys AS (prosjektleder)
Ellen Brox, NORUT IT
Anund Lie, Norsk Regnesentral
Olav Jan Hellesø, Statens Forvaltningstjeneste
Arild Nybakk, Skandinavisk Transport System AS
Finn Eide, Norsk EDIPRO

De enkelte heftene er videre behandlet i en referansegruppe med representer fra ledende virksomheter i Norge. Gjennom dette arbeidet har heftene fått bred støtte og tilslutning fra bl.a. Norsk Hydro, Den norske Bank, Kommunenes Sentralforbund og Arbeids- og administrasjonsdepartementet.

Norsk EDIPRO tar sikte på å etablere et eget fora som vil bli tillagt ansvar for en løpende vurdering av aktualiteten av de enkelte heftene. På anmodning fra dette forumet vil så de enkelte fagutvalg i Norsk EDIPRO være ansvarlig for ajourføring av det faglige innholdet.

Oslo, januar 2000
Jostein Frømyr
prosjektleder

Dette heftet spesielt

Dette heftet tar for seg kommunikasjonsløsninger for utveksling av EDIFACT-utvekslinger, dvs. meldingshåndteringssystem og underliggende transportnett. For EDI er elektronisk post (MHS) stadig den mest aktuelle løsningen, men Internett epost (SMTP/MIME) overtar i forhold til X.400. Vi har derfor prøvd å være nøytrale i forhold til forskjellige protokoller for elektronisk post, og bruker en del plass på å sammenligne X.400 og SMTP/MIME. Denne sammenligningen er samtidig en gjennomgang av egenskapene i MHS-tjenester og hvordan de relaterer seg til EDI.

Jo lenger ned i protokollhierarkiet en kommer, jo mindre EDI-spesifikke er kravene. Derfor er det liten grunn til å gi noen lærebok-aktig diskusjon av transport- og nettverkslag, sikkerhet, tjenestekvalitet osv. på disse nivåene. Det som finnes av generell litteratur her gjelder for EDI like godt som for andre nettverksanvendelser, og er mer fullstendig og uttømmende enn det ville være mulig å gi innenfor veiledningens ramme. Vi har derfor konsentrert oss om å introdusere stoffet og prøve å få frem de momentene og problemstillingene som er mest vesentlige for en som planlegger en EDI-løsning.

Selv om sikkerhet primært tas opp i hefte 8, blir sikkerhetskravene ofte nedfelt i krav til kommunikasjonsløsningen, og vi kan ikke unngå å komme tilbake til sikkerhetskrav i dette heftet. Derfor er det noe overlapp med hefte 8. Men mens hefte 8 primært tar for seg meldingssikkerhet, ligger hovedvekten her på sikkerhetsmekanismer på lavere lag i protokollhierarkiet, dvs. på transport- eller nettverkslaget.

Egne kapitler kommer inn på tema som er av spesiell betydning for EDI:

- Bruk av tjenesteleverandører, og dermed tjenesteleverandørens rolle i løsningen.
- Bruk av portnere for konvertering mellom protokoller og formater.

Vi mener at det i EDI-sammenheng har vært fokusert vel mye på realiseringen av transportløsningen. Kravene som EDI-anvendelsen stiller til transportløsningen er det primære, realiseringen av dem kommer i annen rekke. Siste kapittel er derfor en forholdsvis fyldig oversikt over de krav eller typer av krav som er aktuelle for EDI. Dette kapitlet kan gjerne brukes som en sjekkliste. Det krever nok noe mer bakgrunn innen datakommunikasjon enn resten av heftet, men bakgrunnsstoffet fra de foregående kapitlene samt ordforklaringene i vedlegg A bør være tilstrekkelig til å kunne lese det.

Prosjektgruppen ønsker å rette en spesiell takk til Anund Lie, Norsk Regnesentral, som i hovedsak har ført dette heftet i pennen.

Jostein Frømyr
prosjektleder

Innholdsfortegnelse

1 Innledning ... 1

2 Meldingsutvekslingssystemet ... 2
2.1 EDI-arkitektur og datanett-arkitektur ... 2
2.2 EDI — krav til MHS ... 3
2.3 MHS, filoverføring eller andre typer protokoller? ... 4
2.4 Synkron og asynkron meldingsutveksling ... 5
2.5 MHS: X.400 eller Internett epost-protokoller? ... 7
2.5.1 Sammenligning av spesifikasjonene ... 8
2.5.2 Meldingsoverføring ... 9
2.5.3 Meldingslager ... 9
2.5.4 Basis meldingsformat ... 10
2.5.5 Meldingsinnhold ... 10
2.5.6 Leveringsbekreftelse ... 10
2.5.7 Mottakskvittering ... 11
2.5.8 Transportnivå sikkerhet ... 12
2.5.9 Meldingsnivå sikkerhet ... 13
2.5.10 EDI-meldinger ... 13
2.5.11 Sikkerhet og pålitelighet ... 13
2.5.12 Konklusjoner ... 15

3 Transportnettet ... 16
3.1 IP-protokollen ... 16
3.2 Ulike typer lukkede nett ... 18
3.3 Intra- og ekstranett ... 18
3.4 Brannvegger ... 20
3.5 Transportlagssikkerhet ... 23
3.6 Virtuelle private nett ... 26

4 Tjenesteleverandørens rolle ... 29
4.1 Tjenestetyper ... 29
4.2 Sikkerhetsmessige implikasjoner ved nettløsningen ... 29
4.3 Andre krav til tjenestekvalitet ... 30

5 Protokoll- og formatkonvertering gjennom portnere ... 32

6 Krav til meldingsutvekslingssystemet ... 36
6.1 Innhold ... 36
6.1.1 Transparens ... 36
6.1.2 Utvekslingslengde ... 37
6.1.3 Identifikasjon av EDIFACT-utvekslinger ... 38
6.2 Adressering og ruting ... 38
6.3 Sikkerhet ... 39
6.3.1 Meldingssikkerhet ... 40
6.3.2 Kommunikasjonssikkerhet ... 41
6.3.3 Konfidensialitet ... 41
6.3.4 Integritet ... 41
6.3.5 Autentisitet ... 42
6.3.6 Ikke-benekting ... 42
6.3.7 Sekvensintegritet ... 44
6.3.8 Tjenestenekting ... 45

6.3.9 Vertsmaskinsikkerhet ... 45
6.4 Andre krav til tjenestekvalitet ... 46
6.4.1 Tilgjengelighet ... 46
6.4.2 Tap av meldinger ... 46
6.4.3 Kapasitet ... 47
6.4.4 Overføringstid ... 47
6.5 Andre krav ... 48
6.5.1 Kvittering og feilmeldinger ... 48
6.5.2 Operative forhold ... 49
6.5.3 Drift og overvåkning ... 50

A Terminologiliste ... 53

B Litteraturliste ... 65
B.1.1 ISO-standarder og profiler ... 65
B.1.2 ITU-rekommendasjoner ... 66
B.1.3 Internet Request For Comments (RFC) og Internet Drafts ... 67
B.1.4 Bransjestandarder og andre spesifikasjoner ... 69

1 Innledning

I den gamle versjonen av denne veiledningen ble kun én form for kommunikasjon mellom EDI-brukere anbefalt, nemlig ved X.400-protokollen og bruk av formidlingssentral (lukket nett). Siden 1994 har alternative kommunikasjonsformer blitt stadig mer aktuelle. Spesielt har Internett og Internett-teknologi fått stor utbredelse, og det utvikler seg stadig i retninger som gjør det bedre egnet for bruk i virksomhetskritiske systemer, herunder EDI.

I kapittel 3 i del I av denne veiledningen finnes en mer overordnet diskusjon av ulike teknologier som er aktuelle i forbindelse med EDI, herunder Internett. Der finnes også en skjematisk arkitektur for EDI-løsninger. I dette kapitlet kommer vi nærmere inn på de lavere lagene i denne arkitekturen, nemlig de komponentene som har til oppgave å transportere EDIFACT-utvekslinger[1] mellom avsender og mottaker. Vi betegner disse komponentene under ett som *meldingsutvekslingssystemet* (MHS[2]) Dette kapitlet dekker først og fremst satsvis EDI, bare i liten grad interaktiv EDI.

Figuren nedenfor viser skjematisk arkitekturen for et MHS. *Brukeragentene* er de komponentene som **mottar** og **sender** utvekslinger på vegne av brukerne, altså endepunktene i den veien en utveksling tar gjennom MHS. *Meldingstransportsystemet* (MTS) er det delsystemet som har til oppgave å **videreformidle** utvekslingene fra senderens til mottakerens brukeragent. MTS består av *transportagenter* som kommuniserer gjennom en eller annen form for transportnett (her bare tegnet inn som piler).

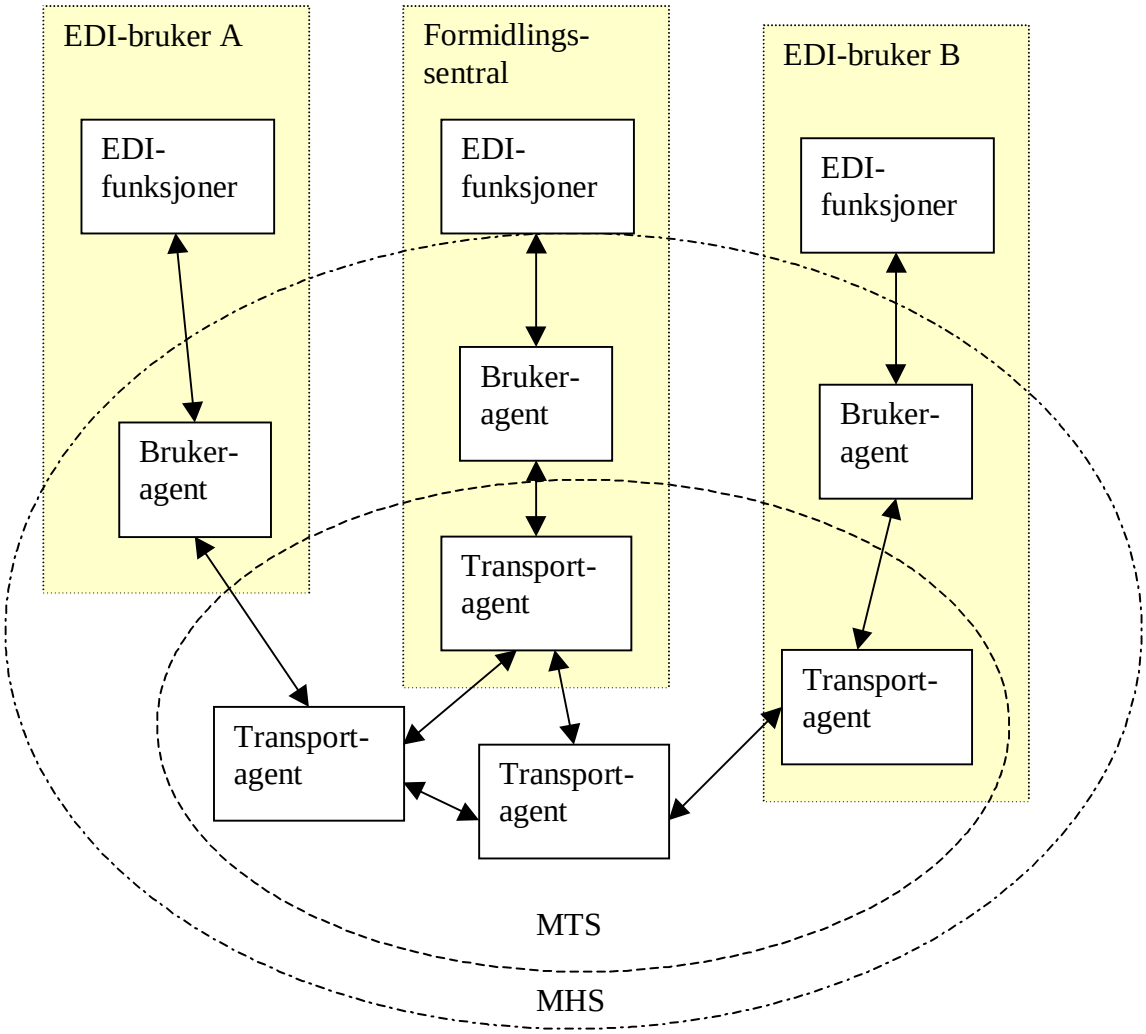
De øvre lagene i EDI-arkitekturen, som faktisk forholder seg til innholdet i EDIFACT-utvekslingene, går vi ikke nærmere inn på her, og i figuren nedenfor er delene slått sammen og betegnet som ”EDI-funksjoner”.

Hver av komponentene opereres av eller på vegne av en bestemt organisasjon. I figuren er to EDI-brukere, A og B, og en formidlingssentral tegnet inn. Formidlingssentralen tilbyr EDI-rettede verdiøkende tjenester, og har derfor både en brukeragent og EDI-funksjoner. Merk at en slik formidlingssentral, sett fra MHS, i prinsippet mottar en utveksling og sender en ny utveksling. Den har altså et aktivt ansvar for innholdet i utvekslingen, også for det grensetilfellet at den nye utvekslingen bare er en kopi av den opprinnelige. Dersom den endelige mottakeren benytter de mekanismene som er tilgjengelige i MHS for å finne avsenderen, vil disse nemlig referere til formidlingssentralen.

[1] Merk at betegnelsen *melding* brukes i to betydninger i dette heftet. En *melding* er det som meldingsutvekslingssystemet transporterer og kan altså som et spesialtilfelle inneholde en EDIFACT-*utveksling*. I dette heftet vil vi stort sett bruke betegnelsen *EDIFACT-melding* når vi snakker om en enkelt-melding innenfor en EDIFACT-utveksling, men vi dropper prefikset når det er klart av sammenhengen at det er snakk om EDIFACT-melding og ikke MHS-melding.

[2] Vi bruker den innarbeidede forkortelsen MHS for den engelske betegnelsen Message Handling System, og i en noe videre forstand enn vanlig, som dekker både X.400, SMTP, filoverføring og andre typer protokoller og systemer som kan anvendes til transport av EDI-meldinger.

2 Meldingsutvekslingssystemet



Figur 1: Komponenter i meldingsutvekslingssystemet

2.1 EDI-arkitektur og datanett-arkitektur

Vår beskrivelse av EDI-arkitekturen er inspirert av den lagdelte arkitekturen i datanett (protokoller, kommunikasjonsprogramvare). OSI-modellen har 7 lag, mens Internett-modellen har 4, som i grove trekk faller sammen med lag 2-4 (nedenfra) samt 7 (det øverste) i OSI-modellen. Hvert av lagene benytter seg av tjenestene i laget under for å realisere sin egen tjeneste, og sørger selv for at laget over slipper å forholde seg til detaljer i laget under. De 4 lagene som er felles er:

Applikasjon. Programmene som benytter seg av kommunikasjonstjenester hører hjemme i applikasjonslaget, dvs. klient- og tjenerprogrammer.

Transport: Transportlaget gjør det mulig å sette opp ”endepunkter” (porter i TCP og UDP) og, for såkalt ”forbindelsesorienterte” tjenester, forbindelser mellom disse endepunktene som er ”virtuelt feilfrie”.
Nettverk: Nettverkslaget styrer kommunikasjon maskin til maskin, og har ansvaret for å rute trafikken via mellomliggende noder når maskinene ikke er direkte sammenkoblet, samt for å håndtere detaljer som avhenger av hvilke nett trafikken er innom (f.eks. velge en maksimal pakkestørrelse).
Linje/grensesnitt: (data link/interface) Dette laget styrer trafikken på et enkelt hopp fra node til node og tilpasser nettverkslaget til den fysiske nettverkstilkoblingen.

Merk: Applikasjonslaget i datanett-sammenheng omfatter praktisk talt hele EDI-løsningen: Transportagent, brukeragent, EDI-programvare og det som EDI-brukerne vil oppfatte som applikasjonsprogramvare. Vær altså oppmerksom på denne forskjellen i språkbruk mellom to miljøer. Transportagenten (MTA) er en del av applikasjonslaget, ikke transportlaget.

Forskjellen mellom transport- og nettverkslag kan være litt vanskelig å få tak på, men er svært vesentlig, f.eks. for å forstå forskjellen mellom sikkerhetsmekanismer på transport- og nettverkslag:

- Applikasjonsprogrammene kan lage så mange transport-endepunkter som de har bruk for, f.eks. vil en tjener vanligvis ha ett for hver klient den betjener, og transportlaget har ansvaret for å holde datastrømmene fra hverandre. Nettverkslaget har ett og bare ett endepunkt for hver maskin.
- Nettverkslaget **garanterer ikke** at data som er sendt kommer fram, eller blir levert intakt, men gjør bare et ”beste forsøk”. Transportlaget (forbindelsesorientert) garanterer vanligvis både at data kommer frem, uten feil, og i rett rekkefølge.

Transportlaget gir altså applikasjonsprogrammene illusjonen av å kunne koble seg direkte til et annet applikasjonsprogram på en annen maskin, via en ideell kanal som ikke mister eller forvrenger data.

2.2 EDI — krav til MHS

Transportfunksjonen som MHS utfører er i sin natur generell og i utgangspunktet uavhengig av at datainnholdet som transporteres faktisk er EDIFACT-utvekslinger. Protokollene som benyttes på disse nivåene er også stort sett de samme som benyttes for andre typer meldingsutveksling. EDIFACT er bare en av flere mulige typer innhold som disse protokollene kan transportere. Datainnholdet blir ikke tolket underveis, og protokollenes oppgave er å sørge for at det som kommer frem til mottaker er identisk med det som ble sendt fra avsender.

EDI-meldingsutveksling stiller likevel en del spesielle krav:

- Brukeragenten har et program-grensesnitt mot EDI-programvaren. For person-til-person epost er brukeragenten et vanlig epost-program som styres av brukeren, mens en EDI-brukeragent er et subprogram som kan styres av (f.eks. kalles fra) et annet program (f.eks. EDI-programvaren som har gjort klar en EDIFACT-utveksling som skal sendes).
- Adressering av EDIFACT-utvekslinger skjer vanligvis ut fra EDI-parts-identifikatorene i utvekslingshodet. Det er også annen informasjon i utvekslingshodet som potensielt er av interesse for MHS. EDI-brukeragenten

kan derfor ha behov for å tolke utvekslingshodet, selv om dette strengt tatt er et brudd på lagdelingsprinsippet.

- EDI stiller strengere krav til sikkerhet, pålitelighet, tilgjengelighet, kapasitet m.m. Disse kravene er ikke vesensforskjellige fra de som gjelder for meldingsutveksling generelt, og de tilsier vanligvis ikke EDI-spesifikke protokoller og mekanismer. Men en kan ikke uten videre gå ut fra at kravene oppfylles av produkter (utstyr, programvare, tjenester) for mindre krevende anvendelser, som f.eks. vanlig person-til-person-epost.
- Satsvis EDI krever bare enveis, asynkron meldingsutveksling, dvs. uten at avsender og mottaker har noen direkte interaksjon, og eventuelle kvitteringsmeldinger kan sendes tilbake med en viss tidsforsinkelse.

Normalt vil det være mulig å gjøre en EDI-løsning nokså uavhengig av hvilken løsning som er valgt for meldingsutvekslingen. EDI-funksjonene (de overordnede, anvendelsesspesifikke komponentene) ser i prinsippet bare EDIFACT-utvekslinger eller EDIFACT-meldinger, og skal i utgangspunktet ikke ha behov for å kjenne til detaljer om hvordan disse skal bli eller har blitt utvekslet. Det kan være aktuelt å ha flere alternative kommunikasjonsløsninger innenfor samme EDI-løsning, f.eks. for å nå forskjellige kundegrupper med ulike former for nettverkstilknytning.

Dette betyr også at en kan tillate seg å være pragmatisk i valget av kommunikasjonsløsninger. Hver ny løsning som tas i bruk krever selvfølgelig en investering i utstyr, programvare, opplæring osv., men dette bør kunne vurderes noenlunde isolert fra resten av EDI-løsningen. Det er ikke først og fremst de funksjonelle egenskapene ved EDI-anvendelsen (meldingstyper, gangen gjennom meldingsscenarier og lignende) som er utslagsgivende. Det er viktigere å vurdere omkringliggende forhold, som hvilke partnere en skal kommunisere med og hvilke forutsetninger de har, hvilke andre kommunikasjonsbehov en selv har, tilgjengelighet og kostnad for tjenester, utstyr og programvare m.m.

Etter hvert som EDI blir mer utbredt, må en regne med at samme organisasjon benytter flere EDI-løsninger i forskjellige sammenhenger. Dessuten benyttes andre former for datakommunikasjon. Hvis det ikke er spesielle forhold som taler for det, kan og bør disse dele den samme infrastrukturen for datakommunikasjon og meldingsutveksling. Dette er hovedargumentet for å benytte åpne nett, dvs. Internettet.

2.3 MHS, filoverføring eller andre typer protokoller?

MHS-protokoller i egentlig forstand (elektronisk post: X.400 eller SMTP) er i sin natur asynkron meldingsutveksling. Denne typen protokoller er dermed mest direkte tilpasset kommunikasjonsmønsteret i EDI og klart mest utbredt. Men som nevnt legger EDI få restriksjoner på protokoller og nett, og det er mange ulike typer nett-tjenester som kan være aktuelle:

- Filoverføringsprotokoller (generelle som FTAM og FTP, eller EDI-spesifikke som ODETTE) er også svært utbredt. Avsender og mottaker har filområder på en tjener (oftest hos en formidlingssentral) som de benytter som postkasse og laster opp eller ned filer mot.
- HTTP (meldingsutvekslingsprotokollen i Web) kan også benyttes for transport av EDIFACT-utvekslinger.
- ”Remote Job Entry”, ”Network Job Entry”-protokoller for overføring av satsvise jobber til stormaskiner. (EDIFACT-utvekslingen plasseres som et

datasett mellom kommandoer som starter opp behandlingen av den hos mottaker.)

- Ulike proprietære protokoller. For en del formål kan det f.eks. være tilstrekkelig å overføre EDIFACT-utvekslinger over ”rå” transportforbindelser, ISDN-samband eller lignende.
- Vi skal heller ikke utelukke fysiske media, f.eks. disketter eller magnetbånd.

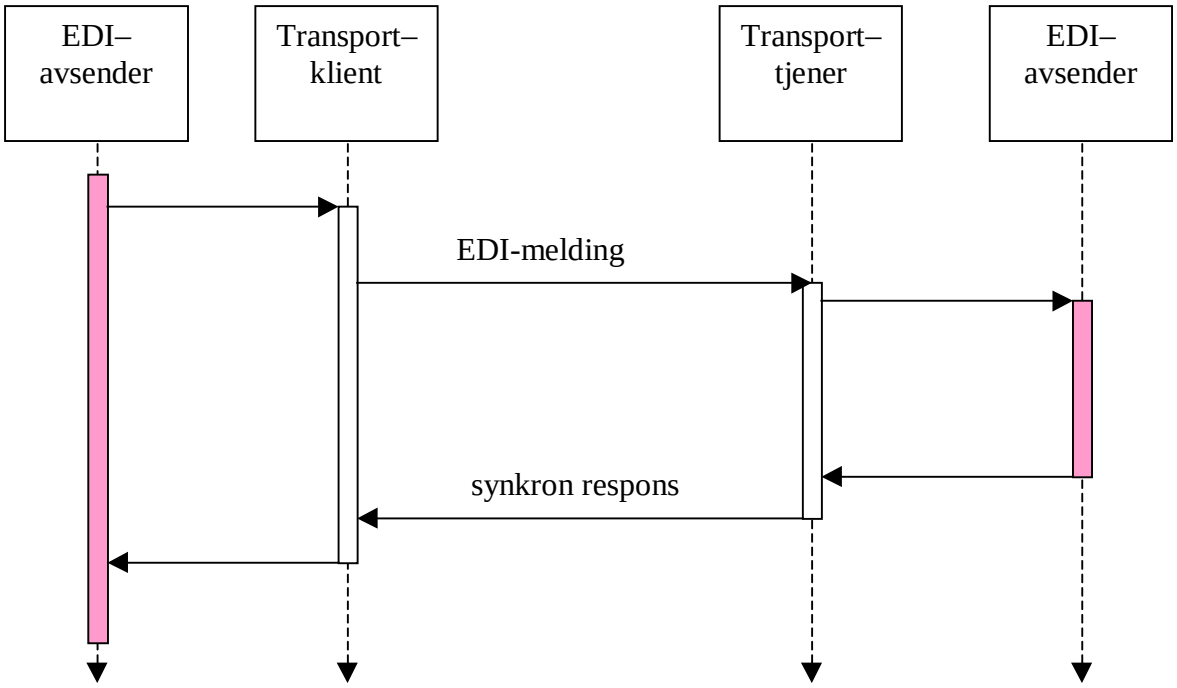
Vanligvis brukes disse andre protokollene på en måte som simulerer MHS-type meldingsutveksling. (Det er grunnen til at vi gjennomgående bruker MHS-terminologi i dette heftet.) Hovedgrunnen til å bruke andre protokoller enn MHS er vanligvis at det finnes rent praktiske hindringer for å bruke MHS-protokoller, f.eks. at EDI-løsningen må kjøre på en plattform der det ikke finnes brukbare implementasjoner av SMTP eller X.400.

Det er vanskelig å finne satsvise EDI-anvendelser hvor det er logikken i anvendelsen i seg selv som tilsier bruk av andre protokoller enn MHS, men det kan heller ikke utelukkes. FTP eller HTTP kan f.eks. egne seg dersom kommunikasjonsmønstret er en til mange, dersom det er mer naturlig at det er mottaker som aktivt henter meldinger i stedet for passivt å motta. (En av grunnene til at slike EDI-anvendelser ikke er vanlige, kan rett og slett være at meldingstyper og scenarier for satsvis EDIFACT er utviklet i en tradisjon der asynkron ”lagre og videresend”-meldingsutveksling har vært dominerende.)

En meget vesentlig grunn til å holde seg til X.400 og SMTP (og etter hvert HTTP) er at det bare er for disse protokollene det finnes standardiserte EDI-profiler. Ulike produkter og tjenester er rimelig interoperable, og det er lite behov for (eller spillerom for) lokale konvensjoner som en må tilpasse seg hos den andre parten. De andre protokoll(typene) vi har nevnt krever i praksis alltid lokale, proprietære konvensjoner, som det ofte vil være vanskelig å få sin egen kommunikasjonsprogramvare tilpasset. (Bruker en f.eks. FTP eller FTAM til mottakerens filområde, må en avtale konvensjoner om hvordan filnavnene er oppbygget og for å forhindre at en ved et uhell overskriver filer før de er behandlet, administrere brukernavn og passord til filområdene m.m.)

2.4 Synkron og asynkron meldingsutveksling

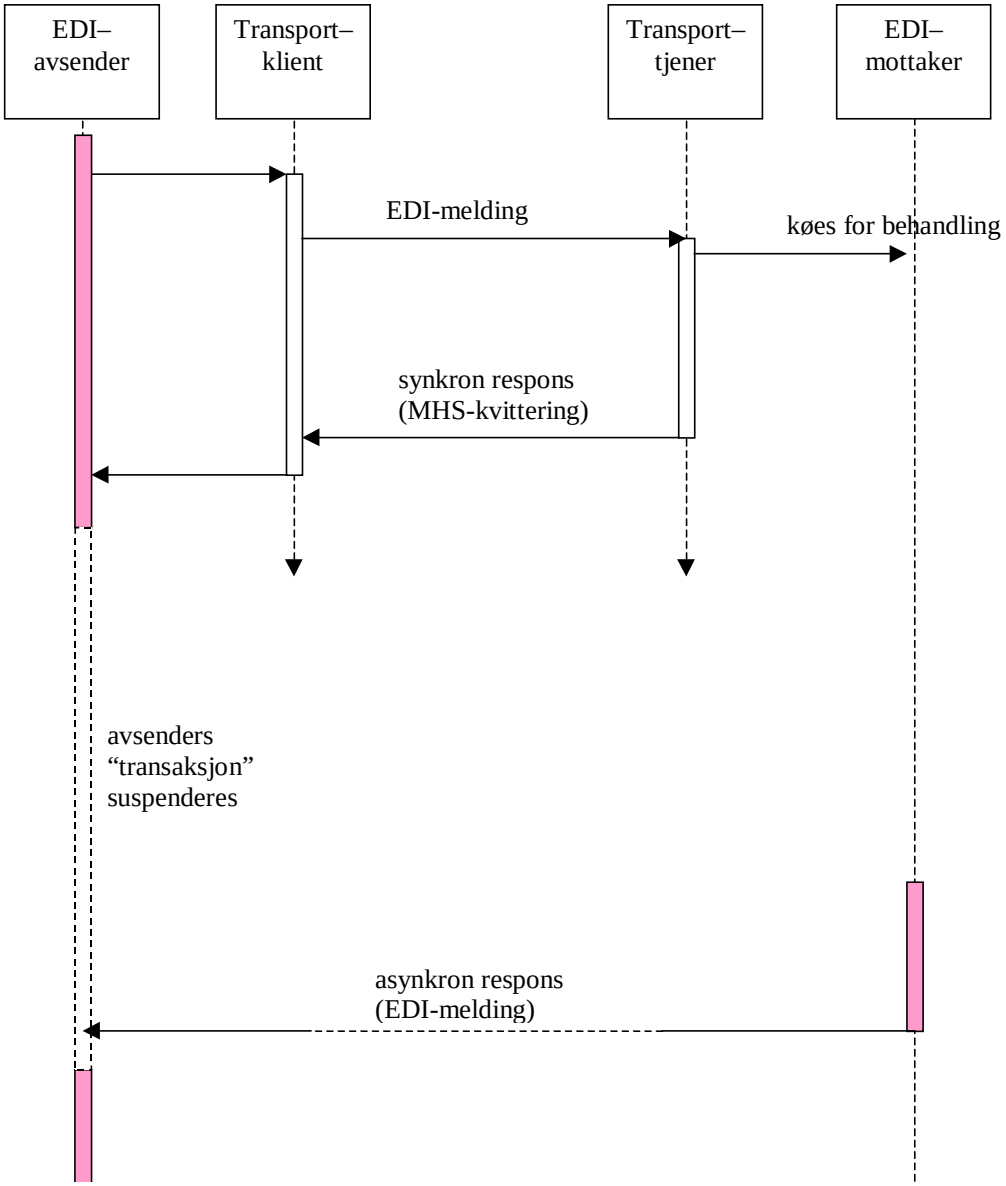
Enkelte av protokollene som er nevnt ovenfor er synkrone. Men synkroniteten i protokollene utnyttes i svært liten grad. Et unntak er HTTP, der hver interaksjon mellom klient og tjener består av en ”forespørsel” (request) og en ”respons” (response). I prinsippet kan forespørselen inneholde en EDI-melding, og responsen inneholde en applikasjonsnivå-respons på denne, dvs. en ny EDI-melding, som vist i figuren nedenfor.



Figur 2 Synkron EDI-behandling

Det som er er karakteristisk for satsvis EDI er nettopp at tiden det tar før meldingen behandles og den egentlige responsen foreligger er stor i forhold til tiden mellom forespørsel og respons i synkrone protokoller. I praksis er det uaktuelt å la avsenderen vente i en aktiv tilstand på at responsen foreligger. Dette vil for det første ta for mye ressurser (i form av aktive prosesser, transaksjoner, åpne nettverksforbindelser osv.) på sendende og mottakende maskin. Dessuten øker risikoen for brudd i nettet og for krasj (eller bare planlagt omstart) på en av maskinene i løpet av ventetiden, og avsender- og mottakerprogrammene må derfor i praksis utformes slik at all mellomtilstand lagres permanent i en database eller lignende, og det er mulig å gjenoppta behandlingen selv etter et slikt avbrudd.

EDIINT-spesifikasjonen for EDI over HTTP foreskriver derfor bare at den synkrone responsen består av en MHS-nivå mottakskvittering. Den egentlige responsen på EDI-meldingen, som er en ny EDI-melding, må sendes tilbake asynkront.



Figur 3 Asynkron EDI-behandling (satsvis EDI)

Interaktiv EDIFACT er synkron, men dette er en egen protokoll, som bør implementeres for seg på toppen av transportlaget. Verken MHS, filoverføringsprotokoller eller HTTP er særlig egnet som basis for interaktiv EDIFACT. (Dette er en ganske vesentlig forskjell mellom satsvis EDIFACT, ISO 9735-2, og interaktiv EDIFACT, ISO 9735-3, som slett ikke er åpenbar fra standardene selv.)

Vi kommer ikke nærmere inn på interaktiv EDI i dette heftet, men stor del av stoffet om transportnettet, sikkerhet, tjenestekvalitet osv. er like relevant for interaktiv som for satsvis EDI.

2.5 MHS: X.400 eller Internett epost-protokoller?

For elektronisk post finnes to konkurrerende sett av standarder, hhv. X.400 og Internett-standardene SMTP, RFC822 og MIME. Tidligere har X.400 vært anbefalt for EDI. Noe av motivasjonen for det har nok vært en antakelse om at X.400 ville bli den dominerende standarden for elektronisk post. Dette har ikke

skjedd, og Internett-standardene har fått en vel så dominerende plass. Tekniske forskjeller mellom standardene har også spilt en rolle, men delvis har disse nok vært overdrevet i en litt opphetet debatt, delvis har Internett-standardene blitt komplettert med nye spesifikasjoner på punkter som er vesentlige for EDI, så i utgangspunktet har ikke den ene standarden noen avgjørende tekniske fordeler fremfor den andre.

Selv om Internett-standardene har blitt svært dominerende, betyr det ikke at X.400 har blitt uaktuelt. Vi regner med at begge sett med standarder vil sameksistere fremover, og hva som er aktuelt å bruke må vurderes for hver enkelt anvendelse. X.400 og Internett-epostsystemer kan knyttes sammen gjennom portnere, og dessuten er det mulig å kjøre X.400 over Internett[1].

Den mest fundamentale forskjellen mellom Internett og X.400 ligger i den rollen tjenesteleverandørene er tenkt å ha. Internett-tjenester generelt (ikke bare epost) er ”ende til ende”, dvs. at det normale tilfellet er at avsender og mottaker, klient og tjener, knytter seg direkte til hverandre uten å gjøre bruk av tredjeparts tjenesteleverandører. Nettet selv har i prinsippet bare én funksjon, nemlig å transportere IP-pakker. X.400 har en ”lagre og videresend”-modell der tjenesteleverandørene har en mer direkte og aktiv rolle, også i implementasjonen av X.400-tjenesten selv. Avsender leverer meldingen til sin X.400-tjenesteleverandør, som overtar ansvaret for å levere den videre, kanskje via en kjede av andre operatører, til mottakerens tjenesteleverandør, for så å bli endelig levert til mottakeren.

I praksis er situasjonen noe mer nyansert: Også for Internett-epost vil mange brukere overlate til en tjenesteleverandør å operere selve SMTP-tjeneren. Det er heller ikke noe rent teknisk i veien for å bruke X.400 direkte mellom sender og mottaker uten noen tredjeparts tjenesteleverandør. Det kan imidlertid være administrative begrensninger som forhindrer dette.

Denne forskjellen gir opphav til en del av forskjellene i teknisk utforming, og kanskje først og fremst ulik prioritering av en del mekanismer: For eksempel oppfattes egne kvitteringsmeldinger som mindre vesentlige i Internett-sammenheng — når avsenderen har gjennomført en SMTP-sesjon mot mottakerens maskin, uten at noen feil er detektert, er meldingen faktisk levert til mottaker, og leveringskvitteringen er implisitt.

2.5.1 Sammenligning av spesifikasjonene

Tabellen nedenfor gir en sammenligning mellom X.400- og Internett-epost-spesifikasjonene, gruppert etter funksjoner. Merk at strukturen i spesifikasjonene er noe forskjellig: Mens X.400 typisk samler mer i hver enkelt spesifikasjon, og det blir gitt ut nye utgaver når enkelte deler revideres eller utvides, er Internett-spesifikasjonene delt opp i mindre moduler, der de grunnleggende delene ligger forholdsvis fast. (For X.400 er det store forskjeller mellom 1984- og 1988-utgavene, mens senere revisjoner er mindre gjennomgripende. Vi ser her helt bort fra 1984-utgaven.)

	X.400	Internett
Meldingsoverføring	P1: X.411 -, X.419 -	SMTP -, ESMTP -

[1] Og vice versa: SMTP kan i og for seg kjøres over hvilken som helst transportprotokoll, ikke bare TCP, selv om dette knapt forekommer.

Meldingslager	P7: X.413-, X.419	POP3-, IMAP4 -
Basis meldingsformat	X.411 (X.420-, X.435-)	RFC 822 -
Meldingsinnhold	X.420 (IPM) -, X.435 (EDIM) -	MIME ----
Leveringsbekreftelse	X.411 (DR/NDR)	DSN --
Mottakskvittering	X.420 (IPN), X.435 (EDIN)	MDN -
Transportnivå sikkerhet	X.411 + underliggende lag	SSL -, TLS -, --
Meldingsnivå sikkerhet	X.411, X.420 (PKCS#7 body part)-	S/MIME --, PGP/MIME - m.fl.
EDI-meldinger	X.435	RFC 1767 -, EDIINT--

Dette delkapitlet tar for seg forskjeller som skyldes tekniske egenskaper i protokollene selv. I praksis vil ofte kvaliteten på implementasjonen av programvare og tjenester, uavhengig av hvilke protokoller de er basert på, være vel så avgjørende.

2.5.2 Meldingsoverføring

Det er en del tekniske forskjeller i protokollene som benyttes for å overføre meldinger mellom transportagentene. X.400 har et stort antall parametre i sin ”envelope”-spesifikasjon, som styrer forskjellige aspekter av hvordan transportagentene behandler behandler meldingene. SMTP er i sammenligning svært enkel (og er i de fleste tilfeller tilstrekkelig), mens den nyere ESMTP-spesifikasjonen danner et rammeverk for å definere tjenesteutvidelser. Et eksempel på en slik tjenesteutvidelse er mekanismen definert i RFC 1891 for å be om leveringsbekreftelse (DSN, se nedenfor).

P1-protokollen i X.400 kan bare ”skyve” meldinger, dvs. at det er transportagenten som sender som tar initiativet til overføringen og opererer som klient. I SMTP kan klienten, altså transportagenten som tar initiativet til overføringen, snu overføringsretningen slik at den opererer som tjener og mottar meldinger. (Denne mekanismen er imidlertid ikke påkrevet, og er forholdsvis lite utbredt i implementasjoner av SMTP.)

Selv om protokollene skyver meldinger fra seg, betyr ikke det nødvendigvis at meldingene sendes videre så snart de kommer inn. Det er den konkrete MTA-implementasjonen som bestemmer når den skal kontakte neste ledd i kjeden. Ofte er det konfigurerbart hva som utløser videresending. F.eks.: når det er samlet opp en viss mengde trafikk som skal videre til en bestemt annen MTA, ved faste tidsintervaller o.l. Motivasjonen bak dette er å optimalisere bruk av kommunikasjonsnettet. Dette er funksjonalitet som ikke avhenger av protokollen, men av implementasjonen av den.

2.5.3 Meldingslager

Sluttbrukere vil ofte unngå å kjøre en transportagent selv, siden dette betyr at transportagenten og nettforbindelsen må være nærmest kontinuerlig tilgjengelig. I stedet vil tjenesteleverandøren deres levere meldinger til et meldingslager, og sluttbrukerne kobler seg opp og henter ut meldinger fra meldingslageret når det passer dem selv. Dette er også aktuelt for EDI. X.400 MS (Message Store) og IMAP4 (Internet Message Access Protocol) gir omtrent tilsvarende tjenester. Den enklere POP3-protokollen har kun én postboks på tjeneren, slik at brukeren i

praksis bør arkivere meldinger lokalt. (Lar brukeren meldingene ligge i postboksen på tjeneren, blir nye og arkiverte meldinger blandet.)

2.5.4 Basis meldingsformat

X.400 MTS transporterer i prinsippet meldinger med hvilket som helst innhold og format (tekst eller binært), og spesielt de to innebygde innholdstypene definert i X.420 og X.435. I Internett er det underliggende meldingsformatet definert av RFC 822.

RFC 822 definerer betydningen av en del generelle felter i meldingshodet, f.eks. avsender, mottaker, dato og overskrift. Tilsvarende meldingshode er definert i X.420 eller X.435. (Merk: Meldingshodet er en del av meldingen og passerer normalt gjennom MTS uten endringer, i motsetning til ”envelope” som leses og modifiseres av transportagentene underveis. Jfr. brevhode og konvolutt for vanlig post.) Andre meldingsorienterte tjenester i Internett (Web og News) benytter også RFC 822 (+ MIME) meldingsformat.

X.400-meldinger representeres i et binært format, mens RFC 822-meldinger er tekstlige. For en vanlig bruker er dette ikke så vesentlig, men for utvikling og drift kan RFC 822-meldinger være enklere å håndtere. Baksiden av medaljen er at binære data må kodes i tekstformat.

2.5.5 Meldingsinnhold

Basis RFC 822-meldinger kan kun inneholde tekst i ASCII-tegnkode. MIME-spesifikasjonene fyller to formål: For det første finnes mekanismer for å kode vilkårlig (binært) innhold så det lar seg overføre gjennom MTS. Videre beskriver MIME-spesifikasjonene hvordan meldingene skal merkes med innholdstype slik at de kan behandles korrekt og evt. automatisk av mottaker. En del innholdstyper er predefinert, f.eks. ”text/plain” for tekst (med angivelse av tegnsett og –koding for å støtte ulike nasjonale tegnsett). Men dessuten definerer MIME hvordan nye innholdstyper skal registreres. Katalogen over innholdstyper administreres av IANA, se ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/. En av disse innholdstypene er ”application/edifact”, som identifiserer innholdet i meldingen som en EDIFACT-utveksling.

X.400-meldinger identifiseres i første omgang som en av de innebygde meldingstypene ”inter-personal message” (X.420, IPM) eller ”EDI message” (X.435, EDIM). (Det finnes også andre meldingstyper, men det er disse som er aktuelle for EDIFACT.) X.420 er beregnet for meldinger som utveksles person-til-person (IPM), men er også i utstrakt bruk for EDI. X.435 definerer en egen meldingstype for EDI (EDIM). En IPM eller EDIM består av en eller flere innholdsdeler, enten av predefinert type definert i X.420 hhv. X.435, eller av en ekstern type identifisert ved en ASN.1 ”object identifier”. EDIFACT har en egen predefinert innholdstype i X.435.

2.5.6 Leveringsbekreftelse

En leveringsbekreftelse er en bekreftelse fra *MTS* (dvs. siste transportagent som hadde ansvaret for meldingen) om at meldingen er levert til mottakeren, evt. at meldingen ikke lot seg levere. Leveringsbekreftelsen gir ingen indikasjon på om

mottakeren har akseptert eller er i stand til å behandle meldingen. Hovedformålet med leveringsbekreftelser er å øke påliteligheten i meldingstransporten.

Positiv leveringsbekreftelse kan benyttes dersom en oppfatter MTS som upålitelig og ønsker en ekstra bekreftelse på at meldingen er levert. Avsenders brukeragent kan da evt. lagre sendte meldinger inntil positiv bekreftelse er mottatt. Positiv leveringsbekreftelse genereres **ikke** med mindre avsenders transportagent eksplisitt har bedt om det. Denne funksjonen er inkludert i X.411. Den er ikke inkludert i SMTP, men finnes som en ESMTP tjenesteutvidelse (RFC 1891). Dersom en har ansvarlige tjenesteleverandører i alle ledd bør imidlertid denne tjenesten være overflødig, både for X.400 MTS og SMTP. Når en melding er overført til en ny transportagent (uten feilindikasjon), har operatøren av denne transportagenten overtatt ansvaret for å levere meldingen.

Både X.400 MTS og SMTP **skal** gi **negativ** leveringsbekreftelse (dvs. feilmelding) når en melding av en eller annen grunn ikke kan leveres[1]. Feil faller grovt sett i to kategorier, permanente feil som krever manuelt inngrep, og midlertidige feil som kan håndteres automatisk ved å sende meldingen som feilet om igjen senere. For automatisk feilhåndtering er standardiserte feilmeldingsformater og feilkoder essensielt. På dette punktet ser det ut til at X.400 og SMTP bygger på noe forskjellige antakelser. Når en SMTP-transportagent har overtatt en melding vil den vanligvis håndtere midlertidige feil selv, uten å rapportere dem tilbake til den opprinnelige avsenderen. Dermed er det stort sett bare de feilene som må håndteres manuelt som det overhodet er aktuelt å rapportere tilbake til avsender. X.400-transportagenter ser ut til å ha noe større tendens til å rapportere midlertidige feil tilbake til avsender, så automatisk feilhåndtering er mer aktuelt.

I tråd med dette, definerer SMTP selv ikke noe standard format for leveringsbekreftelser. RFC 1894 definerer leveringsbekreftelser (Delivery Status Notification, DSN) som en egen MIME meldingstype. ”Delivery Report” (DR) og ”Non-delivery Report” (NDR) er inkludert i X.400.

En X.400 DR kan forsterkes med ”proof of delivery” (dvs. slik at muligheten for forfalskede DR utelukkes). Internett DSN kan også benyttes til å rapportere om en forsinkelse, typisk pga. midlertidige feil eller en annen uventet eller unormal situasjon. Utover dette, er det ikke noen avgjørende forskjell på Internett DSN og X.400 DR/NDR for EDI-trafikk.

Merk også språkbruken, som faktisk er konsistent mellom X.400 og Internett: En DSN (på samme måte som en MDN, IPN og EDIN) betegnes ”notification”. Notifikasjonene er spesielle meldingstyper, og de fleste tjenestene (f.eks. sikkerhet) gjelder for notifikasjoner på samme måte som for andre meldingstyper. En DR (”report”) er ikke en melding, og leveres og presenteres på en annen måte.

2.5.7 Mottakskvittering

En mottakskvittering genereres av ansvarlig mottaker (dvs. brukeragenten) og indikerer på en eller annen måte om mottaker har blitt gjort kjent med meldingen.

[1] Det indikerer altså en feil i tjenesteleverandørs implementasjon av protokollene, ikke en svakhet i protokollen selv, dersom meldinger sporløst forsvinner ”i nettet”.

I motsetning til leveringsrapporter, blir mottakskvitteringer **bare** generert dersom avsender eksplisitt har bedt om det.

I X.400 har hver type av meldinger sine egne typer av (positive og negative) mottakskvitteringer. X.435 definerer ”EDI notifications” (EDIN). X.420 definerer ”Inter-personal notifications” (IPN). For Internett finnes en spesifikasjon for mottakskvitteringer, ”Delivery Status Notification” (DSN), som har en viss utbredelse. Forskjellen mellom de tre standardene ligger først og fremst i hva slags status de kan rapportere, dvs. nyanser i hva ”mottak” eller ”ikke-mottak” betyr: IPN er mest begrenset, og i EDI-sammenheng primært et alternativ til leveringsrapporter. EDIN tillater mer detaljert rapportering av feilsituasjoner som er aktuelle for EDI-trafikk, videresending m.m., og dessuten en sterkere form for mottakskvittering som inkluderer signatur over den mottatte meldingen. DSN dekker det samme som begge de to andre. Den er på enkelte punkter noe sterkere enn de to andre, bortsett fra at standarden ikke definerer like detaljerte feilkoder for ikke-mottak, og i stedet har større åpning for fritekst feilmeldinger.

Mottakskvitteringer har noen generelle problemer, og det at MDN ikke er like veletablert som mottakskvitteringer i X.400 er langt på vei på grunn av skepsis til mottakskvitteringer i seg selv.

- Muligheter for misbruk, i verste fall i ondsinnet hensikt. Mottakskvitteringer kan benyttes av uvedkommende til å kartlegge interne forhold hos mottakeren. De kan også benyttes til få en uskyldig tredjepart til å generere trafikk som f.eks. kommer inn gjennom en brannvegg der angriperen ellers ikke hadde adgang.
- Betydningen av ”mottak” er ikke veldefinert, og spesielt ikke i forbindelse med distribusjonslister og videresending. (Dette er først og fremst et problem i forbindelse med person-til-person-epost.)

MDN-spesifikasjonen tar opp disse problemene og løsninger på dem, noe X.400-spesifikasjonene ikke gjør.

Det er også verdt å merke seg at ”mottak” normalt ikke innebærer noen akseptanse av innhold, bare at mottaker bekrefter at meldingen er sett og tar ansvar for at den kan behandles videre. Uansett hvilken type mottakskvitteringer en benytter, må en regne med å presisere betydningen av dem nærmere i avtalen mellom partene.

Når EDI-anvendelsen også benytter EDIFACT-kvitteringsmeldinger er mottakskvittering (og leveringsbekreftelse) i praksisk betingede. Mottakskvittering genereres fra brukeragenten, før meldingen behandles på EDI-nivå. Selv om en positiv mottakskvittering er mottatt, vil det i praksis vanligvis være slik at mottakeren senere kan avvise meldingen med en negativ CONTRL, AUTACK, APERAK eller tilsvarende.

2.5.8 Transportnivå sikkerhet

Både X.400 og SMTP kan uten videre kjøre over et sikkert nett, f.eks. et lukket subnett bak en brannvegg eller et virtuelt privat nett. Sikkerheten kan også knyttes til transportforbindelser eller sesjoner mellom de to transportagentene, f.eks. ved bruk av SSL/TLS. For Internett håndteres slike mekanismer ganske uavhengig av SMTP-protokollen. Det vil være opp til hver enkelt transportagent å velge bruk av sikkerhetsmekanismer i lavere lag, basert på f.eks. meldingens adressat. I X.400 kan meldinger merkes med sikkerhetsklasse (”security label”),

slik at dette kan styre ruting av meldinger og bruk av underliggende sikkerhetsmekanismer. Definisjon og håndtering av sikkerhetsklasser er imidlertid opp til tjenesteleverandøren.

X.400 inkluderer også mekanismer for gjensidig autentisering av brukeragenter og transportagenter. For SMTP over SSL eller TLS utføres denne autentiseringen i SSL/TLS.

2.5.9 Meldingsnivå sikkerhet

De grunnleggende sikkerhetstjenestene på meldingsnivå er konfidensialitet, integritet, autentisitet og sporbarhet (ikke-benekting), i praksis kryptering og/eller digital signering av meldingsinnhold. I tillegg kommer sekvensintegritet, dvs. sikring mot tap, duplisering eller endring av rekkefølge av meldinger. Disse tjenestene er i sin natur ende til ende (dvs. realiseres av brukeragentene). I X.400 er mekanismer for å realisere disse tjenestene del av meldingens ”envelope” (selv om de altså ikke behandles av MTS).

For Internett epost benyttes en uavhengig *sikkerhetssyntaks*, i prinsippet et filformat for koding av signerte og krypterte data, og dette transporteres som en egen MIME innholdstype. Dette er også i prinsippet mulig i X.400 (”body part security”). Fordelen med å benytte en sikkerhetssyntaks på innholdsnivå, er at den er uavhengig av MHS, og kryptering og signaturer kan f.eks. påføres for å beskytte meldingene allerede før de overleveres til brukeragenten, eller beholdes etter at de er hentet ut av brukeragenten hos mottakeren. Dessuten kan sikkerhetssyntaks passere uforandret 0gjennom en MHS-nivå portner.

Sikkerhetsmessig og funksjonelt er det ikke vesentlige forskjeller på de to typene mekanismer. X.400 spesifiserer imidlertid ikke sikkerhetsmekanismene i detalj, og brukerne må avtale bl.a. hvilke kryptografiske algoritmer som skal benyttes. I praksis vil det trolig være problemer med interoperabilitet mellom X.400-implementasjoner fra ulike leverandører. De sikkerhetssyntaksene som er aktuelle i Internett-sammenheng er mer spesifikt definert, og angir bl.a. hvilke kryptografiske algoritmer som kan eller bør benyttes. Så spesifikke standarder er imidlertid ofte kontroversielle, og dette har nok vært med på å forsinke standardiseringen.

2.5.10 EDI-meldinger

X.435 definerer hvordan EDI-meldinger skal utveksles i X.400, men interimsløsninger basert på X.420 er i praksis vel så vanlige. For Internett foreligger utkast til spesifikasjoner (EDIINT) for transport av EDI-meldinger. (Disse har vi tatt for oss mer i detalj i hefte 8.) I motsetning til X.435, som er en separat standard og definerer en ny kategori av meldinger, er EDIINT en profil som spesifiserer hvordan de underliggende standardene (nevnt ovenfor) skal benyttes for EDI. EDIINT baserer seg i første rekke på MDN-spesifikasjonen og meldingssikkerhetsspesifikasjonene i Internett, som vi har sammenlignet med tilsvarende mekanismer i X.400 (X.435) ovenfor.

2.5.11 Sikkerhet og pålitelighet

Det er en vanlig oppfatning, særlig i EDI-kretser, at Internett epost er usikker og upålitelig, mens X.400 er relativt sikker og pålitelig. Dersom en sammenligner

Internett epost **generelt** med en **spesifikk** X.400-tjeneste er dette rett, men denne sammenligningen er misvisende. En stor del av forskjellen må tilskrives at Internettet er et åpent nett, mens X.400-tjenestene benytter lukkede transportnett. Det er ingen holdepunkter for at en X.400-tjeneste i samme skala som Internettet ikke ville ha tilsvarende problemer. Omvendt kan Internett-**protokollene** anvendes i lukkede nett Vi diskuterer avveiningen mellom åpne og lukkede nett nærmere i kapittel 3.2 og 4.2.

Sikkerhet i relativt åpne nett må basere seg på ende-til-ende-mekanismer, dvs. kryptering og digital signatur fra avsender til mottaker. For elektronisk post betyr dette i alminnelighet at disse mekanismene innpasses på meldingsnivå. Som diskutert ovenfor, er det ingen vesentlige sikkerhetsmessige eller funksjonelle forskjeller på disse punktene i X.400 og Internett epost. Dessverre er ende-til-ende-sikkerhet ennå lite utbredt, og i liten grad støttet av eksisterende programvare, og her ligger kanskje X.400-programvare enda dårligere an.

Konfidensialitet av meldinger baserer seg i dag i praksis på at transportnettet vanskelig kan avlyttes, og på tillit til operatørene av transportagenter. Det er ingen mekanismer i meldingstransporttjenesten verken i X.400 eller Internett som har noen betydning på dette punktet. Risikoen for avlytting av trafikk i Internett er absolutt til stede, men blir av og til overdrevet.

Integritet og *autentisitet* av meldinger baserer seg først og fremst på tillit til transportagentene, dvs. at enhver transportagent som mottar en melding fra en bruker eller en annen transportagent har tatt alle nødvendige skritt for å autentisere denne og sikre integritet under overføringen. Det er på dette punktet de største svakhetene i Internettet finnes: For det første har SMTP-tjener begrenset mulighet til å autentisere klienten, dvs. en bruker eller transportagent på en annen vertsmaskin. Dette må i tilfelle skje ved IP-adresse, som er mulig å forfalske. (SMTP over TLS/SSL gir imidlertid sikker autentisering.) Men dessuten vil mange av de transportagentene som er i bruk ikke en gang utføre den autentiseringen som det faktisk er rimelig og mulig å gjennomføre[1]. Og selv om de gjør det, filtreres de feltene i meldingshodet som gir denne informasjonen ofte bort, pga. misforstått "brukervennlighet", i brukeragenter eller portnere inn i lokale epost-systemer. Det er altså vesentlig rom for en del forbedring på dette punktet, og vi anbefaler at EDI-brukere er oppmerksom på dette og stiller krav til leverandører av tjenester og programvare.

Det mer grunnleggende problemet med å autentisere hvert enkelt ledd i kjeden på denne måten, er at situasjonen blir svært uoversiktlig i større skala. For X.400 vil situasjonen da normalt være at meldingene rutes i transitt gjennom flere operatører, som brukerne har et mer perifert forhold til. Ens egen tjenesteleverandør er antakelig ikke villig til å ta på seg det samme ansvaret som for trafikk innenfor eget nett.

X.400-spesifikasjonen er nok noe mer eksplisitt på at denne autentiseringen skal utføres, og en kan håpe på at programvaren som benyttes faktisk gjør det i større grad enn for Internett, men protokollene selv har ikke noen avgjørende forskjeller på dette punktet heller.

[1] Jfr. problemet med at mange SMTP-tjenere lar seg utnytte som "spam relay", dvs. for å tilsløre opphavet til masseutsendelser av reklame eller annen uønsket epost. Dette er i prinsippet enkelt å unngå, siden det vanligvis ikke er noen legitim grunn til å videresende meldinger som ikke stammer fra eget nett, men problemet er likevel utbredt pga. en kombinasjon av at programvaren installeres med uheldige konfigurasjonsvalg fra leverandørens side, og driftspersonell har for dårlig tid eller kompetanse til å oppdage dette.

Kapasitet, pålitelighet og *tilgjengelighet* avhenger først og fremst av transportnett og kvaliteten av programvare og utstyr. Alle åpne nett vil være utsatte for tjenestenektingsangrep, eller bare overbelastning pga. tilfeldige variasjoner i trafikken, feil eller uheldige sammentreff. Slike problemer er kroniske i Internettet, men dette er snarere et problem med nettets skala enn med protokollene selv. Manglende utbredelse av kvitteringsmekanismer blir oppfattet som et problem med påliteligheten i Internett epost. Som nevnt skal feilmeldinger ubetinget returneres når meldinger ikke kan leveres (og altså slettes underveis i nettet). Når det likevel ikke skjer, er dette en feil i implementasjonen eller drift/konfigurasjon av transportagenter eller brukeragenter.

I mange tilfeller skyldes problemer med bortkomne meldinger trolig portnere inn i bedriftsinterne epostsystemer basert på andre protokoller og formater, som transformerer meldinger eller feilmeldinger til det ugjenkjennelige. Resultatet kan da bli at feilmeldinger riktignok blir generert, men ikke er mulig å returnere til den opprinnelige avsenderen.

2.5.12 Konklusjoner

Selv om det er en del tekniske forskjeller mellom protokollene, er disse i seg selv neppe utslagsgivende for hva en bør velge. Tilgjengelighet av produkter og tjenester som oppfyller kravene en vil stille til meldingsutvekslingstjenesten er det vesentlige. En må også være oppmerksom på at standardene ofte bare er delvis implementert, slik at krav som standardene i og for seg oppfyller ikke behøver å være oppfylt av konkrete produkter eller tjenester.

På en del vesentlige punkter for EDI, spesielt mottakskvittering og sikkerhetsmekanismer, er standardiseringsprosessen på Internett-siden formelt sett ikke kommet lenger enn til foreslåtte standarder (proposed standard). Tilsvarende X.400-standarder er offisielle standarder. Vi vurderer likevel ikke risikoen ved å basere seg på disse Internett-spesifikasjonene som særlig stor i forhold til X.400-standardene. Det finnes allerede et rimelig utvalg av programvareprodukter som implementerer dem, og som er demonstrert interoperable. (Internett-spesifikasjoner kan ikke bli standarder uten praktisk uttesting med flere uavhengige og interoperable implementasjoner. Derfor er de normalt klare til praktisk bruk før de når status som standard. For X.400 foreligger gjerne standardene før praktiske implementasjoner av dem er tilgjengelig.)

3 Transportnettet

Egenskapene ved meldingsutvekslingstjenesten, spesielt ikke-funksjonelle som pålitelighet, tilgjengelighet og sikkerhet, avhenger vel så mye av hvilke konkrete nett og underliggende transporttjenester den er basert på som hvilke protokoller og standarder som benyttes.

Transportnettet kan karakteriseres som mer eller mindre åpent. Et åpent nett er tilgjengelig for hvem som helst, med få eller ingen restriksjoner på hvem som har anledning til å knytte seg til det. En vesentlig forutsetning for et åpent nett er at det baserer seg på standardiserte protokoller og utstyr og programvare som i praksis er tilgjengelig, men dette i seg selv gjør ikke nettet åpent.

Internettet er i ekstrem grad et åpent nett, men Internett-protokoller kan bli og blir også benyttet i lukkede nett. Legg også merke til at en bruker normalt er knyttet til Internettet via en tjenesteleverandør, og tjenesteleverandøren står fritt til å regulere adgangen til sitt eget nett, både hvem som får knytte seg direkte til det og til en viss grad hvilken trafikk som kan passere ut og inn av det til og fra egne brukere eller andre deler av Internettet (brannvegger).

Omvendt er X.400-protokollene utviklet for bruk i åpne nett, men EDI-formidlingssentraler basert på X.400 i praksis opererer lukkede nett der adgangen er begrenset til kunder av formidlingssentralen (eller andre formidlingssentraler der forholdet er regulert gjennom samtrafikkavtaler).

Valg av transportnett for en EDI-løsning er en avveining av flere hensyn.

- Internettet har karakter av allmenn infrastruktur. I de fleste tilfeller er både organisasjonen selv og de partene en ønsker å kommunisere med allerede tilknyttet nettet, eller kostnaden for tilknytning er marginal. Samme tilknytning vil kunne benyttes for flere EDI-anvendelser.
- Lukkede nett gir vanligvis mer forutsigbar tjenestekvalitet og mindre utfordringer rundt sikkerhet, pålitelighet, tilgjengelighet og kapasitet. Med spesielle krav på disse områdene, og forutsatt at hver av partene har en nytte av EDI-anvendelsen som står i forhold til kostnadene deres, kan det forsvares å bygge opp et privat nett (f.eks. basert på faste samband) for én EDI-anvendelse.
- Formidlingssentraler står i en mellomstilling mellom helt åpne og helt lukkede nett, avhengig av hvordan tjenesten er utformet og hvordan den drives. Ved siden av de fordelene som et relativt lukket nett kan gi, vil dessuten formidlingssentralene ofte tilby EDI-relaterte tjenester, som konvertering, tidsstempling, notarisering. Disse tjenestene kan imidlertid godt løsrives fra den grunnleggende nettverkstjenesten.

3.1 IP-protokollen

Fellesnevneren for Internettet er IP-protokollen. All trafikk brytes ned til IP-pakker, og basisfunksjonen til Internettet er bare det å transportere IP-pakker. IP-protokollen skjuler på sin side den konkrete nettverksteknologien som benyttes for å transportere pakkene. Det er mulig å kjøre IP over praktisk talt all aktuell nettverksteknologi.

Av og til blir forskjellige nyere nettverksteknologier (spesielt: forskjellige typer av høyhastighetsnett) presentert som alternativer til Internett. Det er nokså misvisende: Så lenge det lar seg gjøre å kjøre IP over slike nett (og det gjør det i de fleste tilfellene) kan nett av disse typene godt innlemmes i Internettet eller bedriftsinterne IP-nett. En EDI-løsning eller andre anvendelser som bruker vanlige Internett-tjenester som elektronisk post, Web, filoverføring osv. vil i utgangspunktet ikke merke noen forskjell (annet enn bedre ytelse o.l.)

IP og IP-baserte protokoller vil ikke alltid være i stand til å utnytte spesielle egenskaper ved disse nettverksteknologiene. Dette kommer inn i bildet i forbindelse med tjenester med spesielle krav til båndbredde, sanntidsegenskaper eller gradering av tjenestekvalitet. Den nye versjonen av IP-protokollen (IPv6) har forbedringer på en del av disse punktene i forhold til nåværende versjon (IPv4). Spesielt kan nevnes mekanismer for å håndtere tjenestekvalitet.

På disse områdene har EDI likevel ganske moderate krav, i forhold til f.eks. sanntids multimedia, og dette i seg selv gir neppe vesentlige argumenter mot en IPv4-basert kommunikasjonsløsning for EDI.

I forbindelse med IPv6 hører en av og til den noe misvisende betegnelsen ”Internett 2”. De to versjonene av IP er ikke interoperable, og i den den forstand blir IPv4- og IPv6-internettene to separate ”virtuelle” nett. Men de blir ikke fysisk separate: IPv4 og IPv6 kan brukes parallelt i samme nett og på samme maskiner. Etter hvert som IPv6 blir mer aktuelt, må vi anta at tjenesteleverandørene setter opp tjenestetilbud basert på IPv6, mens IPv4 gradvis fases ut.

Sett fra transportagenter og brukeragenter vil denne overgangen foregå forholdsvis transparent. Transportprotokollene (TCP, UDP) og protokoller på høyere nivå er de samme. På en maskin der både IPv4 og IPv6 støttes, er det typen adresse som bestemmer hvilken IP-versjon som faktisk brukes. Internett-tjener- og klientprogrammer kan ganske enkelt oppgraderes til å støtte begge typer adresser. EDI-programvare som bare forholder seg til RFC822-adresser, URL-er og andre adresseformat der maskinnavn inngår og ikke bruker IP-adresser noe sted berøres ikke.

Behovet for å ta i bruk IPv6 kommer først når en trenger tilgang til tjenestetilbud som bare er tilgjengelige i IPv6. Det mest åpenbare tilfellet er at tjenesten i seg selv krever funksjonalitet som mangler i IPv4. Etter hvert som IPv6 blir mer utbredt kan det nok også tenkes at tjenesteleverandører slutter å støtte enkelte tjenester på både IPv4 og IPv6 (dvs. kjøre tjenerne på maskiner med både IPv4 og IPv6-protokollstakk) selv om tjenestene i seg selv kan støttes på IPv4. Eventuelt er tjenestene tilgjengelige på begge IP-versjoner, men IPv6-versjonen gir på en eller annen måte bedre kvalitet.

For å bruke IPv6 trengs følgende:

- En tjenesteleverandør som tilbyr IPv6-aksess.
- Oppgradering av (programvare på) rutere og annet utstyr i lokalt nett til å rute IPv6.
- Installasjon av en IPv6-protokollmodul i vertsmaskinene som skal kjøre IPv6. (Sannsynligvis et ledd i en normal operativsystem-oppgradering på et eller annet tidspunkt.)
- Oppgradering av klient- og tjenerprogrammer til å håndtere IPv6-adresseformat.

Denne oppgraderingen behøver ikke berøre EDI-spesifikk programvare i løsningen i det hele tatt. (Med unntak av det siste, dersom brukeragent/transportagent-funksjonalitet er tett innvevd i EDI-programvaren.)

3.2 Ulike typer lukkede nett

Lukkede nett kan realiseres på forskjellige måter:

- Fysisk lukkede nett, dvs. hvor uvedkommende fysisk ikke har adgang til maskinvare og kabling. (Dette er i praksis bare realiserbart for lokalnett.)
- Leide, faste samband.
- Oppringte samband som ISDN, X.25 eller X.21 med lukket brukergruppe.
- ATM, frame relay og andre nyere nett-teknologier kan brukes tilsvarende forrige punkt og gir et utvalg av forskjellige grader av tjenestekvalitet.
- Virtuelle private nett, der trafikken bæres av et åpent nett, men beskyttes ved kryptografiske mekanismer (kryptering og autentisering) mellom noder i det virtuelle nettet.

Alle mekanismene kan gi tilstrekkelig sikkerhet for majoriteten av anvendelser, selv om sikkerhetsegenskapene varierer noe. Virtuelle private nett gir noen av fordelene med åpne nett. (Nye deltakere kan benytte tilknytninger de allerede har, og de trenger i beste fall bare ny programvare eller omkonfigurering av portnere.) Tjenestekvaliteten arver virtuelle private nett imidlertid fra det underliggende åpne nettet, med de problemer det måtte ha.

Lukkede nett er heller ikke nødvendigvis avskåret fra omverdenen. De kan ha samtrafikk med andre nett, men regulert på en eller annen måte, typisk gjennom en eller annen form for portner- eller brannveggløsning.

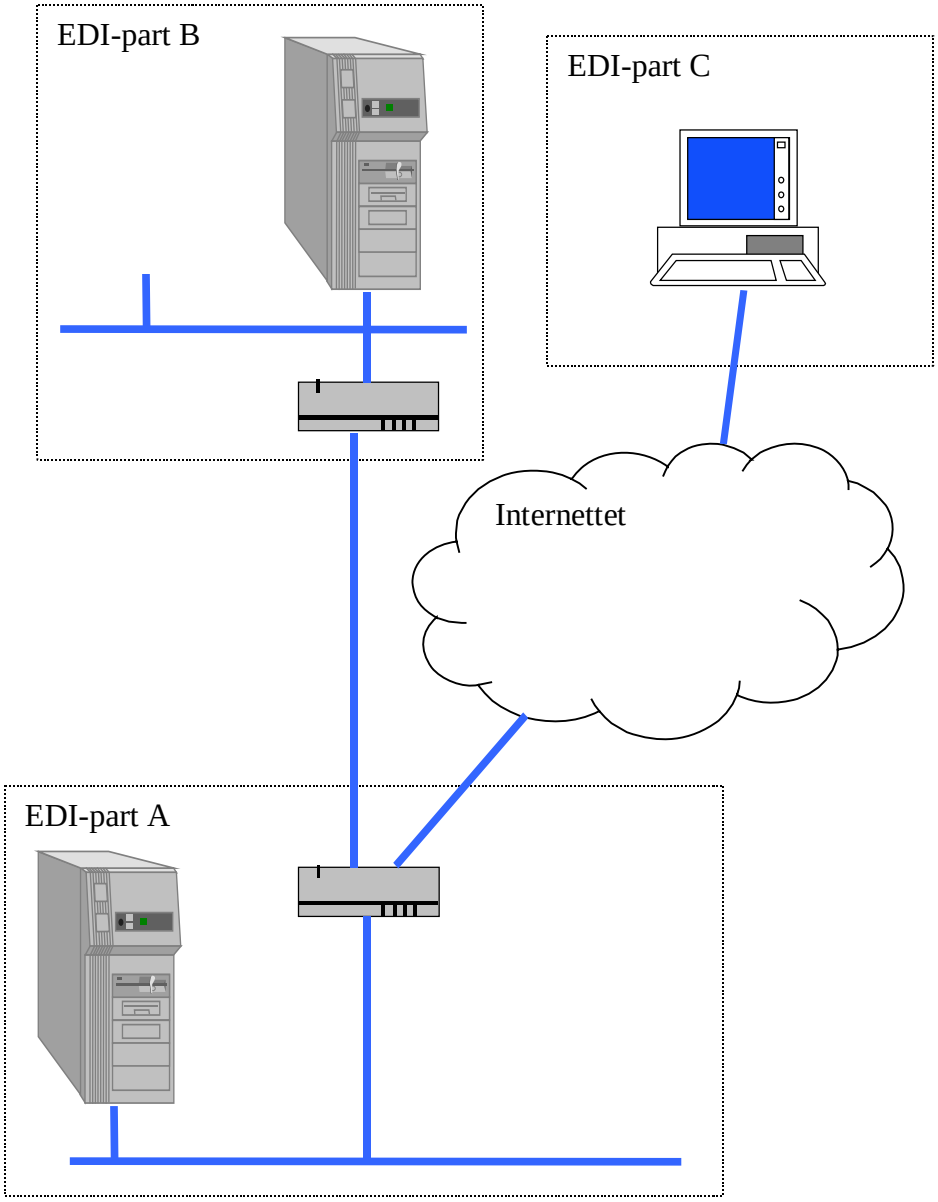
3.3 Intra- og ekstranett

Vi har beskrevet Internettet som et bestemt åpent nett. Det er egentlig en overforenkling. Et internett er en samling av innbyrdes sammenknyttede nett. Disse enkeltnettene er relativt autonome og kan godt være mer eller mindre lukkede. En samling av X.400-operatører (management domains) med samtrafikk er eksempel på et internett, det verdensomspennende telefonnettet likeså. Hvilken tjenestekvalitet en faktisk får i et internett, og Internettet spesielt, avhenger av hvilket av del-nettene en er tilknyttet, dvs. hvilken Internett-tjenesteleverandør (ISP) en benytter. Tjenestekvaliteten avhenger bl.a. av kapasiteten i ISPens eget stamnett, hvilke direkte samtrafikkmuligheter som finnes mot andre ISPer og kapasiteten på dem, brannvegger som skjermer ISPens nett og kunder, tilrettelegging og assistanse for kundene. ISPer har til nå konkurrert først og fremst på pris. Med profesjonelle kunder som etterspør tjenestekvalitet og er villige til å betale for det, må vi anta at dette kommer til å endre seg.

Også bedriftsinterne nett knyttet til Internettet gjennom brannvegger er i en viss forstand en del av Internettet. En må ikke nødvendigvis ofre de fordelene en Internett-tilknytning gir selv om EDI-anvendelsen krever bedre tjenestekvalitet eller sikkerhet enn det Internettet i seg selv gir. Dersom en f.eks. i én EDI-anvendelse både skal dekke store brukere med stor trafikkmengde og store krav til tilgjengelighet og mindre brukere med relativt sporadisk trafikk, kan en god løsning være å dekke de mindre brukerne gjennom en generell Internett-tilknytning, men komplettere dette med egne samband til de større brukerne. En

slik nettløsning kan bygges ut nokså fleksibelt og trolig uten at det påvirker høyere lag i EDI-løsningen.

Figur 4 viser ett eksempel på en slik nettløsning. Bedriften A har EDI-trafikk både mot den store bedriften B og mindre kunder, som C. For kommunikasjonen mellom A og B er det leid et fast samband som er koblet opp til ruterne inn til deres interne nett. For de små kundene er ikke en slik løsning regningssvarende, og trafikken ledes i stedet via Internettet.



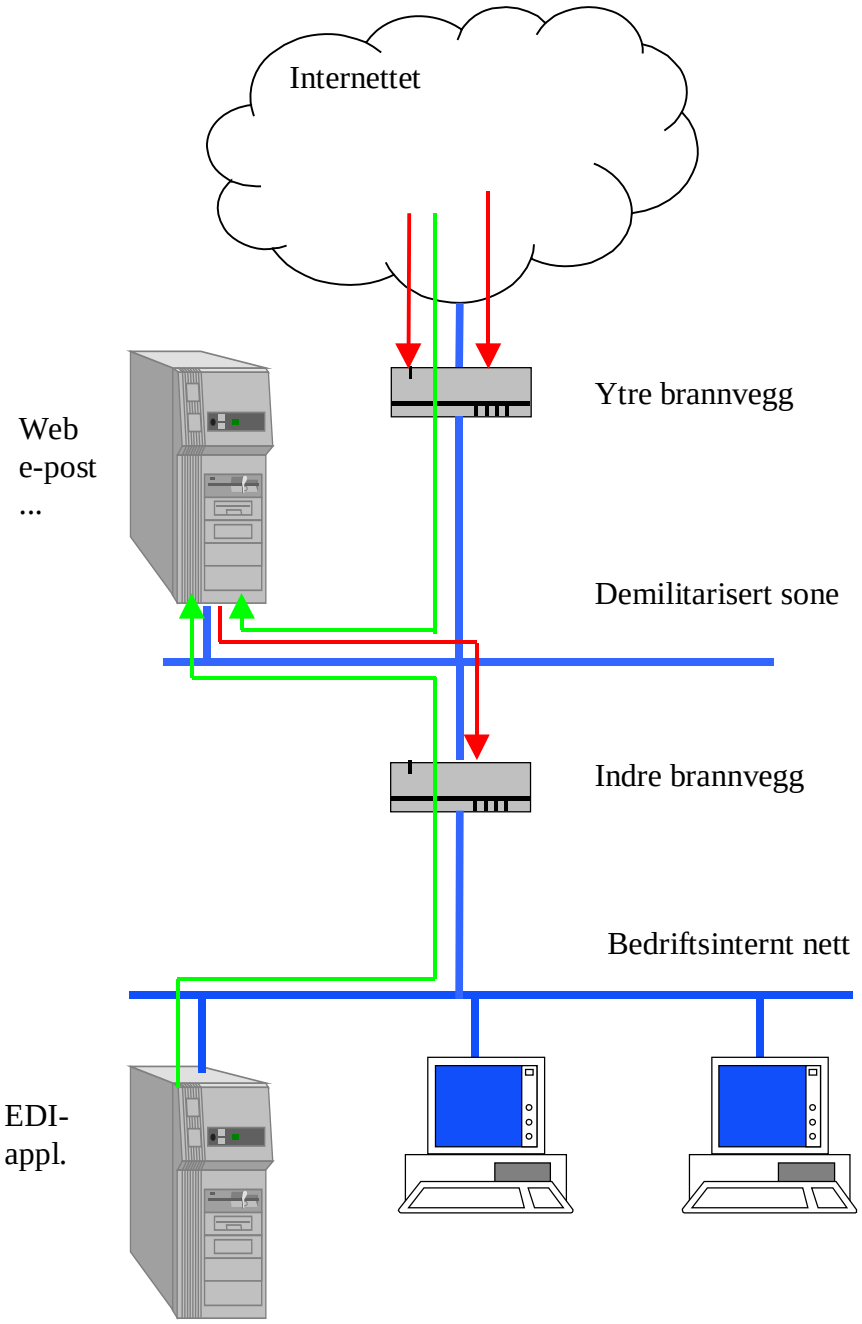
Figur 4 Kombinert nettløsning

Når Internett-teknologi benyttes i mer eller mindre lukkede nett, benyttes gjerne begrepene *intranett*[1] og *ekstranett*. Et intranett binder sammen enheter innenfor en organisasjon og består typisk av en samling lokalnett som er sammenknyttet gjennom faste samband. Et ekstranett binder sammen separate organisasjoner,

[1] Intranett benyttes ofte i en mer snever forstand om en bedrifts interne Web-løsning.

som f.eks. en bedrift og dens leverandører eller kunder og er typisk bygget opp som et virtuelt privat nett (VPN). Internett-baserte EDI-formidlingssentraler har gjerne en ekstranett-type løsning.

3.4 Brannvegger



Figur 5 Brannveggløsning med demilitarisert sone

Formålet med en brannvegg er å knytte et mer lukket (sikkerhetskritisk) nett sammen med et mer åpent (usikkert) nett på en kontrollert måte, altså hindre at et angrep som stammer fra det usikre nettet forplanter seg til den andre siden av brannveggen. Det mest vanlige tilfellet er å bruke en brannvegg for å knytte en bedrifts interne nett til Internettet. Men det er også aktuelt å bruke brannvegger f.eks. internt i en større bedrift for å skille mer sensitive avdelingsnett fra

bedriftens stamnett. Når en bedrift opererer tjenestemaskiner, f.eks. Web- eller epost-tjenere, er det vanlig å plassere disse i en såkalt ”demilitarisert sone” mellom en ytre og en indre brannvegg[1]. Siden disse maskinene er direkte tilgjengelige utenfra, er de mer utsatt for angrep, og en kan derfor ikke ha helt den samme tillliten til dem som maskiner innenfor den indre brannveggen.

Poenget med å bruke en brannvegg er å gi nett-ansvarlig ett kontrollpunkt, slik at det er mulig å kontrollere at en sikkerhetspolicy blir overholdt uten å holde detaljert og løpende kontroll over alle maskiner og all programvare internt. F.eks. kan risikable nett-tjenester som bare er beregnet for bruk i lokalnett blokkeres i brannveggen, i stedet for at en er avhengig av at disse er avslått (eller har passende adgangskontroll implementert og konfigurert) på hver enkelt maskin innenfor.

Det andre poenget er at brannvegg-maskinen normalt skal være dedisert til denne oppgaven. Den kan kjøre et mer spesialisert, enklere operativsystem, har få eller ingen nettverkstjenester tilgjengelig, kjører ikke applikasjonsprogramvare, kontrolleres bare gjennom egne fysiske konsollterminaler osv. Derfor er det overkommelig å verifisere at brannvegg-maskinen gir akkurat de adgangsbegrensningene den skal, og ikke minst at det ikke lar seg gjøre å angripe brannveggen selv og åpne for utilsiktet adgang. For en vanlig arbeidsstasjon eller tjenestemaskin, som kjører et generelt operativsystem og mange applikasjonsprogrammer for forskjellige brukere, kan man vanskelig ha samme grad av tillit. Brannveggen bør være slik at den har ”obviously no faults”, ikke ”no obvious faults” — det siste er ofte tilfellet med generelle maskiner...

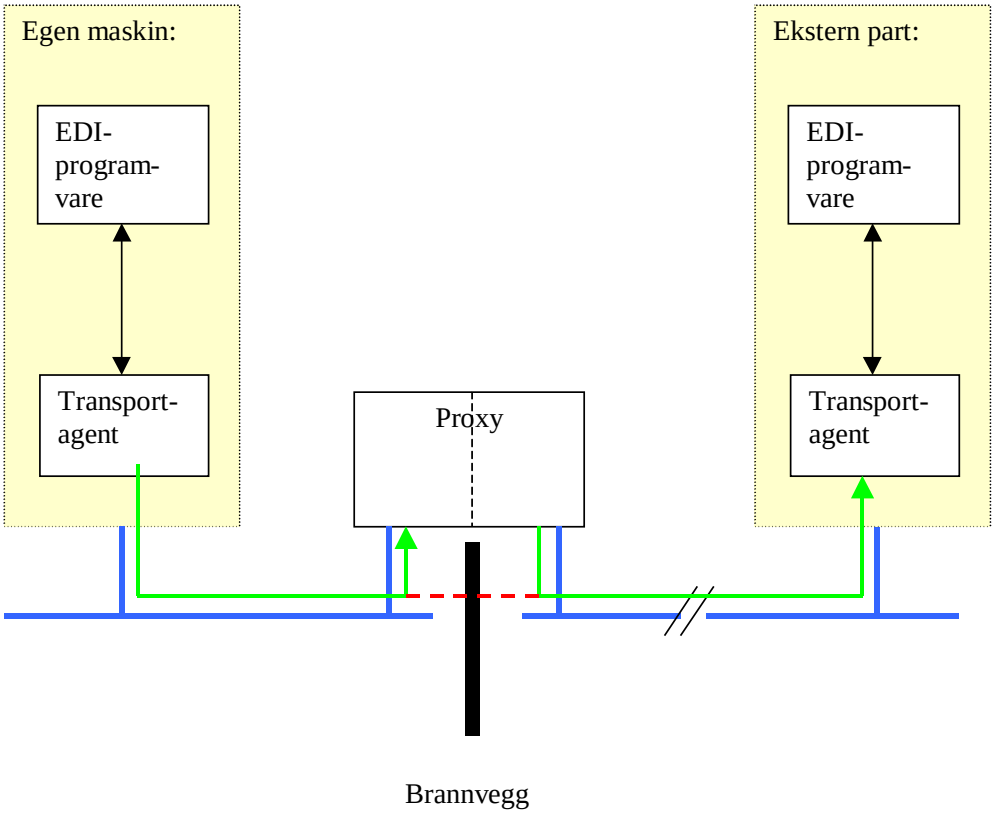
Det finnes to hovedtyper av brannvegger:

- Pakke-filtre, som undersøker innholdet i datapakkene som skal fra det ene nettet til det andre, og avgjør om pakken skal blokkeres eller slippes gjennom. Typisk bruk av pakkefiltre er å begrense trafikken på bakgrunn av avsenders eller mottakers IP-adresse samt portnumre, slik at ekstern trafikk til interne maskiner er begrenset til bestemte maskiner og tjenester.
- Applikasjons-portnere (proxy[2]), som er tjenere for en bestemt tjeneste, f.eks. SMTP, HTTP eller FTP. Den egentlige klienten kontakter proxy-tjeneren, som videreformidler trafikken til den egentlige tjeneren etter å ha utført de sikkerhetskontrollene den skal. Proxy kan benyttes både for innkommende og utgående trafikk. F.eks. kan en gi adgang til en intern Web-tjener gjennom en proxy som sørger for at eksterne klienter bare har adgang til en delmengde av sidene. Omvendt en HTTP-proxy brukes for å kontrollere at en intern Web-klient ikke mottar Web-sider med virusinfisert innhold.

I praksis brukes gjerne en kombinasjon av disse teknikkene. Pakke-filtre var i utgangspunktet tilstandsløse, dvs. reglene for om en viss pakke skulle slippe gjennom eller ikke tar bare hensyn til innholdet i pakken selv. Men nyere pakke-filtre lagrer vanligvis en del informasjon på høyere protokollnivå (stateful inspection), og svarer i praksis mer til proxy-er.

[1] I figuren har vi tegnet to separate brannvegg-maskiner for å få frem den logiske strukturen. Det er imidlertid mulig å realisere brannveggen i én maskin som er tilknyttet både det eksterne nettet, den demilitariserte sonen og det interne nettet.

[2] ”Proxy” betyr ”stedfortreder”. Noe godt norsk ord er ikke etablert. Ordet henspeiler på at proxyen både opptrer som stedfortreder for den egentlige tjeneren, sett fra klientens synspunkt, og som stedfortreder for den egentlige klienten, sett fra tjeneren.



Figur 6 Kontrollert aksess gjennom brannvegg via proxy

Figur 6 viser hvordan en proxy-transportagent[1] brukes for å transportere trafikk mellom det lokale nettet og et eksternt nett. Her er brannveggen tegnet som et fullstendig skille på nettverkslaget, og proxyen er eneste vei ut og inn. Den interne transportagenten har satt opp en forbindelse med den ene siden (tjenersiden) av proxyen og sender kommandoer til denne som om den var den eksterne transportagententen. Forbindelsen til den eksterne transportagenten kommer fra proxyen. Proxyen foretar sikkerhetskontroll av kommandoer og responser, og vil i noen tilfeller også modifisere data og kommandoer, før den videresender trafikken ut eller inn av nettet. Men de to transportagentene som er endepunktene for trafikken, oppfører seg som om de kommuniserte direkte, som den stiplede linjen gjennom veggen antyder. (I praksis vil det være noen ekstra elementer i protokollen slik at klienten, når den kontakter proxyen, kan fortelle hvilken ekstern maskin proxyen skal koble seg mot.)

Brannveggprodukter er alltid konfigurerbare, dvs. at de styres av et sett med regler som bestemmer hvilken trafikk som skal tillates. Denne konfigurasjonen må tilpasses hver enkelt installasjon og vil ofte være et kompromiss mellom den sikkerheten en skal ha og de tjenestene og applikasjonene som skal brukes gjennom brannveggen. Brannvegger er ikke ”pakkeløsninger”. En kan ikke regne med å kjøpe et standard brannveggprodukt hos en leverandør, få det installert av leverandørens konsulenter og deretter ha alt på det tørre sikkerhetsmessig. Bedriften må selv ha en gjennomtenkt sikkerhetspolicy og utforme

[1] Siden SMTP og andre epost-protokoller i sin natur tillater ”store and forward” er det ikke så aktuelt å bruke proxyer med disse. Men for FTP, HTTP og andre synkrone protokoller er det meget aktuelt å bruke proxy.

brannveggløsning ut fra den. Når brannveggen er satt i drift, må en regne med at den krever overvåkning og vedlikehold som svar på nye krav og trusler.

For EDI-løsninger, med faste kommunikasjonspartnere og –mønstre, er det naturlig å sette opp strikte brannveggkonfigurasjoner, som kan gi svært god sikkerhet. Bare en håndfull tjenester trenger å være tilgjengelige (epost, samt navnetjeneste etc. som er nødvendig for at epost skal fungere), og bare for kjente parter. All trafikk som ikke eksplisitt er tillatt blokkeres. Tjenestene kan kanskje til og med være åpne bare til visse tider (f.eks. når det er driftspersonell på vakt). Men så "tette" brannvegger må ligge nær EDI-løsningen selv. Den brannveggen som regulerer bedriftens generelle adgang til Internettet må antakelig være mindre restriktiv i visse henseende. Dermed kommer raskt behovet for å dele opp det interne nettet i flere nett med forskjellig sikkerhetspolicy.

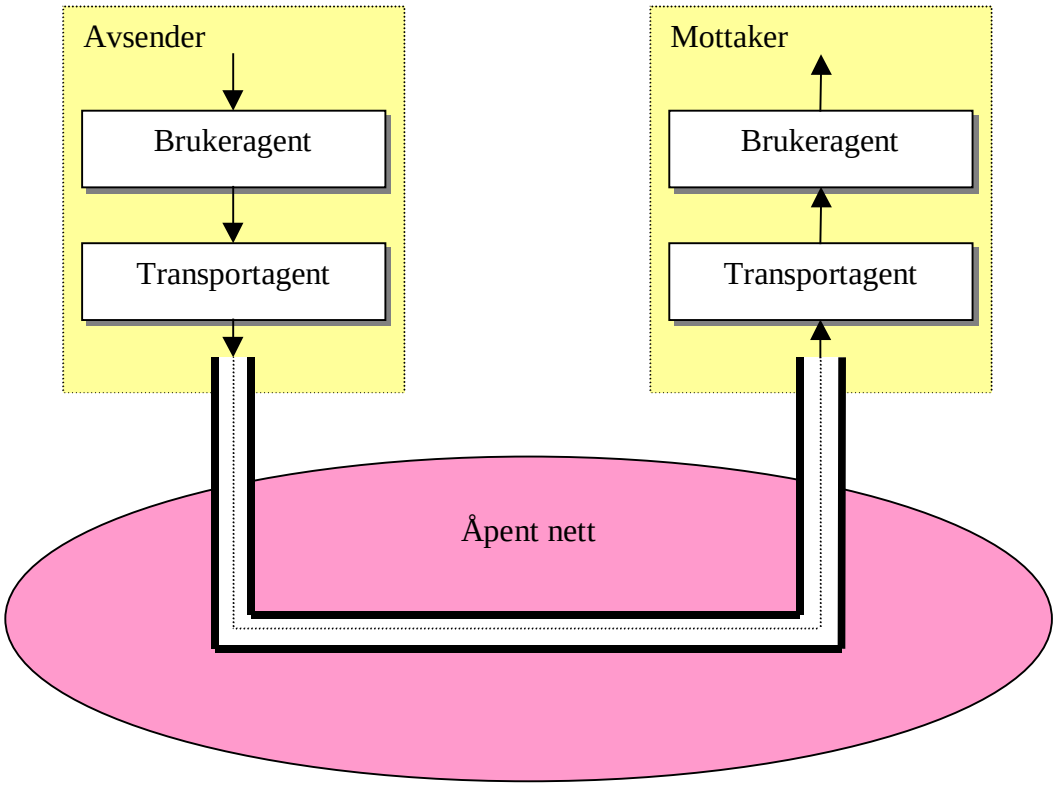
3.5 Transportlagssikkerhet

Transportlagssikkerhet betyr at sikkerhetsmekanismene plasseres umiddelbart over transportlaget, dvs. mellom transportnett og transportagent i vår EDI-arkitekturskisse. Transportlagssikkerhet beskytter data på hver enkelt transportforbindelse, der spesifikke brukere/programmer har kontroll over endepunktene. M.a.o., data som utveksles på en transportforbindelse er beskyttet både mot eksterne trusler og andre brukere/programmer på sendende eller mottakende maskin.

I internett-sammenheng benyttes protokollene SSL- (Secure Socket Layer) og på sikt TLS- (Transport Level Security). TLS er i praksis ikke noe annet enn en ny versjon av SSL, men det er TLS-spesifikasjonen som kommer til å bli en offisiell Internett-standard. For enkelhets skyld skriver vi TLS nedenfor, men alt som står gjelder både SSL og TLS. Produkter basert på TLS og SSL vil i hovedtrekk være interoperable.

Transportlaget i internett er TCP. En klient (f.eks. en sendende transportagent) kommuniserer med tjeneren (f.eks. mottakende transportagent, dvs. SMTP-tjener) ved å åpne en TCP-forbindelse til en bestemt **port** på tjener-maskinen, hvor tjeneren venter på innkommende forbindelser. Når forbindelsen er åpen, har klient og tjener en toveis, full duplex, feilfri[1] forbindelse. TLS gir en transportforbindelse med samme egenskaper, dvs. TCP-baserte tjenester kan stort sett flyttes til TLS uten vesentlige endringer, verken i spesifikasjonen av tjenesten eller implementasjonen av klient- og tjenerprogramvare. Så også når det gjelder SMTP, POP3, IMAP4, HTTP og FTP som er aktuelle i EDI-sammenheng, alle disse protokollene finnes i TLS-baserte versjoner (--).

[1] Feilfri i den forstand at kommunikasjonsfeil på lavere nivå rettes ved at IP-pakker retransmitteres, og at avsender og mottaker kan gå ut fra at data mottas som sendt så lenge TCP-implementasjonen ikke rapporterer feil.



Figur 7 Transportlagssikkerhet mellom transportagentene

I tillegg gir TLS følgende tjenester:

- Konfidensialitet, dvs. kryptering av utvekslede data.
- Autentisering av tjener.
- Valgfritt, autentisering av klient.

TLS-tjeneren må ha et X.509-sertifikat som klienten aksepterer. Tilsvarende for klienten dersom klient-autentisering benyttes. Når klienten oppretter en forbindelse til tjeneren, vil de først utveksle (og verifisere) sertifikater. Så genereres[1] hemmelige sesjonsnøkler som benyttes for å kryptere og autentisere (signere) data som utveksles i begge retninger.

TLS/SSL kan brukes med et utvalg av forskjellige kryptografiske algoritmer. Den implementasjonen av SSL som er utbredt gjennom Netscapes og Microsofts nettlesere i "eksportversjon" tilbyr bare et begrenset utvalg på grunn av amerikanske eksportrestriksjoner. Autentisering og nøkkelutveksling skjer ved 512-biters RSA-nøkler, og kryptering av data med 40-biters RC4- eller RC2-nøkler. Dette gir ikke tilstrekkelig sikkerhet. Problemet ligger imidlertid ikke i SSL-protokollen selv, og TLS/SLL-implementasjoner av god kvalitet og som tilbyr sterke kryptografiske algoritmer kan regnes for å være meget sikre. (128-biters symmetriske og 1024-biters eller større asymmetriske nøkler.)

[1] Klienten genererer en tilfeldig bitstreng, en såkalt "pre-master secret", som sendes til tjeneren kryptert med tjenerens offentlige nøkkel i følge sertifikatet som tjeneren returnerte da forbindelsen ble satt opp. Tjeneren — men ikke angripere som avlytter nettet — har den private nøkkelen og kan dekryptere bitstrengen. Deretter bruker begge parter en fast algoritme for å avlede nøklene som trengs fra denne felles hemmeligheten.

Sikkerhetsegenskapene ved en TLS-basert løsning avhenger av hvordan den er integrert med klient- og tjenerprogramvaren. Det er først og fremst to kritiske operasjoner her:

- Verifikasjon av sertifikater, dvs. hvordan klient og evt. tjener avgjør at den andre partens sertifikat er gyldig, om sertifiseringsautoritet og -policy er akseptabel, og dessuten om denne parten aksepterer å kommunisere med den andre parten overhodet.
- Operasjoner med private nøkler, når den ene parten skal autentisere seg overfor den andre.

For TLS er det vanlig at disse operasjonene skjer implisitt inne i klient- hhv. tjenerprogrammet. Den første operasjonen skjer typisk ved at klient og tjener har tabeller over akseptable sertifikater, som angir i hvor stor grad denne klienten eller tjeneren har tillit til eieren av sertifikatet. Men det er også mulig bare å verifisere sertifikatet og gjøre det tilgjengelig på høyere nivå, f.eks. for EDI-programvaren slik at den kan kontrollere innholdet i sertifikatet mot EDI-avsenderen i EDIFACT-utvekslingen.

Når operasjoner med private nøkler skjer implisitt, betyr det at den autentiserte parten faktisk er en applikasjon eller tjener, ikke en person. Eller sagt på en annen måte: Det som er autentisert med en privat nøkkel, kan stamme fra hvem som helst som har kontroll over eller kan påvirke operasjoner med den private nøkkelen. (Og omvendt for konfidensialitet: Det som er kryptert, må en anta er tilgjengelig for alle disse.) Avhengig av hvordan den interne sikkerheten er på maskinen der den private nøkkelen blir brukt, kan denne gruppen være temmelig stor: Andre brukere med adgang til maskin eller nett, driftspersonell osv.

Dette er en viktig forskjell fra digital signatur, hvor det er et krav om at signeringsoperasjonen skal eksplisitt utløses (ved PIN-kode eller lignende) av den personen som faktisk signerer. Betydningen av en autentisering står og faller på hvordan den private nøkkelen er beskyttet. (Dette vil vanligvis være spesifisert gjennom sertifiseringspolicy som er angitt i sertifikatet.)

Når da TLS brukes for å sikre kommunikasjonen mellom to transportagenter, er det **transportagentene** som autentiserer seg overfor hverandre. Dette er det viktig å være klar over hvis transportagent og EDI-aksesspunkt/applikasjon kjører på forskjellige maskiner, i forskjellige sikkerhetsdomener, eller transportagenter hos eksterne tjenesteleverandører benyttes. Ønsker en at det faktisk er EDI-aksesspunkt eller applikasjon som som skal autentiseres må transportagenten altså i det typiske tilfellet kjøres på samme maskin/lokale nett, under samme bruker, som EDI-programvaren selv.

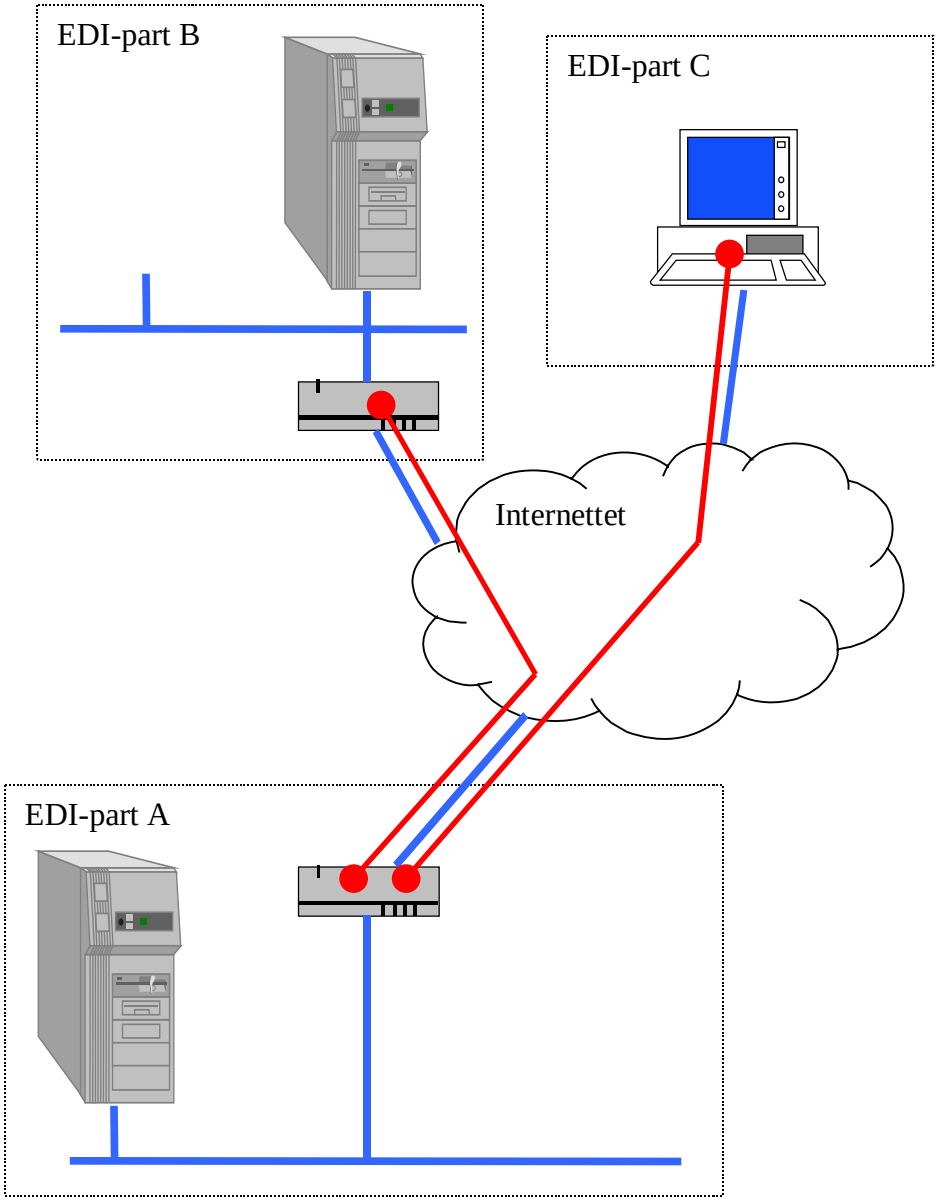
Transportlagssikkerhet, og spesielt TLS, bygges vanligvis inn i bruker/tjenesteprogramvaren. En er avhengig av transportagenter (SMTP-tjenere, HTTP-tjenere og klienter) som selv tilbyr disse protokollene. Har en eldre programvare som ikke inkluderer TLS i utgangspunktet, må en altså inn og modifisere eller oppgradere hvert enkelt program for seg, og det lar seg selvfølgelig ikke alltid gjøre. I noen tilfeller kan det være aktuelt å bruke en TLS ”proxy”, f.eks. i forbindelse med en brannvegg. Proxy-tjeneren kjører på samme nett som tjeneren den beskytter. Den mottar oppkoblinger fra eksterne TLS-klienter, og opptrer selv som SSL-tjener, men formidler data videre til og fra den egentlige tjeneren som om den var en ordinær klient, på en vanlig TCP-forbindelse (ubeskyttet). (Vice versa for en proxy-klient.) Løsningen en da får svarer sikkerhetsmessig mer til virtuelle private nett og ser ut til å være lite utbredt. Fordelen er at man slipper med å installere én proxy i lokalnettet, i

stedet for å modifisere klient- og tjener-programvare. Men samtidig er det denne ene proxyen som blir sikkerhetsparten for all trafikk (autentiserer og krypterer utgående, verifiserer og dekrypterer innkommende).

3.6 Virtuelle private nett

Ved virtuelle private nett flyttes sikkerhetsmekanismene enda ett hakk lenger ned i protokollhierarkiet, til nettverkslaget. Det betyr at all trafikk til eller fra en bestemt maskin eller (lokal)nett beskyttes under ett. Sammenlignet med transportlagssikkerhet betyr det at en ikke lenger kan skille mellom ulike programmer eller brukere på sender eller mottakersiden. Utgående trafikk kan stamme fra, og innkommende trafikk kan leses av hvem som helst med adgang til nett eller maskin. (Men merk at for flerbruker operativsystemer er det normalt bare operativsystemet selv, ikke de enkelte brukerne, som har adgang til nettet på nettverksnivå eller lavere.)

Fordelen med slike løsninger er at det ikke er nødvendig å modifisere eller omkonfigurere eksisterende klient- og tjenerprogramvare i noen vesentlig grad. Ved å installere og konfigurere en VPN-løsning i maskin eller brannvegg lar det seg gjøre å beskytte all trafikk til og fra andre steder med samme type VPN-løsning. Sikkerhetsegenskapene blir mer oversiktlige når system/nettverksansvarlig kan vite at all trafikk er dekket, uten å måtte gå detaljert gjennom (eller selv ha kontroll over) all programvare og alle anvendelser som kjører, og uten at brukere og applikasjonsansvarlige i samme grad trenger å bli involvert.



Figur 8 Virtuelt privat nett

For den enkelte bruker/anvendelse kan dette både være en fordel og en ulempe. Har anvendelsen ikke noen spesielle sikkerhetskrav utover det som er dekket av VPN-løsningen, er det selvfølgelig en fordel å slippe å bygge inn sikkerhetsmekanismer i sine egne programmer. Men det er heller ikke mulig å endre eller omkonfigurere sikkerhetsmekanismene uten å involvere den som har ansvaret for VPN-løsningen, og i verste fall har andre anvendelser i samme nett motstridende krav. Når sikkerheten er basert på meldingssikkerhet eller typiske transportsikkerhetsløsninger, er det til sammenligning brukerens eget valg å bruke programmer med passende sikkerhetsmekanismer.

Betegnelsen ”virtuelle private nett” henspeiler på at sikkerhetsegenskapene er som om maskiner eller lokale nett er sammenknyttet ved et privat (lukket) nett, selv om trafikken faktisk transporteres over et offentlig nett. Teknisk løses dette ved at en etablerer ”tunneller” gjennom det offentlige nettet, dvs. kanaler der trafikken beskyttes kryptografisk. Det ”virtuelle” nettet er det som består av (dvs.

er koblet sammen gjennom) disse tunnellene. Figur 8 viser den samme EDI-anvendelsen som i Figur 4, men denne gangen realisert med virtuelle private nett. Den fysiske kommunikasjonen skjer gjennom Internettet, men som om det var direkte, sikker forbindelse mellom ruterne (brannveggene). Disse "IP-tunnellene" er tegnet inn ved siden av de fysiske nettverksforbindelsene, og det er vist hvordan disse ender i ruterne – helt tilsvarende direkte samband i et privat nett. C, som ikke har noen ruter, bruker en implementasjon av samme VPN-protokoll rett på maskinen som kjører EDI-programvaren, slik at IP-tunnellen ender der.

Detaljene i dette varierer imidlertid ganske mye fra protokoll til protokoll og produkt til produkt, og både sikkerhets- og bruksegenskaper avhenger naturlig nok av dette. Eksempler på variasjonsmuligheter en må regne med å ta stilling til når en vurderer løsning og produkt:

- VPN-protokoll. Det finnes en håndfull ulike protokoller. Tilgjengelighet av programvare/utstyr og rene tekniske avveininger er nok forholdsvis mer viktig her enn det å velge standard protokoller ut fra hensynet til åpenhet og interoperabilitet. Tross alt ligger det i sakens natur at det er en forholdsvis begrenset og oversiktlig gruppe av kommunikasjonsparter en skal etablere et VPN sammen med.
- Hvordan VPN-produktet teknisk integreres med eksisterende systemer og programvare. (F.eks.: Programvaremoduler som plugges inn i brannvegger, -i operativsystem. Separate demon-prosesser på en eller alle maskiner i nettet. Dynamiske biblioteker — DLL.)
- Mulighet for sameksistens mellom forskjellige produkter og løsninger. (Bl.a.: Kan samme lokalnett eller maskin involveres i forskjellige VPN, ved bruk av forskjellige protokoller og produkter, forskjellig sikkerhetsnivå osv.)
- Hvordan nøkler håndteres. (F.eks.: Hvor sterkt nøklene beskyttes. Hvordan nøkkeldistribusjon løses. Hvor stor fleksibilitet som finnes til å bruke forskjellige nøkler, f.eks. for trafikk til forskjellige parter, eller som rutes forskjellig.)

Mange eksisterende EDI-løsninger, der sikkerheten vanligvis er basert på en eller annen grad av lukkede nett i utgangspunktet, kan antakelig flyttes til en passende VPN-løsning uten noen særlig endring. Dette er nettopp en konsekvens av at VPN-løsningen normalt bygges inn i nett- eller operativsystem-infrastrukturen, ikke i klient-, tjener- eller applikasjonsprogramvare. Det er også denne egenskapen som er hovedgrunnen til at VPN er attraktivt, ikke bare for EDI, men for mange andre nettverksanvendelser.

I mange EDI-anvendelser er risikoen for misbruk fra deltakernes side liten. Deltakerne har rimelig tillit seg imellom, eller det er begrenset hvor mye skade en av dem kan gjøre (evt. uten at det spores og rettes opp.) Hovedformålet med sikkerhetsmekanismene er da å hindre utenforstående i å komme til tjenester eller data. I slike tilfeller egner VPN seg godt. Men i anvendelser som involverer spesielt sensitive data, eller der det er relle krav til ikke-benektbarhet gir VPN for "grovkornet" kontroll.

Det finnes flere proprietære VPN-protokoller (PPTP m.fl.). På sikt ventes IPSec (IP Security)-standardene å bli dominerende (-----), . Se også VPN Consortium (http://www.vpnc.org)

4 Tjenesteleverandørens rolle

4.1 Tjenestetyper

EDI-brukeres forhold til tjenesteleverandøren reguleres i en *kommunikasjonsavtale*. Det finnes standard kommunikasjonsavtaler [ref], men disse er hovedsakelig innrettet mot tradisjonelle EDI-formidlingssentraler. Med bruk av åpne nett kan det være mer aktuelt å kjøpe forskjellige tjenester fra hver sin leverandør, f.eks. basis transporttjeneste fra en vanlig Internett-tjenesteleverandør, sertifiseringstjenester fra en annen og spesifikke EDI-tjenester fra en tredje.

En grov klassifisering av aktuelle tjenester er:

- Basis transporttjeneste, f.eks. X.25, X.21, ISDN-abonnement, leie av faste linjer, Internett-tilknytning, altså tjenester på nettverkslag eller lignende.
- Meldingsutvekslingstjeneste, dvs. tjenesteleverandøren driver en transportagent, X.400, SMTP eller annet.
- Generelle verdiøkende tjenester og TTP-tjenester, f.eks. sertifiseringsautoritet, elektronisk notar (tidsstempling).
- EDI-orienterte verdiøkende tjenester og TTP-tjenester, f.eks. portnere, konvertering til og fra EDIFACT.

Denne klassifiseringen svarer til nivåene i vår skjematiske EDI-arkitektur.

På tvers av dette er det dessuten et skille mellom tjenester der tjenesteleverandøren står for leveranse, installasjon og drift av utstyr og programvare hos brukeren, og der brukeren selv står ansvarlig for dette.

Ansvarsforholdene mellom bruker og tjenesteleverandør varierer sterkt mellom disse kategoriene, ettersom innholdet i tjenestene er vesensforskjellig. En formidlingssentral tilbyr gjerne alle typer tjenester i én pakke, og det er rimelig for tjenesteleverandøren å ta et overordnet ansvar for alle forhold som angår meldingsutvekslingen. En X.400-tjenesteleverandør kan påta seg ansvar for at meldinger som er sendt blir levert mottaker innen en viss tidsfrist. Og omvendt, en bruker av en X.400-tjeneste er avhengig av tjenesteleverandørens garantier på dette punktet.

Dersom EDI-partene selv opererer transportagentene og bare kjøper basis kommunikasjonstjenester (f.eks. Internett-tilknytning) for å knytte dem sammen, kan en naturlig nok ikke stille tjesteleverandørene til ansvar for at sendte meldinger blir mottatt. Kvaliteten av transporttjenesten måles i tilgjengelighet og båndbredde. Utover dette har tjenesteleverandøren ingen innflytelse på leveransen av EDI-meldinger. Men dette er ikke nødvendigvis en ulempe: Samtidig er brukernes avhengighet av tjenesteleverandøren redusert på tilsvarende måte.

4.2 Sikkerhetsmessige implikasjoner ved nettløsningen

Den viktigste grunnen til å velge et relativt lukket nett har gjerne vært sikkerhetskrav. Med åpne nett, som Internettet, er det vanligvis ikke noen vei utenom å håndtere sikkerhet ende til ende, enten ved kryptering og digital

signatur på meldingene eller ved tilsvarende kryptografiske mekanismer på transportlaget eller nettverkslaget. Dette gjør sikkerheten til brukernes ansvar, og det er både styrken og svakheten ved denne angrepsmåten.

Rent teknisk må sikkerhetsløsningene bygges inn i den programvaren brukerne selv benytter, dvs. brukeragent eller høyere. Omvendt, dersom den graden av sikkerhet som et lukket nett i seg selv tilbyr er tilstrekkelig, behøver ikke sikkerhetsmekanismene direkte berøre brukerne eller den programvaren de benytter. Dette er det som i praksis er tilfelle for de fleste EDI-anvendelser som er i drift nå.

Baseres sikkerheten på et lukket nett, blir brukerne imidlertid svært avhengige av tillit til tjenesteleverandøren(e). Dersom tjenesteleverandøren kan gi juridisk bindende garantier om å oppfylle sikkerhetskravene brukerne har, kan dette være en god løsning. I de fleste tilfeller regner vi med at brukerne i prinsippet er henvist til å vurdere leverandørens praksis og tekniske løsninger og gjennom dette om risikoen er akseptabel. Denne vurderingen involverer mange faktorer:

- I hvilken forstand, og i hvor stor grad nettet er lukket (dvs. hvilken *sikkerhetspolicy* som gjelder). — Hvilke brukere som har legitim adgang, hvilke krav som stilles til dem, hvilke typer trafikk som tillates, hvilke portnere som finnes mot andre nett osv. Selv nominelt legitim trafikk og brukere kan være en sikkerhetsrisiko.
- Hvor effektivt kan sikkerhetspolicy gjennomføres, dvs. kvalitet på driftsopplegg, programvare, konfigurasjoner, adgangskontrollmekanismer osv.

Behov for samtrafikk mellom flere nett eller leverandører forvansker situasjonen adskillig. En må regne med at svakheter akkumuleres, dvs. at problemer i det ene eller andre av dem innvirker på kombinasjonen av alle nettene og i verste fall gjensidig forsterker hverandre. Dersom sikkerhetspolitikken for nettene ikke er sammenfallende, reduseres den totale sikkerheten typisk til et minste felles multiplum. I praksis regner vi med at tjenesteleverandørene i mange tilfeller må avslå å åpne for samtrafikk, fordi de da ikke vil se seg i stand til å oppfylle de sikkerhetsmessige forpliktelsene de har tatt på seg.

Ende-til-ende-sikkerhetsløsninger hugger over denne “gordiske knuten” ved å unngå denne avhengigheten til tjenesteleverandørene. Antakelsen er helt enkelt at en ikke kan ha noen spesiell tillit til sikkerheten i transportnettet. Slik sett blir sikkerhetsløsningen enklere og mer oversiktlig. I stedet kommer tillit til:

- Kryptosystemene som benyttes.
- Implementasjonen av sikkerhetsløsningene.
- Hemmelighold av private nøkler og autentisitet av offentlige nøkler.

De to første punktene er en spesialistoppgave (på samme måte som en sikkerhetsgjennomgang av en tjenesteleverandør), men i praksis vil dette oppfylles ved å benytte anerkjente produkter og teknikker. Det siste punktet avhenger først og fremst av brukerne selv, men også av programvaren som håndterer nøkler (foregående punkt), og eventuelt av en tredjeparts sertifiseringsautoritet.

4.3 Andre krav til tjenestekvalitet

EDI, som annen profesjonell bruk av datanett, stiller flere krav til tjenestekvalitet, som pålitelighet, tilgjengelighet og kapasitet. Tjenestekvalitetskrav er enklere å

oppfylle i lukkede nett, hvor operatøren har bedre oversikt over brukerne, deres kapasitetsbehov og trafikkprofiler osv. I åpne nett kan det være vanskelig å beskytte mot tjenestenektingsangrep, dvs. ondsinnede forsøk på å redusere kvaliteten av en tjeneste ved direkte overbelastning eller ved å utnytte feil og svakheter i protokoller og implementasjoner av dem. Misbruk, feil bruk, feilkonfigurering hos brukerne av en tjeneste kan også gjøre det vanskelig å opprettholde tjenestekvaliteten for andre brukere, og er mer sannsynlig når brukergruppen er stor og uoversiktlig, kanskje med mange relativt ukyndige brukere og mange forskjellige typer tredjeparts programvare og utstyr hos brukerne. Men merk at nettopp det at en har en slik brukergruppe kan være grunnen til at en ønsker å benytte et åpent nett (som Internett).

Tjenesteleverandørene, spesielt formidlingssentraler, yter også andre tjenester enn datanett-tjenesten i seg selv. Eksempler kan være feilretting, brukerstøtte, tilrettelegging, installasjon og drift av programvare osv. Dette faller ikke inn under det en normalt kaller tjenestekvalitet, men disse tjenestene kan også være vesentlige for valg av tjenesteleverandør. I Internettet og andre åpne nett skal programvare, utstyr og tjenester fra mange ulike parter spille sammen, og for brukere uten dyp teknisk kompetanse kan det være behov for en tjenesteleverandør som tar ansvaret for helheten når det ikke er så klart hvor feil og problemer ligger.

Tjenestekvalitet i forbindelse med nettverksprotokoller, f.eks. IPv6, ATM o.l., gjelder spesifikke krav til båndbredde, feilrater, forsinkelser og andre sanntidsegenskaper. De fleste av denne typen krav er ikke spesielt vesentlige i EDI (spesielt ikke satsvis EDI) hvor sanntidskravene er svært moderate. Av disse mekanismene er det først og fremst muligheter for reservasjon av båndbredde eller prioritering av trafikk som kan være aktuelle.

Det er ikke protokollene (eller nettverksteknologiene) selv som tilbyr tjenestekvalitet, men de konkrete implementasjonene av dem i form av tjenestetilbud hos leverandørene. Protokollene legger føringer for hva leverandørene kan tilby. Det at en protokoll (f.eks. ATM eller IPv6) har mekanismer for å håndtere tjenestekvalitet, betyr ikke noe særlig mer enn at protokollen gir mulighet for å identifisere hvilken tjenestekvalitetsklasse en viss trafikkstrøm tilhører, og i enkelte tilfeller, for å reservere ressurser i nettet til å håndtere trafikkstrømmen. Tjenesteleverandørene får noen ”håndtak” som de kan bruke til å prioritere visse typer trafikk, men det er de prioriteringsreglene de følger og den kapasiteten de faktisk har i nettet sitt som er utslagsgivende.

5 Protokoll- og formatkonvertering gjennom portnere

Portnere[1] benyttes når det er behov for samspill mellom løsninger som benytter forskjellige protokoller eller dataformater. Bruk av portnere for disse formålene må regnes som en nødløsning og har ofte uheldige konsekvenser. I forbindelse med EDI er det i første rekke snakk om portnere mellom ulike typer MHS. Tjenester som konverterer meldinger mellom forskjellige EDI-formater er også portnere. Dessuten kan det være behov for portnere i forbindelse med katalogtjenester og andre tjenester som understøtter MHS.

Hovedproblemene med portnere er:

- Protokollene eller formatene som portnerne konverterer mellom er aldri helt sammenlignbare. Det vil finnes mekanismer i den ene som ikke kan oversettes til den andre, eller kan oversettes, men på en måte som bare er gyldig under visse forutsetninger eller har nyanseforskjeller.
- De introduserer et ekstra ledd som bryter opp kjeden mellom avsender og mottaker og gjør det vanskeligere å oppfylle ende til ende-egenskaper. Spesielt sikkerhetsmekanismene i tjenestene lider under dette.

Merk: X.400 kan kjøre over ulike transportnett (også IP). SMTP og andre Internett-protokoller baserer seg på TCP-, (UDP-) og IP-protokollene, som også kan kjøre over forskjellige underliggende nett-teknologier. Dette er *ikke* basert på portnere på transportagent-nivå eller høyere og er ikke belemret med disse problemene.

Vi har tidligere (kapittel 3.4) diskutert bruk av portnere (proxyer) i brannvegger. Disse portnerne utfører vanligvis ikke noen konvertering fra en protokoll til en annen og skaper derfor heller ikke denne typen problemer. (Men det kan finnes mekanismer i en protokoll som kan skape sikkerhetsproblemer, og en av de vesentlige grunnene til å ha en brannvegg er nettopp å blokkere disse.)

Portnere mellom to ulike MHS A og B kan operere på to forskjellige måter:

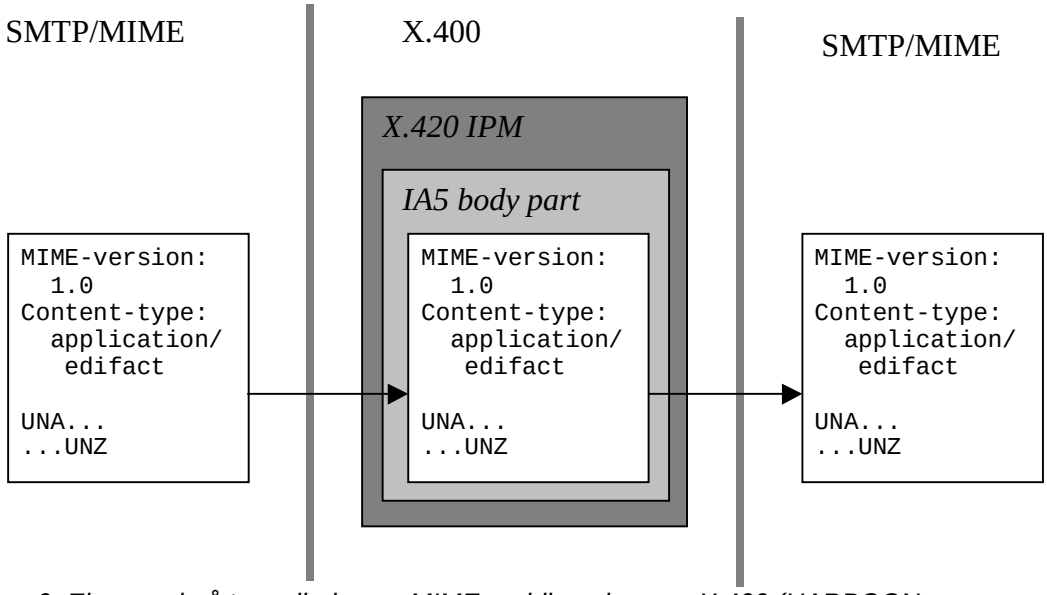
- **Innkapsling**: En melding i A pakkes inn som en innholdsdel i en ny melding i format B.
- **Konvertering**: En melding i A konverteres til en mest mulig ekvivalent melding i B.

Innkapsling er en form for tunnellering (jfr. VPN, kapittel 3.6). Fordelen er at innkapslingen kan reverseres og den opprinnelige meldingen gjenvinnes intakt. Dermed kan en A-melding sendes (tunnelleres) gjennom et B-nett og ut igjen i et annet A-nett. Men hvis avsender og mottaker er i hvert sitt nett, trenger de brukeragenter som forstår hverandres formater. Og hvis meldingen passerer flere portnere, og konvensjonene for å identifisere innkapslingen ikke er standardiserte og entydige, risikerer en flere lag med innkapsling.

Et typisk eksempel på innkapsling er HARPOON -, som spesifiserer hvordan en (vilkårlig) MIME-innholdsdel direkte kan kapsles inn i en X.420 IA5-innholdsdel, rett og slett ved å inkludere MIME-hodet som en del av IA5-innholdet og bruke

[1] Dessverre er ikke bruken av ordet ”portner” helt entydig. Vi bruker her ordet bare om applikasjonsnivå-portnere, dvs. portnere som opererer på applikasjons-protokollnivå. Ordet ”portner” brukes bl.a. også om enheter som kobler sammen ulike nett. ”Portnere” på IP-nivå bør kalles ”rutere” for å unngå forvirring.

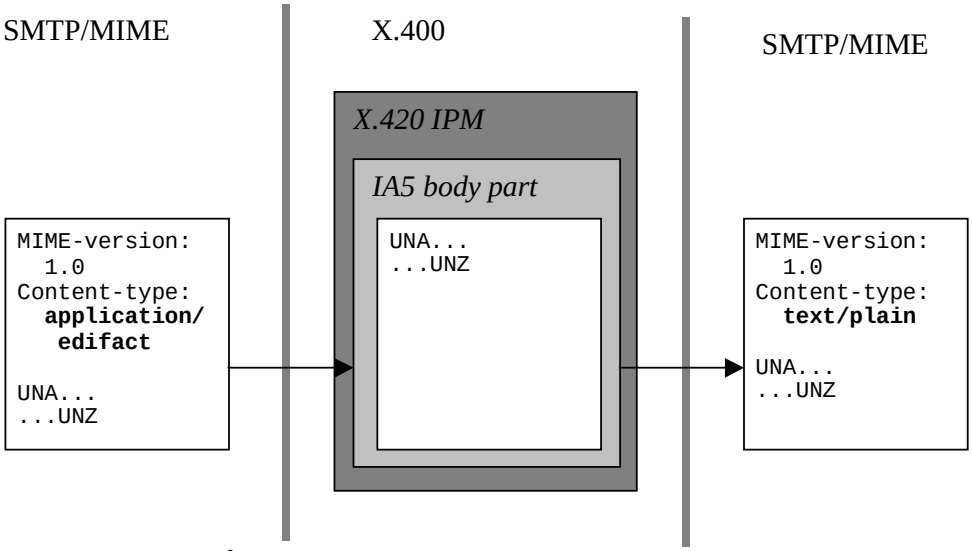
MIME content-transfer-encoding når det er nødvendig for innhold ut over 7-bits tekst (IA5/ISO 646 IRV).



Figur 9 Eksempel på tunnellering av MIME-melding gjennom X.400 (HARPOON-innkapsling)

Ved konvertering risikerer en å miste funksjonalitet og/eller informasjon. Når en melding konverteres fra A til B og tilbake til A-format, vil den vanligvis ikke være eksakt den samme som den opprinnelige. Men en mottaker i B-nettet kan forholde seg direkte til den.

I figuren nedenfor vises et eksempel på at en (ikke MIXER-kompatibel) portner konverterer en MIME application/edifact-melding til en IA5 body part og mister MIME-innholdstypen ved konvertering tilbake. Poenget med konvertering, i motsetning til innkapsling, er at informasjonen fra MIME-hodet eller RFC822-hodet skal opp i IPM-meldingshodet eller hodet på innholdsdelen, slik at det blir tilgjengelig for X.420-brukeragenter, som ikke vil gi seg i kast med å tolke innholdet i innholdsdelen. Når det ikke finnes noe tilsvarende felt til MIME Content-type, går denne informasjonen tapt. (Eksempelet er ikke så søkt som det kanskje virker. Mange EDI-løsninger transporterer EDIFACT i IPM-meldinger på den måten som er vist, og baserer seg på en konvensjon om at mottakeren på den gitte adressen tolker innholdet som en EDIFACT-utveksling.)



Figur 10 Eksempel på konvertering av MIME-melding til X.420 IPM og tilbake.

For portnere mellom X.400 (IPM) og Internett-epost er de mest grunnleggende funksjonene standardiserte (MIXER: --). Den grunnleggende strategien i MIXER er å konvertere det som lar seg konvertere, og bruke innkapsling for resten. For portnere mellom X.435 (EDI-meldinger) og Internett-epost, eller mer spesifikt EDIINT, er det imidlertid ikke definert noen standard.

De punktene som først og fremst skaper problemer mellom X.400 og Internett-epost, er:

- Utvalget av innholdstyper og varianter av dem (f.eks. tegnkoder) samsvarer ikke helt mellom X.420 og MIME. Dette er mindre relevant for EDI-trafikk som kun bruker EDIFACT. Konvertering av innhold kan imidlertid skape problemer dersom en bruker spesielle konvensjoner for innholdsrepresentasjon, f.eks. interimsløsningen for EDIFACT i X.420 IPM-, eller koder tekst i NS ISO 646 som innholdstyper som egentlig impliserer tegnsett ASCII/IA5/ISO 646 IRV. Disse konvensjonene går ut over forutsetningene i -!
- Kvitteringsmekanismer fungerer i alminnelighet ikke gjennom portnerne. Dette vil trolig forbedre seg når kvitteringsmekanismene i Internett-epost blir mer utbredt i implementasjonene. Hovedtrekkene i mekanismene på begge sider er de samme, men de tekniske detaljene er likevel så forskjellige at en må regne med å miste informasjon ved konvertering. Konvertering av kvitteringer fra X.400 til Internett-format vil trolig ha bedre sjanse for å lykkes enn omvendt. (Internett-formatene er noe rikere enn X.400, og er dessuten til en viss grad lagt til rette for at slik konvertering skal være mulig.)
- Ende til ende-sikkerhetsmekanismer på MHS-nivå kan ikke konverteres. Når en melding er beskyttet med kryptering eller digital signatur, er den ikke tilgjengelig for en portner. I tilfelle må portneren fjerne denne ”sikkerhetskonvolutten” og legge på sin egen, og beskyttelsen er ikke lenger ende til ende, men ende til portner og portner til ende.

Erfaringsmessig er det enda større problemer med portnere mellom Internett (og X.400) på den ene siden og forskjellige proprietære epost-systemer på den andre siden. (Epost-systemer på Windows-plattform er i stor grad basert på proprietære protokoller og formater internt, selv om de utad tilbyr tilgang gjennom standard Internett- eller X.400-protokoller.) En del av Internettets rykte for upålitelighet kan nok i virkeligheten tilskrives slike portnere: Mottakerens adresse

er tilsynelatende en normal RFC822-adresse, og avsenderen vil ofte ikke være klar over at det er en portner inne i bildet.

Dersom det er behov for portnerløsninger mellom Internett og X.400, anbefales det:

- Portneren må drives av en ansvarlig tjenesteleverandør, og spesifikasjoner må være tilgjengelige, spesielt dokumentasjon av begrensninger og forutsetninger.
- Sikkerhetsmekanismer bør legges på EDIFACT-nivå. Alternativt benyttes S/MIME i henhold til EDIINT-spesifikasjonene. S/MIME-innholdsdeler transporteres som sådan i X.400, dvs. *innkapslet* i X.420-innholdsdeler (IA5, BP14, BP15 eller FTBP, se -) eller i X.435 som en EDI-innholdsdel (EDIBodyPart) med type ”undefined” eller ”private”. (S/MIME-innholdsdeler på Internett-siden vil dessuten under visse forutsetninger kunne *konverteres* til PKCS#7-innholdsdeler i X.420.)
- For automatiske kvitteringsmekanismer benyttes hovedsakelig EDIFACT. Kvitteringsmekanismer på MHS-nivå benyttes først og fremst for manuell feilrapportering og feilfinning.

For samtrafikk med EDI-løsninger som benytter interimsløsningen i - eller lignende konvensjoner for innkapsling av EDIFACT-utvekslinger i IPM, bør håndteringen av disse konvensjonene ligge i **portneren**. Disse konvensjonene er ikke direkte inkompatible med MIXER, men med en vanlig MIXER-portner som ikke tar hensyn til dem må avsender/mottaker på Internett-siden også følge disse spesielle konvensjonene. Dette er ikke kompatibelt med EDIINT!

De innholdstypene som er aktuelle for EDI-trafikk behandles på følgende måte i en MIXER-portner:

Fra MIME til X.420:

- MIME application/edifact eller S/MIME-innhold vil vanligvis bli HARPOON-innkapslet i en IA5-innholdsdel (evt. BP14, BP15 eller FTBP-innkapsling).
- Alternativt kan S/MIME klartekst-signert innhold (multipart/signed) bli konvertert til to X.420-innholdsdeler, hhv. for den opprinnelige innholdsdelen (EDIFACT-utvekslingen) og signaturen.
- Dersom X.400-siden forventer en konvensjon som - som koder EDIFACT-utvekslingen i en IA5-innholdsdel, må EDIFACT-utvekslingen kodes som text/plain med tilsvarende konvensjoner på Internett-siden.

Fra X.420 til MIME:

- Dersom X.420-innholdsdelen inneholder en innkapslet MIME-innholdsdel fjernes innkapslingen.
- En IA5-innholdsdel som inneholder en EDIFACT-utveksling konverteres til MIME text/plain, ikke application/edifact! Merk: EDIFACT-utvekslinger vil ikke inneholde linjeskift. IA5-innholdsdeler kan ha vilkårlig lange linjer (selv om dette ikke anbefales). For MIME text/plain må i tilfelle innholdskoding base64 eller quoted-printable benyttes. Slurvete bruk av tegnsett utover IA5 (8-bits) vil også være problematisk.

6 Krav til meldingsutvekslingssystemet

I dette kapitlet behandler vi de kravene som en EDI-løsning kan stille til meldingsutvekslingssystemet. Kapitlet kan gjerne brukes som en sjekkliste. For å holde kapitlet relativt kortfattet og kompakt, diskuterer vi ikke den tekniske bakgrunnen altfor detaljert, så en viss bakgrunn i datakommunikasjon og drift av nett og applikasjoner er nødvendig. De foregående kapitlene burde hjelpe et godt stykke på vei.

6.1 Innhold

6.1.1 Transparens

Innhold skal håndteres *transparent*, dvs. at EDIFACT-utvekslingen som leveres til mottaker bit for bit er den samme som den som opprinnelig ble avsendt. Det skal ikke utføres noen konverteringer av innhold på meldingstransportnivå. (Dersom konverteringer skjer, må dette utføres av EDIFACT-orientert programvare i en *portner*, altså utenfor MHS.)

Dette kravet vil normalt være oppfylt ved å merke meldingens innhold med rett innholdstype i MHS, f.eks. benytte X.435 EDIM i X.400 eller MIME-type application/edifact i Internett-protokoller. Der dette ikke lar seg gjøre, velges normalt en meldingstype for binære data av ukjent/utolket type, som X.420 IPM med innholdstype Undefined eller MIME-type application/binary. Innholdstyper for tekst anbefales vanligvis ikke, selv om EDIFACT-utvekslinger nominelt er tekst, av følgende grunner:

- Disse innholdstypene er mest problemfrie for 7-bits tegnkoder (ISO 646 IRV eller ASCII), men dette er sjelden tilstrekkelig. Nasjonale varianter av ISO 646 (7-bits) kan vanligvis ikke identifiseres korrekt av innholdstypen og anbefales **ikke** brukt. Ved bruk av 8-bits tegnkoder må disse identifiseres korrekt, og det er ikke helt sjelden at en finner programvare og tjenester som tar snarveier som strengt tatt ikke er hjemlet i standardene.
- Enkelte portnere kan komme til å foreta omkoding fra en tegnkode til en annen, uten at dette reflekteres i EDIFACT-utvekslingshodet.
- Når EDIFACTs mekanismer for å håndtere ulike tegnkoder blir brukt, er utvekslingshodet (UNA og UNB) alltid kodet i ISO 646, mens resten av utvekslingen er kodet i den tegnkoden som UNB-segmentet spesifiserer. Dersom denne tegnkoden ikke faller sammen med ISO 646, er det ikke mulig å transportere utvekslingen i en innholdsdel av type tekst uten å ”lyve” om tegnkoden. (En tekst-innholdsdel har bare én tegnkode som gjelder hele innholdsdelen, både i MIME og X.420.)
- MHS kan ha begrensninger på linjelengde i tekstmeldinger, eller til og med foreta automatisk linjeombrekking. EDIFACT-utvekslinger inneholder ikke linjeskille.
- Bruk av en annen innholdstype enn tekst kan gjøre det lettere å skille EDI-utvekslinger fra vanlige epost-meldinger som sendes til samme adresse ved en feiltakelse (eller fordi avsender antar at de vil bli sett av en person).

6.1.2 Utvekslingslengde

Ideelt sett skal utvekslinger av vilkårlig lengde kunne overføres, men i praksis finnes begrensninger.

- Protokollene selv kan ha maksimumsgrenser, men disse byr neppe på problemer i realistiske situasjoner.
- Implementasjoner av brukeragenter og transportagenter kan ha innebygde begrensninger.
- Transportagenter er ofte konfigurert til ikke å akseptere meldinger over en viss maksimal størrelse (som beskyttelse mot overbelastning).
- Maskinene som brukeragenter og transportagenter kjører på kan ha (midlertidige) kapasitetsproblemer.
- Det underliggende transportnettet kan også skape problemer, men mer indirekte, f.eks. ved at sannsynligheten for feil som avbryter en lang overføring blir uakseptabelt høy.

Begrensningene har altså to former:

- ”Harde” grenser, som gjerne er administrativt satt, eller unntaksvis gjenspeiler begrensninger i utstyr etc.: Utvekslinger over en viss størrelse kan ikke transporteres.
- ”Myke” grenser, der risikoen for tap av meldinger, avbrudd, forsinkelse eller andre uregelmessigheter blir uakseptabelt høy. (Dette betyr ikke at EDI-anvendelsen eller MHS blir upålitelig – slike feil må uansett håndteres. Men retransmisjon og andre feilopprettende tiltak koster.)

Lengden på de enkelte EDIFACT-meldingene skaper sjelden problemer. Formålet med å samle flere meldinger i en utveksling er vanligvis rent praktisk, å utnytte MHS bedre (når det er en forholdsvis stor kostnad pr. utveksling). Det er sjelden EDI-løsningens logikk som tilsier at flere meldinger må grupperes i en enkelt utveksling. Hvis det er en kjent maksimumsstørrelse, er det altså enkelt å unngå å samle opp for store utvekslinger.

Den praktiske maksimumsstørrelsen for utvekslinger er vanskelig å bestemme. Den kan avhenge av hvem en kommuniserer med, annen belastning på nett og tjenester og andre forhold en ikke har oversikt over. I de aller fleste tilfeller er det antakelig liten grunn til å presse grensene: En gjør et overslag over hvor store utvekslinger som i praksis forekommer og kan forsikre seg om at dette, med rimelig margin, ikke skaper praktiske problemer.

Dersom en utveksling overskrider en satt maksimumsstørrelse, eller overføringen feiler pga. midlertidige kapasitetsproblemer, er dette et spesialtilfelle av ”ikke-leveranse” og skal feilmeldes, typisk med en X.400 NDR eller Internett DSN til avsender.

Dersom anvendelsen krever større enkeltvise EDIFACT-meldinger enn MHS kan håndtere pålitelig, kreves spesielle tiltak. Avsenders brukeragent kan på forhånd bryte opp utvekslinger i fragmenter som sendes som enkeltvise meldinger gjennom MTS, og så settes sammen igjen i mottakers brukeragent. Enkelte protokoller, bl.a. MIME, har standardiserte mekanismer for dette. En annen type mekanisme er ”checkpoint/restart”, hvor en overføring kan gjenopptas etter et avbrudd uten å starte fra begynnelsen av.

6.1.3 Identifikasjon av EDIFACT-utvekslinger

Når meldinger leveres til mottakers brukeragent, må de kunne identifiseres som EDIFACT-utvekslinger slik at brukeragenten kan aktivere rett program for viderebehandling av dem. Dette kan skje på flere måter:

- Spesifikke MHS-innholdstyper for EDIFACT. (Se ovenfor.)
- Egne mottakeradresser i MHS. (Konvensjoner om filnavn og katalognavn ved bruk av filoverføringsprotokoller kommer også inn under dette punktet.)
- Bilateralt avtalte konvensjoner om innhold (f.eks. bruk av spesielle kodeord i meldingens overskriftsfelt).

Videre må det være mulig å skille mellom ulike EDI-anvendelser som sameksisterer hos en bruker. Dette kan gjøres i MHS gjennom de samme mekanismene som over, men også ved EDIFACT-spesifikk programvare som ruter utvekslingen eller enkeltmeldingene til rett anvendelse på grunnlag av informasjon i utvekslingens eller meldingenes hode (f.eks. mottakers identifikator, meldingenes type).

6.2 Adressering og ruting

EDIFACT-utvekslinger adresseres primært ved mottakers EDIFACT-identifikator (i utvekslingens UNB-segment). Disse identifikatorene er globalt entydige. En EDI-brukeragent bør støtte denne formen for adressering.

For å kunne transportere utvekslingen gjennom et MHS må identifikatoren oversettes til en gyldig MHS-adresse. Adresseformatene varierer fra MHS til MHS. Denne oversettingen kan løses på flere måter:

- Manuelt (eller automatisk) oppdaterte tabeller hos avsender.
- Transformasjonsregler for å avlede MHS-adresse fra EDIFACT-identifikatoren. (For et delmengde av EDIFACT-identifikatorer, f.eks. én enkelt navneautoritet, det finnes ingen allmengyldige regler.)
- Oppslag i katalogtjeneste, f.eks. X.500.
- EDI-trafikk rutes til formidlingssentral, som utfører adresse-oversettingen.

Oversettingen vil vanligvis bare være gyldig for en viss delmengde av EDIFACT-identifikatorer, og hva som skal dekkes må spesifiseres nærmere.

En MHS-adresse sier i prinsippet bare hvem mottaker er, uavhengig av hvilken vei gjennom nettet som MTS skal bruke for å nå mottakeren. (I praksis gjenspeiler ofte strukturen i MHS-adressene nettstrukturen, men en bør unngå å bygge inn antakelser om dette i MHS eller EDI-anvendelse.) Ruting-funksjonen, det å velge transportvei for en gitt melding, hører først og fremst hjemme i MTS.

Likevel kan avsenders brukeragent også måtte ta stilling til ruting av meldingen, f.eks. valg av formidlingssentral eller MHS. Dette kan være tilfelle når ulike mottakere nås gjennom forskjellige MHS eller har kundeforhold til forskjellige formidlingssentraler. Det kan også være tilfelle når forskjellige meldinger til samme mottaker kan ha forskjellige krav til f.eks. sikkerhet og tjenestekvalitet eller forskjellige behov for tredjepartstjenester. Antakelig vil ruting dekkes av de samme mekanismene som benyttes for å oversette EDIFACT-identifikator til MHS-adresse.

Ved bruk av filoverføringsprotokoller som FTP og FTAM blir adressering og ruting av meldinger tettere sammenvevd, og mer avhengig av anvendelses- eller parts-spesifikke konvensjoner, utforming av sikkerhetsmekanismer m.m. I stedet for at en EDIFACT-identifikator direkte oversettes til en MHS-adresse må senderen vite:

- Adresse(r) til filtjener(e).
- Brukernavn/passord eller andre mekanismer for autentisering mot filtjener.
- Navn på filområde og andre konvensjoner for filnavn.

Mens MHS-adressen ikke avhenger av noe annet enn EDIFACT-identifikatoren til mottakeren, og evt. type MHS, vil detaljene ovenfor typisk være forskjellige for hver enkelt sender som skal nå en bestemt mottaker.

6.3 Sikkerhet

Sikkerhet i EDI-løsninger omfatter flere punkter, og de viktigste er:

- *Meldingssikkerhet*, dvs. beskyttelse av de enkelte meldingene eller utvekslingene under transport.
- *Kommunikasjonssikkerhet*, dvs. beskyttelse av transportkanalene. Dette kan realiseres som *transportlagssikkerhet* eller *nettverkslagssikkerhet*.
- *Vertsmaskinsikkerhet*, dvs. beskyttelse av de vertsmaskinene og lokale nettene som EDI-løsningen kjører på når disse kobles opp mot et eksternt nett.

Veiledningens del 5 omhandler sikkerhet generelt og tar dessuten spesifikt for seg mekanismene for å realisere meldingssikkerhet. I den sammenhengen kommer den også inn på de overordnede kravene til kryptografiske sikkerhetsmekanismer, dvs. til kryptosystemer, nøkkeladministrasjon og sertifisering. Disse kravene er de samme også når kryptografi blir brukt for å realisere kommunikasjonssikkerhet, enten på transportlag eller nettverkslag.

For å formulere sikkerhetskrav må det normalt gjennomføres en analyse av hvilke sikkerhetstrusler en anser for reelle.

- Hvilke typer angrep som kan tenkes.
- Hvem de mulige angriperne er, hvilken motivasjon de har, og hvilke ressurser de kan sette inn.
- Hvilken risiko partene selv løper (sannsynlighet for gjennomført angrep multiplisert med konsekvens/skade).

Bedriften selv bør allerede ha en sikkerhetspolicy, men når den skal knytte virksomhetskritiske systemer til et eksternt nett er det i hvert fall på tide å definere en. EDI-løsningen kan ha sin egen sikkerhetspolicy, evt. underordnet bedriftens generelle.

Et vesentlig element i en sikkerhetspolicy er å definere en sikkerhetsarkitektur: hvilke sikkerhetsdomener en har, plassere systemkomponenter, data og tjenester i domenene, definere tillitsforhold, rettigheter, adgangskontroll osv. Dette nedfeller seg konkret i f.eks.:
Rutiner og administrative tiltak.
Nettstruktur, regelsett i brannvegger, ruter-konfigurasjon.
Brukere, autentiseringsmekanismer, aksessrettigheter på filer og andre systemobjekter.

Håndtering av passord, kryptografiske nøkler, adgang til nøkkelmedia.

Likevel er det om å gjøre å frikoble spesifikasjonen av policy fra den konkrete, tekniske implementasjonen av den. En sikkerhetspolicy er et så viktig dokument at den bør kunne leses og forstås av bedriftens ledelse og andre uten spesialisert teknisk kunnskap. Dessuten vil det ofte være slik at en viss policy kan implementeres på ulike måter, og vice versa, det er ikke alltid åpenbart hva som er intensjonen bak en viss implementasjon eller hvordan forskjellige elementer i implementasjonen spiller sammen for å realisere en intensjon.

Sikkerhetspolicy bør også definere rutiner på andre punkter:
Verifisere systemets integritet, både automatisk overvåkning og kontroll, og manuell sikkerhetsrevisjon.
Endringskontroll for kritiske elementer: Oppgraderinger av programvare, endringer i brannvegg-konfigurasjoner etc.
Logging, revisjonsspor for å sikre at kritiske hendelsesforløp kan rekonstrueres (og feil/skade kan rettes eller kompenseres).
Overvåkning for å detektere angrep, forsøk på angrep eller andre kritiske hendelser.
Tiltak for å håndtere faktiske angrep og andre kritiske hendelser.

En sikkerhetspolicy behøver ikke være et stort dokument. For enkle anvendelser kan mange av disse punktene vise seg å være trivielle. Hvis man f.eks. har en løsning der systemene ikke er permanent tilknyttet et eksternt nett, men bruker meldingslager hos en tjenesteleverandør, er det kanskje ikke nødvendig med noen særlige tiltak for å beskytte seg mot eksterne innbrudd. Men da er det desto mer viktig å dokumentere i sikkerhetspolicy at denne forutsetningen gjelder.

6.3.1 Meldingssikkerhet

Sikkerhetsmekanismene er knyttet direkte til en melding eller EDIFACT-utveksling, og sikkerhetsgarantiene følger denne ende til ende gjennom MHS. Disse er uten unntak basert på kryptografi, normalt offentlig nøkkel-kryptosystemer.

- På EDIFACT-nivå, dvs. ved å bruke sikkerhetssyntaksen definert som en del av EDIFACT-syntaks.
- På innholdsdel-nivå, med en generell sikkerhetssyntaks.
- På MHS-nivå, med MHS-spesifikk sikkerhetssyntaks.

Aktuelle mekanismer for meldingssikkerhet på hhv. EDIFACT-nivå og innholdsdel-nivå er behandlet i veiledningens del 5. I X.400 finnes også MHS-nivå meldingssikkerhetsmekanismer. Trolig er disse mekanismene lite aktuelle i praksis, bl.a. pga. lite utvalg av implementasjoner, og vi har ikke behandlet dem spesifikt. Grunnprinsippene i disse mekanismene er imidlertid de samme, bare at de er integrert i X.400-meldingssyntaks.

Typer av krav som tilsier bruk av meldingssikkerhet:

- Sikkerheten skal ikke baseres for mye på tillit til andre parter som er involvert i utveksling av meldinger eller har adgang til systemene, f.eks. tjenesteleverandører, andre brukere, driftspersonell.
- Sikkerhetsmekanismene skal kunne implementeres uavhengig av MTS og underliggende nett.

- Sikkerhetsnivået skal være uavhengig av detaljer ved MHS, f.eks. hvilke nett eller tjenesteleverandører en melding er rutet gjennom.
- Sikkerhetsmekanismene skal også kunne anvendes ved intern lagring av meldinger.
- Sikkerhetsgarantier skal kunne verifiseres av ihendehaver av en melding, bare basert på informasjon i meldingen selv.
- Sikkerhetsgarantier skal gjelde mellom meldingenes egentlige avsender og mottaker, ikke parter som opererer på vegne av dem.

6.3.2 Kommunikasjonssikkerhet

Sikkerhetsmekanismene er knyttet til kommunikasjonskanalen(e) som meldingen følger. Kommunikasjonssikkerhet kan baseres både på begrenset adgang til kommunikasjonsnettet (lukket nett) og på kryptografi.

- På transportlag (eller høyere), dvs. til den (virtuelle) transportkanalen, f.eks. en ACSE-assosiasjon eller TCP-forbindelse mellom to av komponentene i MHS.
- På nettverkslag eller lavere, dvs. for all trafikk mellom par eller grupper av maskiner.

Typer av krav som kan tilsi bruk av kommunikasjonssikkerhet:

- Vansker med å integrere sikkerhetsmekanismer i anvendelsesprogrammer eller utstyr, f.eks. bruk av ”hyllevare” som ikke inneholder de rette mekanismene.
- Behov for å påtvinge uniform bruk av sikkerhetsmekanismer på tvers av anvendelser, brukere etc., eller unngå at korrekt bruk av dem blir avhengig av individuelle brukere, applikasjons-ansvarlige etc.
- Unngå eller redusere behov for infrastruktur som sertifisering/nøkkeladministrasjon.
- Behov for å hindre trafikkanalyse (dvs. at uvedkommende skaffer seg oversikt over trafikkmønstre — hvem kommuniserer med hvem, når).
- Beskyttelse mot uvedkommende som genererer uønsket trafikk (duplikat av genuine meldinger, tjenestenektingsangrep etc.)

6.3.3 Konfidensialitet

Krav til konfidensialitet kan løses både gjennom kommunikasjonssikkerhet og meldingssikkerhet. Meldingssikkerhet vil normalt gi sterkere garantier om at meldingens innhold bare er tilgjengelig til de enkeltpersoner eller –parter som meldingsutstederen har tilsiktet.

Meldingssikkerhet gjør det mulig å beskytte meldingenes innhold. Det at det foregår trafikk mellom to parter er vanskeligere å skjule. Dersom det er krav om å hindre slik trafikkanalyse, kan det tilsi bruk av kommunikasjonssikkerhet (lukket nett).

6.3.4 Integritet

Det vil vanligvis ikke være noen grunn til å kreve egne mekanismer (f.eks. ende til ende-kontrollsummer) for å ivareta krav til integritet. Alle aktuelle MHS vil ha mekanismer som sikrer mot utilsiktede endringer i meldingene under overføring.

Dersom disse ikke gir god nok beskyttelse mot utilsiktede endringer, vil det trolig også være behov for å sikre mot bevisste forfalskninger, dvs. autentisering av meldingsinnhold. (For meldingssikkerhet vil autentisitet alltid implisere integritet. For kommunikasjonssikkerhet kan autentisitet sikres gjennom mekanismer som ikke nødvendigvis gir integritet.)

6.3.5 Autentisitet

EDI har normalt et sterkt krav til autentisering av meldinger, ettersom en EDI-melding både kan forplikte avsender og utløse kritiske handlinger hos mottaker.

Autentisering i sin sterkeste og mest direkte form (dvs. en eller annen form for digital signatur) er en meldingssikkerhetsmekanisme. Autentisitet av en melding betyr sikkerhet mot bevisst forfalskning av en melding, dvs. meldingen (med gitt innhold) er beviselig utstedt av den angivelige utstederen. Autentisitet i denne forstand impliserer nødvendigvis integritet.

Autentisering i forbindelse med kommunikasjonssikkerhet betyr at parten i den andre enden av kommunikasjonskanalen er kjent, og sikkerhet for at data som mottas virkelig stammer fra denne parten (og i noen tilfeller, for at data som sendes bare mottas av denne parten.) Kommunikasjonssikkerhet på transportnivå er ofte ende-til-ende (f.eks. SSL/TLS), men ellers kan autentisiteten av utvekslede data vanligvis ikke etterprøves direkte av mottakeren. I stedet må mottakeren basere seg på en kjede av tillit mellom operatører av transportagenter og nett. Et sikkerhetsbrudd hos en av dem river bort fundamentet for autentiseringen.

Det er også verdt å merke seg at det som autentiseres er endepunktene for kommunikasjonskanalen, slik at reelt får en ofte bare sikkerhet for at data stammer fra en viss maskin, innenfor et visst nett eller lignende. Data kan med andre ord stamme fra hvem som helst som rettmessig eller urettmessig har adgang til maskinen.

Krav til autentisering varierer fra anvendelse til anvendelse:

- Hvor stor sikkerhet som kreves mot forfalsket autentisering.
- Hvilke parter som kan gi akseptabel autentisering (evt. på vegne av utsteder).
- Hvilke meldinger som må autentiseres.
- Hva autentiseringen skal omfatte. (EDIFACT-utvekslinger eller -meldinger inneholder vanligvis så mye kontekstuell informasjon at de kan autentiseres alene. Dersom mottaker er avhengig av informasjon utover meldingen selv for å kunne tolke den korrekt, må denne informasjonen imidlertid autentiseres sammen med meldingen. En digital signatur vil f.eks. vanligvis inneholde og autentisere tidspunktet den ble generert.)

Ved siden av EDIFACT-utvekslingene selv kan det være behov for autentisering av forskjellige typer tjeneste- eller kontrollmeldinger.

6.3.6 Ikke-benekting

Ikke-benekting av opphav (non-repudiation, også kalt vedkjenning) kan oppfattes som en sterkere form for autentisitet: Beviset for meldingens opphav er sterkt nok til både:

- å utelukke en forfalskning (som for autentisitet)

- å forhindre at utsteder senere benekter meldingen

Ikke-benekting er nødvendig dersom partene ikke har tilstrekkelig tillit til hverandre til at autentisering alene er tilstrekkelig. Det ligger i sakens natur at dette krever en tredjepart som kan gi de nødvendige garantiene, og som både utsteder og mottaker har tillit til og er villige til å forplikte seg overfor. (Uten tredjeparten degenererer ikke-benekting til autentisitet.)

Krav om ikke-benekting forekommer typisk i situasjoner der:
- partene har kryssende interesser, eller
- partene på forhånd er ukjente for hverandre[1].

Krav om ikke-benekting tar i utgangspunktet samme form som krav til autentisitet. I tillegg til de typene krav som er nevnt for autentisering, kommer imidlertid:

- Krav til rutiner/praksis hos tredjeparten.
- Hvilke krav fra tredjeparten til rutiner/praksis hos en selv som en er villig til å akseptere.
- Hvilket ansvar tredjeparten må påta seg.
- Hvilke forbehold som gjelder (f.eks. begrensninger i økonomisk ansvar).

Ikke-benekting er i enda sterkere grad enn autentisitet en meldingssikkerhetstjeneste og vil knapt være meningsfylt på transportlag eller lavere.

Det finnes andre former for ikke-benekting, bl.a.:
- *Ikke-benekting av mottak* betyr at mottaker må vedkjenne seg, ikke bare **at** en melding er mottatt, men også **hva** som er meldingens innhold. Denne tjenesten beskytter altså mot at mottaker påstår å ha mottatt et annet innhold enn det reelle.
- En enda kraftigere variant kalles av og til *ikke-benekting av innhold*, og innebærer at begge parter vedkjenner seg å være enige om hva som er utvekslet[2].

Ikke-benekting av mottak kan være aktuelt å stille som krav til MHS, men kan også være aktuelt å realisere på EDIFACT-nivå. Andre varianter må en i praksis regne med å realisere på EDIFACT- eller anvendelsesnivå, og dette er behandlet i veiledningens del 5.

Når EDI-anvendelsen krever ikke-benekting på EDIFACT- eller anvendelsesnivå (dvs. AUTACK, APERAK eller anvendelsesspesifikke EDIFACT-meldingstyper), er ikke-benekting på MHS-nivå vanligvis redundant. Derimot kan det være aktuelt å ha autentisering på lavere nivå, slik at uvedkommende trafikk kan avvises så tidlig som mulig, f.eks. som beskyttelse mot tjenestenektingsangrep.

[1] Det er et definisjonsspørsmål om situasjonen med ukjent motpart betraktes som autentisering eller ikke-benekting, men den praktiske konsekvensen er den samme: at tredjeparten tar et ansvar på vegne av motparten.

[2] Dette kravet opptrer når mottaker ikke kan behandle en melding før avsender er klar over at meldingen er mottatt.

6.3.7 Sekvensintegritet

Sekvensintegritet vil si sikkerhet for at mottaker ser den samme sekvensen av meldinger som avsender sendte, dvs.:

- Ingen meldinger går tapt.
- Ingen meldinger dupliseres.
- Meldinger mottas i samme rekkefølge som de ble sendt.

Det første kravet er vanligvis absolutt, og kan også ses på som et krav til pålitelighet. Dette diskuteres videre nedenfor. Graden av de to andre kravene varierer med anvendelsen. Duplisering eller rekkefølge av meldinger kan være helt uvesentlig. I andre tilfeller kan det være en viss kostnad forbundet med å rydde opp, men anvendelsen er ellers i stand til å håndtere det. Men det er også anvendelser hvor duplisering eller endring av rekkefølge er alvorlige sikkerhetsproblemer. (Hvis en ikke tar forholdsregler mot det, vil ofte en angriper være i stand til å avlytte en autentisk melding og så injisere en kopi av den senere, når den kan gjøre skade. Ved naiv bruk av meldingsbasert autentisering vil kopien også være ”autentisk”. Dette er en velkjent angrepsteknikk.)

Hvor god sekvensintegritet som kan oppnås, varierer også sterkt fra MHS til MHS. Det finnes sjelden eksplisitte, standardiserte mekanismer for å oppnå sekvensintegritet, men systemer som opererer ende til ende og synkront vil typisk ha bedre egenskaper enn systemer som opererer med lagring og videresending. Der kravene til sekvensintegritet er sterke, bør de trolig løses på anvendelsesnivå.

Sekvensintegritet (som en separat tjeneste) er først og fremst en meldingssikkerhetstjeneste. For kommunikasjonssikkerhet er det ikke særlig meningsfylt å skille mellom sekvensintegritet og integritet generelt.

Selv om disse begrepene ikke blir brukt, kan en i og for seg også snakke om sekvens-*autentisitet* og sekvens-*ikke-benekting*, analogt til hierarkiet integritet, autentisitet og ikke-benekting: Altså sikkerhet mot hhv. utilsiktet endring, bevisst angrep fra uvedkommende eller bevisst svindel fra den andre parten.

På meldingsnivå kan sekvensintegritet løses på to ulike måter:

- Ved å inkludere et monotont økende sekvensnummer, slik at mottatte meldinger kan behandles i strikt sekvensiell rekkefølge, dvs. en melding behandles ikke før alle meldinger med lavere sekvensnummer er mottatt og behandlet. (Total orden.)
- Ved å gi hver melding en unik identifikasjon og la hver melding referere tilbake til meldinger som forutsettes mottatt og behandlet før denne meldingen kan behandles. (Partiell orden.)

Innenfor en enkelt transportlagsforbindelse vil sekvensintegritet normalt være garantert. Men for en satsvis EDI-løsning vil normalt brukeragenten ikke ha noen innflytelse på når MTS setter opp nye transportlagsforbindelser og derfor ikke kunne garantere at forskjellige meldinger ”krysser hverandre" på forskjellige transportforbindelser.

6.3.8 Tjenestenekting

Tjenestenekting er en type angrep som tar sikte på å redusere tilgjengelighet, pålitelighet eller ytelse av en tjeneste, typisk ved å overbelaste den eller utnytte feil og svakheter i protokoller eller programvare. EDI-anvendelser vil vanligvis ikke ha direkte krav til motstandsdyktighet mot tjenestenekting, men i stedet stille krav til tjenestekvalitet. Systemer og nett som anvendelsene kjøres på, eller hos tjenesteleverandørene som benyttes, må så beskyttes mot tjenestenektingsangrep for å kunne oppfylle tjenestekvalitetskrav, f.eks. garantert tilgjengelighet.

6.3.9 Vertsmaskinsikkerhet

Ved å knytte seg til et mer eller mindre åpent kommunikasjonsnett vil partene løpe en viss risiko for angrep rettet mot maskiner og utstyr i sin alminnelighet, ikke bare selve EDI-anvendelsen. En EDI-anvendelse vil i første rekke stille generelle krav, på samme måte som andre anvendelser på en nettverkstilknyttet maskin, om at maskinen som anvendelsen kjører på (og programmer og data som hører til anvendelsen) skal være motstandsdyktig mot angrep. Det kan også være anvendelser hvor deler anses for å være så kritiske at det er uaktuelt å kjøre dem på en maskin som er direkte knyttet til kommunikasjonsnettet. I så fall må trusler og risiko vurderes mer spesifikt opp mot anvendelsen og hvordan denne kan splittes opp i mer og mindre kritiske moduler.

Hvilken grad av sikkerhet en kan oppnå på en vertsmaskin avhenger på en helt avgjørende måte av hvilke anvendelser som kjører der. EDI-anvendelsen vil ha krav som må veies opp mot risikoen for anvendelsen selv *og* for andre anvendelser/systemer:

- Adgang til og fra kommunikasjonsnettet, spesielt hvilke nettverkstjenester som må være åpne på vertsmaskinen.
- Adgang til og fra andre interne ressurser, kanskje på mer kritiske maskiner og nett.

Enkle EDI-anvendelser vil ha forholdsvis oversiktlige og håndterbare krav på disse områdene. Ekstern adgang til og fra maskinen kan begrenses til sending og mottak av elektronisk post. Selve MTS-tjenesten har forholdsvis lav risiko. (Angrep på transportagenter som utnytter program- eller installasjonsfeil forekommer, og det anbefales ikke å kjøre egen transportagent uten kompetent drift av den. Ved bruk av meldingslager hos tjenesteleverandør vil egne maskiner normalt utelukkende operere som klienter, og dette gir minimal risiko.)

Det å åpne for elektronisk post utenfra gir en viss risiko for datadrevne angrep (dvs. meldinger som inneholder virus, trojanske hester og lignende) Forutsetningen for slike angrep er at enkelte typer meldingsinnhold behandles automatisk av programmer med et visst potensiale for å gjøre skade. Slik sett har de fleste EDI-anvendelser en viss iboende risiko, altså ved at genuine meldingstyper kan misbrukes. EDIFACT-syntaks i seg selv er imidlertid meget trygg i forhold til andre innholdstyper som det er vanlig å utveksle i elektronisk post (f.eks. slike som kan inneholde makroer, script eller andre elementer med karakter av programmer.)

Slike risikable innholdstyper kan vanligvis ignoreres eller avvises. Men vær oppmerksom på muligheten for makrovirus i Word-dokumenter og tilsvarende andre verktøy med makromuligheter. PostScript, som i prinsippet er et

programmeringsspråk, kan inneholde kommandoer som f.eks. overskriver filer. HTML-dokumenter kan inneholde JavaScript. Slike, og andre risikable innholdstyper kan utveksles som assosierte objekter i en EDIFACT-utveksling, og hvis det er aktuelt må også risikoen ved det vurderes. Og husk at risikoen ikke er bevisste angrep fra forretningsforbindelsenes side, men at systemene deres selv blir offer for virusangrep.

Dersom anvendelsen krever kvitteringsmeldinger, spesielt på MTS- eller MHS-nivået, må en også være oppmerksom på muligheten for misbruk av disse mekanismene. Kvitteringsmeldinger kan brukes av angripere for å kartlegge maskiner bak en brannvegg eller som springbrett for å nå maskiner som ellers ikke ville være tilgjengelige for angriperen. Alle disse problemene kan en gardere seg mot hvis en er klar over det og kan konfigurere brannvegger, transportagenter og brukeragenter til å forhindre dem.
Kvitteringsmeldingene kan inneholde maskinnavn eller IP-adresser til maskiner bak en brannvegg, som en ellers ønsker å holde hemmelig. Angriperen kan også systematisk sende meldinger utenom den normale epost-tjeneren for å prøve å få direkte respons fra interne maskiner.
Brukernavn (for innlogging på maskinene) er ofte gyldige epost-adresser, men en vil vanligvis ikke opplyse disse til eksterne. Kvitteringsmeldingene kan lett komme til å inneholde brukernavn i stedet for de epost-adressene en ønsker å bruke utad. Angriperne kan også ofte finne ut hvilke reelle personer som skjuler seg bak alias og distribusjonslister (f.eks. ”root” eller ”postmaster”).
Meldingen som ber om å bli kvittert, bestemmer hvilken adresse kvitteringen skal sendes til. Denne adressen kan f.eks. referere en maskin innenfor brannveggen som angriperen ellers ikke ville ha anledning til å nå. (Eller hva med å sende ”spam” til et stort antall adresser, store internasjonale epost-lister osv. der kvitteringsadressen settes til å referere et uskyldig offer?)

6.4 Andre krav til tjenestekvalitet

6.4.1 Tilgjengelighet

Med tilgjengelighet menes krav til når og med hvor stor sannsynlighet tjenesten skal være i drift og mulig å nå, f.eks. formulert som krav til gjennomsnittlig oppetid eller maksimal nedetid. Satsvis EDI (med en lagre og videresend-modell for meldingsutveksling) har i utgangspunktet ikke direkte krav til tilgjengelighet, men indirekte må visse krav til tilgjengelighet være oppfylt for å nå krav om kapasitet og overføringstid. Dersom det benyttes synkrone tjenester (klient/tjener i en eller annen forstand) i MHS, kan det være nødvendig å stille spesifikke krav om tilgjengelighet til dem. Spesielt gjelder dette for brukere som ikke selv opererer en transportagent, men er avhengig av meldingslager hos en tjenesteleverandør.

6.4.2 Tap av meldinger

Tap av meldinger er vanligvis ikke akseptabelt, og det stilles krav om

- garantert levering, dvs. MHS påtar seg ansvaret for å hindre tap, eller
- deteksjon av tap, dvs. MHS varsler avsender, men tar ikke selv steg for å sende meldinger på nytt.

I det typiske tilfellet vil en ha en kombinasjon av begge typer krav. MHS vil garantere levering når tap skyldes f.eks. midlertidige feil, men i andre tilfeller bør det være opp til anvendelsen eller ansvarlige personer å håndtere feil.

Det er vanligvis mer alvorlig å tape meldinger enn å duplisere dem. Derfor vil feiltolerante systemer ofte utformes slik at feil fører til duplisering, ikke tap.

Det er flere faktorer som kan gi risiko for tap av meldinger:

- Direkte kommunikasjonsfeil.
- Feil i programvare, utstyr eller konfigurasjon av dem.
- Manglende mellomlagringskapasitet.

6.4.3 Kapasitet

EDI-anvendelser vil stille svært varierende krav til kapasitet i MHS. Det gjelder både størrelsen på enkelt-utvekslinger og gjennomsnittlig trafikkvolum (gjennomstrømming). Faktorer som er avgjørende for kapasiteten, er bl.a.:

- Transmisjonshastighet (båndbredde) og tidsforsinkelse på kommunikasjonskanalene.
- Antall utvekslinger (pga. faste kostnader per utveksling, f.eks. innlogging mot tjenesteleverandør, autentisering og andre sikkerhetsfunksjoner, protokolloverhead.)
- Mellomlagringskapasitet.
- Pålitelighet: Overføringer som må gjentas pga. feil kan spise opp betydelig transportkapasitet.

6.4.4 Overføringstid

Med overføringstid menes her tiden som løper fra en melding leveres til MHS (dvs. avsenders brukeragent) til den er tilgjengelig for mottakers EDI-funksjon (dvs. er mottatt og behandlet av mottakers brukeragent). Overføringstiden er avgjørende for hvor raskt den transaksjonen meldingen inngår i kan gjennomføres. En grov kategorisering av krav til overføringstid er:

- Umiddelbar (synkron) overføring: Interaktive eller transaksjonsorienterte anvendelser med krav til toveis kommunikasjon innenfor en tidsramme på noen få sekunder eller mindre. Dette krever oftest bruk av synkrone overføringstjenester, f.eks. filoverføring eller HTTP.
- Normal overføring: Meldinger overføres så raskt som mulig, uten vesentlig ventetid. Tidsrammen vil være ned mot noen få minutter.
- Overføringstid ikke kritisk: Overføringen av meldinger kan forsinkes, typisk for å redusere kostnadene for overføringen, eller fordi sender eller mottaker ikke kontinuerlig er tilknyttet kommunikasjonssystemet.

Det er en sammenheng mellom overføringstid, kapasitet og kostnader. Kapasiteten i kommunikasjonssystemet kan vanligvis utnyttes bedre dersom kravene til overføringstid slakkes. Ved å holde tilbake meldinger en stund kan de buntes sammen i større utvekslinger. Det kan være mulig å utnytte lavtrafikkperioder når tjenestene er lavere priset (f.eks. telefon på kveld/natt). Tjenesteleverandør kan ha differensierte priser etter trafikkens prioritet.

6.5 Andre krav

6.5.1 Kvittering og feilmeldinger

Kvitteringsmekanismer på MHS- eller MTS-nivå er vanligvis bare et middel til å oppnå god nok pålitelighet. Kvitteringsmekanismer er spesielt viktige i forbindelse med deteksjon og oppretting av feil, under utprøving og idriftsetting av programvare og tjenester, eller ved etablering av kommunikasjon mot nye parter.

- Muligheten for å gi og behandle negative kvitteringer er et absolutt krav. Feil som fører til tap av meldinger må alltid rapporteres, dvs. hvis MTS ikke er i stand til å levere meldingen, eller hvis meldingen inneholder feil som gjør at brukeragenten ikke kan behandle den videre. Andre unormale situasjoner bør også kunne rapporteres, f.eks. unormale forsinkelser.
- Positive kvitteringer bør i de fleste tilfeller ikke benyttes under vanlig drift. Unntaket er ”ikke-benekting av mottak” og lignende sikkerhetskrav, i de tilfellene disse overhodet håndteres på MHS-nivå. (Ved ukritisk bruk av positive kvitteringer vil hver melding utløse en kaskade av lite informative kvitteringsmeldinger.) Det bør imidlertid være mulig å velge å benytte positive kvitteringer når det er behov for det, f.eks. under utprøving og feilretting.

Ved bruk av synkrone eller ende til ende overføringstjenester ligger kvitteringen vanligvis i protokollen mellom klient og tjener, og det er mindre behov for egne kvitteringsmeldinger.

Merk at kvitteringsmekanismer på kommunikasjonsnivået i prinsippet bare kan rapportere kommunikasjonsstatus for utvekslingen som helhet, men ikke hvordan mottaker eller andre parter stiller seg til innholdet i utvekslingen. Positive kvitteringer betyr f.eks. ikke at mottaker ikke kan forkaste utvekslingen pga. syntaktiske feil på EDIFACT-nivå.

Kvitteringer blir av og til benyttet for direkte å kompensere for et upålitelig MTS: Sendte utvekslinger lagres inntil positiv kvittering er mottatt, og retransmitteres automatisk dersom kvittering ikke er mottatt innen en viss tid. Denne praksisen er problematisk:

- Tap av data pga. midlertidige feil skal normalt rettes opp av protokoller på lavere nivå.
- Tap av meldinger i MHS skyldes ofte mer permanente problemer (f.eks. feilkonfigurering), som må rettes manuelt, og automatisk retransmittering fører i beste fall bare til unyttig trafikk.

Med andre ord, det er tre forskjellige tilfeller som kan forekomme ved feilsituasjoner:

1. Feilen rettes internt i MTS, uten at avsender eller mottaker er involvert. Midlertidige nettverksfeil, overbelastning, transportagenter som krasjer eller midlertidig er ute av drift er eksempler på feilsituasjoner som bør håndteres på denne måten.

2. Feilsituasjonen er midlertidig som over, men rapporteres likevel tilbake til avsender uten å bli rettet internt i MTS. Det er sannsynlig at avsender kan sende meldingen på nytt noe senere, uten å gjøre noen endringer.

3. Feilsituasjonen er relativt permanent, altså at det er sannsynlig at avsender eller andre må foreta seg noe for at det skal ha noen hensikt å sende meldingen på nytt. (F.eks. endre adresser eller åpne adgang til tjenester.)

Bruk av kvitteringsmeldinger for automatisk feilhåndtering har bare noe for seg for type 2 situasjoner. Men det er rimelig å stille krav om at tjenester og programvare er slik at disse situasjonene ikke oppstår (altså hånteres automatisk som i tilfelle 1), eller oppstår så sjelden at det er uproblematisk å håndtere dem manuelt som for tilfelle 3.

6.5.2 Operative forhold

De fleste EDI-anvendelser vil ha behov for et MHS som opererer med mellomlagring og videresending, slik at avsender og mottaker ikke er bundet til å operere synkront på EDI-nivå. Minst en av partene, avsender, mottaker eller tjenesteleverandør må operere en tjener, dvs. en maskin som er mer eller mindre kontinuerlig tilgjengelig og avventer oppkobling fra klientene.

I mange tilfeller er det ikke aktuelt for avsender eller mottaker å operere tjenere, og dette faller på en tjenesteleverandør:

- Det er for kostnadskrevende å holde en tjener-maskin kontinuerlig tilgjengelig.
- Bruker-organisasjonen har ikke ressurser eller kompetanse til å håndtere de sikkerhetsmessige aspektene.
- Brukeren ønsker selv å ha initiativet for når oppkobling og kommunikasjon skjer.

For noe større organisasjoner er disse momentene mindre vesentlige.

De fleste transportagenter og MTS-protokoller ”skyver” meldinger fra seg, dvs. det er den som har en melding å sende eller videresende som tar initiativet og opererer som klient, mens den som overtar meldingen må operere som tjener. Dette er imidlertid ikke alltid tilfelle, f.eks. ved bruk av meldingslager, der mottakers brukeragent opererer som klient. Det samme gjelder ofte ved bruk av filoverføringsprotokoller. Enkelte av protokollene mellom transportagenter tillater også at mottaker tar initiativet og ”trekker” meldinger til seg, f.eks. SMTP hvor klienten kan bytte rolle med tjeneren.

Merk at det også er behov for andre typer tjenester enn MTS-tjenesten i seg selv:

- Navnetjenester, f.eks. DNS i Internett, for oversetting mellom MHS-adresser og nettverksadresser.
- Ruting av trafikk på nettverksnivå.
- Navnetjenester for oversetting mellom EDI-identifikatorer og MHS-adresser.
- Sertifiseringstjenester, tidsstempling og andre TTP-tjenester.
- Konverteringstjenester.

For de fleste av disse er det snakk om både selv å være tilgjengelig/registrert i tjenesten og å kunne få tilgang til registreringene som gjelder dem en skal utveksle data med. For katalogtjenester i forbindelse med sertifisering vil f.eks. et vanlig tilfelle være at brukeren forholder seg bare til katalogen hos sin egen sertifiseringsautoritet. Denne tjenesteleverandøren opererer også en portner som gir egne brukere tilgang til sertifikater og annen informasjon i andre

sertifiseringsautoriteters kataloger. I andre sammenhenger er det mer aktuelt for EDI-brukerne å benytte andre leverandørers eller parters tjenester mer direkte.

Noen av disse tjenestene vil EDI-anvendelsen stille nokså direkte krav til. Andre har betydning for f.eks. pålitelighet og tilgjengelighet. Som for selve MHS-tjenesten, kan disse tjenestene leveres av tredjepartsleverandører eller av EDI-brukerne selv. Forskjellige tjenester kan leveres av forskjellige leverandører.

6.5.3 Drift og overvåkning

Alle EDI- og kommunikasjonsløsninger krever en eller annen form for driftsorganisasjon. I de aller enkleste tilfellene er det kanskje tilstrekkelig med en person som er tilgjengelig for å svare på henvendelser fra andre parter, mens en part som gjør nettverkstjenester tilgjengelig i stor skala antakelig trenger funksjoner som ”brukerstøtte”, feilmeldingssenter el.l. som er mer eller mindre kontinuerlig bemannet.

En EDI-løsning med programvare i form av en ”pakkeløsning” fra en tjenesteleverandør, som kjøres hos en bruker med liten eller ingen teknisk kompetanse, har ikke uten videre mindre behov for en gjennomtenkt drifts- og overvåkningsfunksjon enn en stor og tung løsning som er utviklet og vedlikeholdt internt hos den organisasjonen som bruker den.

Eksempler på manuelle driftsoppgaver:

- Rutinemessige manuelle operasjoner (som inngår i normal drift), f.eks. sikkerhetskopiering, dataoverføringer som krever manuelle inngrep.
- Manuell rekonfigurering, f.eks. for å innlemme nye brukere eller parter, endringer av adresser, ruting, generering og installasjon av kryptografiske nøkle, oppgradering av programvare.
- Koordinering mot de andre partene, assistanse til de andre partenes driftspersonell.
- Administrasjon, overvåkning, feildeteksjon og feilretting.

Overvåkning betyr i denne sammenhengen *systematisk innsamling av driftsrelaterte data* (fra et operativt system). I praksis betyr det gjerne et systematisk opplegg for å samle inn og logge (feil)meldinger fra ulike systemkomponenter. Innsamling av statistikk kan av og til kreve spesiell instrumentering av systemene. Overvåkningen har flere formål:

- Feildeteksjon (prioritering og varsling).
- Dimensjonering: Finne og fjerne ytelsesmessige flaskehalser, men samtidig unngå sitte med overdrevet kapasitet på punkter der det ikke er behov for det.
- Sporbarhet og revisjon: Ha nok informasjon til å kunne gjenskape hendelsessekvenser, f.eks. i tvistesituasjoner eller for å kunne rekonstruere data etter feilsituasjoner.

Overvåkning kan til en viss grad automatiseres. Utfordringen er gjerne å samle nok data til de behov en har, uten at mengden av data blir helt uhåndterlig. Mange programvareprodukter kan konfigureres til varierende grad av logging. Det finnes også egne overvåkningsverktøy, som kan utføre oppgaver som å gjennomgå og filtrere logger (skille ut aktuelle meldinger), motta og distribuere alarmsignaler (f.eks. såkalte SNMP traps), gi operatører adgang til å inspisere og kontrollere systemer fra et sentralt administrasjonskonsoll.

Fleksibilitet er ofte svært vesentlig, fordi hvilken informasjon som er aktuell og vesentlig avhenger av situasjonen. Driftspersonell kan ha behov for å endre dynamisk på reglene som bestemmer hva og hvor mye som logges, og hvor og hvordan dette presenteres. I praksis vil det ofte være snakk om automatisk å klassifisere "hendelser" i en håndfull kategorier, som så gis forskjellig behandling, f.eks.:

- Katastrofale feil: Automatisk oppringning til personsøker, alarm til administrasjonskonsoll etc.
- Alvorlige feil: Meldes til rette vedkommende uten opphold (f.eks. meldinger til konsoll)
- Advarsler: Unormale hendelser som indikerer potensielle feil og som må vurderes av en person med den rette kompetansen.
- Informasjon: Hendelsen logges av hensyn til sporbarhet, men i en logg som ikke gjennomgås før behovet oppstår.

M.a.o:

- **Hvor alvorlig** er hendelsen?
- **Hvor raskt** må noen ta seg av den?
- **Hvem** er rette vedkommende til det?

Fjernovervåkning baseres ofte på SNMP-protokollen (Simple Network Management Protocol). De fleste nettverkskomponenter (rutere, tjenere osv.) for Internett-protokoller har SNMP-agent innebygget slik at de uten videre kan administreres/overvåkes på nettverks- og transportnivå. Også tjenester og applikasjoner kan overvåkes gjennom SNMP, men det krever ofte noe mer tilrettelegging. For større organisasjoner som har egne SNMP-baserte nettovervåkningsoppsett, vil det være naturlig å stille krav om at EDI-løsningen også kan overvåkes gjennom SNMP. I så fall må det spesifiseres hvilke variable som skal kunne overvåkes eller påvirkes, altså innholdet i MIB (management information base).

Slike overvåkningsopplegg må ses i sammenheng med krav til tilgjengelighet og sporbarhet, og de er selvfølgelig mest aktuelle for tjenesteoperatører som har eksterne krav på seg. Men alle automatiserte prosesser som inngår i en EDI-løsning må kunne overvåkes. Den eller de ansvarlige må kunne forvisse seg om at prosessen foregår som den skal, og sikre seg at alle vesentlige avvik blir registrert.

EDI har også noen spesielle driftsmessige utfordringer:

- Anvendelsene spenner over flere uavhengige organisasjoner. Feildeteksjon og –retting vil ofte kreve samarbeid på tvers av organisasjonsgrenser, enten det er fordi den ene parten oppdager feil som må rettes hos den andre, eller fordi feilen skyldes uheldige sammentreff av forhold hos flere av partene. Feilretting eller oppgradering av programvare kan berøre parter som ikke selv var berørt av den opprinnelige feilen, og bare det å få utplassert korrigerte programvareversjoner kan i seg selv kreve betydelig planlegging og koordinering.
- Også innenfor en enkelt organisasjon kan flere uavhengige organisasjonsenheter være involvert. Ansvaret for drift av sentrale nett og tjenere, eksterne tilknytninger (brannvegger etc.), sikkerhet, EDI-anvendelser, applikasjoner osv. kan ligge på forskjellige steder i organisasjonen.

- Jo mer automatisert, jo mindre grad det er personer ”i løkken”, jo lettere er det at unormale situasjoner eller i og for seg banale feil forblir uoppdaget og konsekvensene forplanter seg.

A Terminologiliste

ACSE	Association Control Service Element Tjenester i OSI for å sette opp en ”assosiasjon”, dvs. omtrent det samme som en forbindelse (s.d.). (Men ACSE ligger i applikasjonslaget, og assosiasjonen kan altså bruke tjenester fra sesjons- og presentasjonslaget i tillegg til transporttjenester.)
API	Application Programming Interface Programgrensesnitt.
Applikasjon	1) I datakommunikasjonssammenheng: Protokoller og programmer som realiserer bruker-rettede tjenester, som f.eks. meldingsutveksling. 2) I denne veiledningen har vi brukt applikasjon om de programsystemene som sett fra EDI-anvendelsen er det endelige opphavet for eller endelig mottaker for utvekslede data, altså de interne systemene hos partene der behandlingen av data egentlig skjer.
ASCII	American Standard Code for Information Interchange (Kodet) tegnsett med tegnkoder i området 0-127 (7-bit). Se også ISO 646.
Asynkron	”Ikke samtidig”, i vår sammenheng spesielt brukt om asynkron meldingsutveksling, hvor prosessene ved sending og mottak av en melding løper uavhengig av hverandre og til forskjellige tider, og meldingsutvekslingen er enveis (dvs. en evt. svarmelding kommunikasjonsmessig betraktes som en ny, uavhengig melding).
ATM	Asynchronous Transfer Mode Et sett med protokoller for ”celle-svitsjede” datanett, hvor trafikken brytes opp i små ”celler” av fast størrelse, i motsetning til linje- eller pakkesvitsjede nett.
Autentisitet	Sikkerhetstjeneste som gir tillit til at innholdet i et dokument eller en datablokk både er korrekt og stammer fra en bestemt avsender, dvs. beskytter mot forfalskning utført av uvedkommende.
Autentisering	Operasjon med formål å etablere autentisitet for data eller bestemme identiteten for en kommunikasjonspart på en sikker måte. (Brukes både om operasjoner utført av parten hvis identitet skal bestemmes og av motparten. ”A autentiserer seg overfor B” og ”B autentiserer A”. ”A autentiserer meldingen M” kan bety både at A genererer en digital signatur eller lignende på M og at A verifiserer signaturen.)
BMP	Basic Multilingual Plane Delmengden av UCS (ISO 10646, s.d.) med tegnkoder under 2^{16}, og som er sammenfallende med UNICODE (s.d.)

Brannvegg	Ruter eller portner som danner grensen mellom et mer åpent nett (f.eks. Internettet totalt) og et mer lukket nett (f.eks. et bedriftsinternt nett), og som utfører en eller annen form for sikkerhetskontroll på trafikken som passerer den.
Brannmur	Det samme som brannvegg.
Brukeragent	(UA, User Agent) Program som kjører på vegne av brukeren for å motta og sende meldinger fra/til et meldingstransportsystem.
BP14	Innholdstype ”bilaterally-defined” i IPM, dvs. hvor tolkningen av innholdet er opp til nærmere avtale mellom partene.
BP15	Innholdstype ”externally-defined” i IPM, dvs. hvor parametre (objekt-identifikator) refererer til en annen standard eller spesifikasjon som definerer innholdet.
Datagram	En enkelt, selvstendig adressert datapakke/-blokk i en forbindelsesløs tjeneste. Se forbindelsesløs.
Datagramorientert	Se forbindelsesløs.
Digital signatur	Mekanisme som brukes for å realisere sikkerhetstjenestene autentisitet og ikke-benekting. Signaturen genereres med en hemmelig (privat) nøkkel ut fra innholdet i dokumentet (datablokken) som signeres, og kan ikke forfalskes uten kjennskap til denne nøkkelen. For verifikasjon av signaturen brukes den tilsvarende offentlige nøkkelen.
DNS	Domain Name System Katalogtjenesten som brukes for å oversette domene-navn til IP-adresser i Internettet.
DR	Delivery Report (Positiv) leveringsbekreftelse (X.400).
DSN	Delivery Status Notification (Positiv eller negativ) leveringsbekreftelse (Internett).
EDIINT	Profiler for bruk av Internett-protokoller for (SMTP, HTTP) for utveksling av EDI-meldinger.
EDIM	EDI Message Meldingstypen definert i X.435 for transport av EDI-meldinger, også betegnet P_{edi} eller P_{35}.
EDIN	EDI Notification (Positiv eller negativ) mottakskvittering (X.435).
Ekstranett	Virtuelt nett (gjerne realisert gjennom mekanismer for transport- eller nettverkslags-sikkerhet, VPN) som forbinder en bedrifts interne nett med eksterne kunder, utplasserte medarbeidere etc. over Internettet.

ESMTP — Extended SMTP
Standard for å be om/annonsere tjenesteutvidelser i forhold til basis SMTP.

Forbindelse — Connection
Knytning mellom to transport-endepunkter i en forbindelsesorientert transportprotokoll, som fungerer som en kanal for datatrafikk (en virtuell linje). I TCP er f.eks. en forbindelse identifiser ved (avsenders IP, avsenders port, mottakers IP, mottakers port).

Forbindelsesorientert
En transporttjeneste (f.eks. TCP) karakteriseres som forbindelsesorientert når det settes opp en varig forbindelse mellom transport-endepunktene før data utveksles. Forbindelsesorienterte transporttjenester tilbyr vanligvis dataintegritet: Så lenge transportforbindelsen er etablert, garanteres det at data mottas identisk som sendt, og i samme rekkefølge. Motsatt: Forbindelsesløs eller datagramorientert. (”Forbindelsesorientert” brukes mest på transportlaget, ”virtuelt linjesvitsjet” brukes mest på nettverkslaget, men betegnelsene brukes noe om hverandre.)

Forbindelsesløs
En transporttjeneste (f.eks. UDP) karakteriseres som forbindelsesløs (eller datagramorientert) når data utveksles uten at det på forhånd er satt opp noen forbindelse, dvs. hver enkelt datablokk (datagram) som sendes adresseres eksplisitt til et transport-endepunkt. Forbindelsesløse tjenester garanterer oftest integriteten for hvert enkelt datagram når de er mottatt, men kan ikke gi garantert levering eller sekvensintegritet.

FTAM — File Transfer Access Method
Protokoll for filoverføring (OSI).

FTBP — File Transfer Body Part
Innholdstype i IPM for overføring av binære filer med assosierte attributter.

FTP — File Transfer Protocol
Protokoll for filoverføring (Internett).

HARPOON — Innkapsling av meldinger i MIME-format for transport gjennom et X.400-nett, hvor MIME-meldingen representeres direkte (tekstlig) i en X.420 IA5-innholdsdel.

HTTP — Hypertext Transfer Protocol
Transportprotokollen for Web-tjenesten, fundamentalt en synkron meldingsutvekslingsprotokoll.

Ikke-benekting — Sikkerhetstjeneste som gir de samme garantier som autentisitet (s.d.), og i tillegg at avsender ikke kan benekte å ha sendt/utstedt de data som er dekket.

IMAP4 — Internet Message Access Protocol (version 4)
Internett-protokoll for bruk av meldingslager (jfr. POP3).

Innhold	Content, Payload Den delen av en datapakke, melding etc. som ikke tolkes av protokollen på dette nivået, altså oppfattes som en ren bitsekvens som skal overføres uten endringer.
Innholdsdel	Selvstending del av en (MHS-)melding. Typisk et enkelt dokument, en fil, en EDIFACT-utveksling.
Innholdstype	Beskrivelse av format og formål/betydning av en innholdsdel, jfr. meldingstype.
Integritet	Sikkerhetstjeneste som garanterer korrekthet av data, så lenge det ikke er noen bevisste forsøk på forfalskning.
Internett	Et nett som består av flere, relativt autonome enkeltnett som er knyttet sammen med rutere/portnere.
Intranett	1) Et bedriftsinternt nett (gjerne bestående av sammenknyttede avdelingsnett) basert på Internett-teknologi. 2) Tjenester og applikasjoner i 1)
IP	Internet Protocol Nettverksprotokollen i Internettet.
IP-adresse	(Numerisk) adresse som entydig identifiserer en node i Internettet.
IPv4	IP versjon 4 Nåværende versjon av IP.
IPv6	IP versjon 6 Neste versjon av IP, gir bl.a. større adresserom, bedre støtte for mobile enheter, bedre støtte for håndtering av tjenestekvalitet.
IPM	Interpersonal Message Meldingstypen definert i X.420 for transport av person- til person-meldinger (dvs. ordinær elektronisk post), også betegnet P_2.
IPsec	IP security Samling av protokoller for sikkerhet på nettverkslaget i Internett (se også virtuelt privat nett)
IA5	Kodet tegnsett som stort sett er sammenfallende med ASCII. (X.420 har en innholdsdel-type som består av IA5-kodet tekst.)
ISDN	Integrated Services Data Network Digital tjeneste som erstatter analog telefonlinje.
ISO 646	Standard som definerer internasjonale (7-bits) versjoner av ASCII, som varianter av ISO 646 IRV (International Reference Version). Norsk standard 4115 definerer to norske varianter der bl.a. hake- og klammeparenteser i ASCII/ISO 646 IRV er erstattet med æ, ø og å. Siden disse tegnsettene ikke er kompatible med hverandre, er de nå stort sett uaktuelle.

ISO 2022 — Standard som definerer mekanismer (styretegn) for å identifisere og skifte mellom forskjellige tegnsett i samme tegnstreng. Nå stort sett uaktuell, i stedet brukes UNICODE eller ISO 10646.

ISO 8859 — Standard som definerer en rekke 8-biters tegnsett (strengt tatt, tegnsett som kan brukes enten som "øvre halvdel" av 8-biters tegnsett eller som skift-tegnsett i ISO 2022), deriblant ISO 8859-1 som dekker vesteuropeiske språk (med visse unntak).

ISO 10646 — Standard som definerer et universelt kodet tegnsett (UCS) der hvert eneste kjent tegn har en unik 32-biters kode. Kodeverdiene er sammenfallende med UNICODE (16-biters koder), ISO 8859-1 (8-biters koder) og ASCII (7-biters koder). Ulike tegnkodingsmekanismer brukes for å representere tegnene i en mer kompakt form. (UCS-2, UTF-8, UTF-16)

Klient — 1) Program/prosess som benytter tjenester fra en tjener (s.d.)
2) Maskin som primært kjører klient-programvare.
Klienten er den aktive parten, initiatoren, den som tar initiativet til interaksjonen.
Merk: Klient og tjener er roller, samme prosess eller maskin vil ofte opptre i en klientrolle overfor noen av de prosessene/maskinene den kommuniserer med og en tjenerrolle overfor andre.

Konfidensialitet — Sikkerhetstjeneste som beskytter mot uvedkommendes innsyn i data.

Latin-1 — Navn på tegnsettet som defineres i ISO 8859-1.

Leveringsbekreftelse — Kvitteringsmelding fra meldingstransportsystemet som bekrefter om en melding er levert (eller ikke) til mottakeren.

Lukket nett — Datanett som ikke er alment tilgjengelig, ved at det er restriksjoner på hvilke parter som kan knytte seg til det. (Jfr. åpent nett.) Nett kan oppfattes mer eller mindre lukkede avhengig av hvor sterke restriksjoner som finnes.

MDN — Message Disposition Notification
Mottakskvittering (Internett).

Melding — 1) Enheten som utveksles i meldingsutvekslingssystemet.
2) En EDIFACT-melding.

Meldingslager — Tjeneste som tilbys av meldingsutvekslingssystemet, hvor meldinger mellomlagres hos tjenesteleverandør, på en sentral tjener i organisasjonen eller lignende, og endelig mottak skjer først på mottakerens initiativ (når mottakeren kobler opp mot meldingslageret).

Meldingsutvekslingssystem — (MHS, Message Handling System) Systemet som består av brukeragenter og transportagenter og har som formål å utveksle meldinger fra bruker til bruker (EDI-løsning til EDI-løsning.)

Merk: Betegnelsen brukes normalt om protokoller som X.400 og SMTP med en ”lagre og videresend”-modell. Siden så godt som all meldingsutveksling i EDI-sammenheng faller inn under denne modellen, bruker vi betegnelsen (og de relaterte MTS, MTA, UA) i generelle sammenhenger i dette heftet, selv om meldingsutvekslingen kan være realisert med andre typer protokoller. Det er altså ikke hensikten å utelukke f.eks. filoverføringsprotokoller med denne språkbruken.

Meldingstransportsystem
(MTS, Message Transport System) Systemet som består av alle transportagenter og har som formål å transportere meldinger fra avsenders til mottakers maskin.

Meldingstype
Definisjon av format og formål/betyding av en melding, vanligvis eksplisitt identifisert i meldingen selv. Brukes både om meldingstyper på MHS-nivå (RFC822/MIME, IPM, EDIM) og EDIFACT-nivå. Se også innholdstype.

MIB
Management Information Base
Database eller databaseskjema som beskriver variable som er tilgjengelige for fjernovervåkning på en node.

MIME
Multipurpose Internet Mail Extensions
Meldingsformat basert på RFC822 for identifikasjon av innholdstyper samt transport av ikke-tekstlige data.

MIXER
MIME Internet X.400 Enhanced Relay
komm4.dockomm4.docSpesifikasjon for portner mellom X.400 og Internett-epost.

Mottakskvittering
Kvitteringsmelding fra mottakeren som bekrefter om mottakeren aksepterer å ha mottatt meldingen eller ikke.

MTA
Message Transfer Agent
Transportagent.

MTS
Message Transfer System
Meldingstransportsystem.

NDR
Non-delivery Report
(Negativ) leveringsbekreftelse (X.400).

Nett
Alle maskiner som kan nå hverandre direkte, f.eks. fordi de er knyttet til samme Ethernett, uten at trafikken må passere en ruter, er i Internett-sammenheng knyttet til samme nett. (Nettene kan være realisert med mange ulike nettverksteknologier.)

Nettverkslag
Det protokoll-laget i et datanett som håndterer nettverkstopologien, altså hvordan nodene i nettet er forbundet.

NRN
Non-receipt Notification
(Negativ) mottakskvittering (X.420)

Offentlig nøkkel — Den av nøklene i et nøkkelpar for offentlig nøkkel-kryptografi som offentliggjøres, og brukes for kryptering eller signaturverifikasjon.

Offentlig nøkkel kryptosystem
Kryptosystem der kryptering og dekryptering (signaturgenerering og verifikasjon) skjer med distinkte nøkler, og den ene holdes hemmelig (privat nøkkel) mens den andre kan offentliggjøres (offentlig nøkkel).

OSI — Open Systems Interconnection
Bredt initiativ for å utvikle standard nettverksprotokoller i ISO/ITU-regi. (Bl.a. opphavet til den velkjente 7-lags referansemodellen. Protokollene selv er etter hvert nokså uaktuelle.)

Part — Ansvarlig organisasjon eller person som har en rolle i et kommunikasjonsforhold, f.eks. en EDI-anvendelse. (M.a.o. tekniske entiteter som maskiner, programmer osv. er ikke parter, men de drives av og opererer på vegne av partene.)

PKCS — Public-Key Cryptography Standard
Spesifikasjonsserie utgitt av RSA Security Inc. (tidligere RSA Data Security Inc.)

PKCS#7 — Spesifikasjon (i PKCS-serien) for en meldingsformat for å kryptere og/eller digitalt signere et vilkårlig innhold (sikkerhetssyntaks).

POP3 — Post Office Protocol (version 3)
Internett-protokoll for bruk av meldingslager (jfr. IMAP4).

Port — Endepunkt for UDP eller TCP-trafikk. Porter identifiseres med (16-biters) portnumre. All trafikk, enten UDP-datagrammer eller TCP-forbindelser, adresseres til en bestemt port på en bestemt maskin (og sendes også fra en bestemt port). (Se også transport-endepunkt.)

Portner — Node i et datanett som foretar oversetting mellom protokoller.

Privat nøkkel — Den av nøklene i et nøkkelpar for offentlig nøkkel-kryptografi som holdes hemmelig, og brukes for dekryptering eller signaturgenerering.

Profil — Standard/spesifikasjon som gir nærmere detaljer om hvordan en annen standard/spesifikasjon skal anvendes i en nærmere definert sammenheng.

Program — Sekvens av instruksjoner som kan utføres av en (fysisk eller virtuell) maskin. (Se også prosess.)

Programgrensesnitt
(API: Application Programming Interface)
Program til program-grensesnitt, realisert ved subrutine- eller metodekall fra en programmodul til en annen. (I motsetning til et protokollbasert grensesnitt, hvor modulene ligger i separate prosesser og utveksler data.)

Prosess	En aktiv utførelse av et program. (Ett og samme program kan utføres av flere uavhengige prosesser på samme maskin.)
Proxy	”Stedfortreder” 1) Lokalt objekt/lokal enhet som representerer et annet objekt/en annen enhet et annet sted i nettet, og som en lokal klient kommuniserer direkte med. 2) (Element i) en brannvegg som brukes som stedfortreder for en tjeneste innenfor eller utenfor brannveggen, for å unngå at lokale klienter kommuniserer direkte med eksterne (og muligens ondsinnede) tjenere, eller vice versa.
RFC	Dokumentserie som omfatter Internett-standarder og andre registrerte dokumenter i Internett-sammenheng.
RFC822	Det grunnleggende meldingsformatet for Internett epost og andre meldingsorienterte tjenester i Internett. (Har bare navn etter RFC-en som definerer det.)
RSA	Offentlig nøkkel-kryptosystem, som kan brukes både for digital signatur og konfidensialitet. (Oppkalt etter oppfinnerne: Rivest, Shamir, Adleman)
RN	Receipt Notification (Positiv) mottakskvittering (X.420)
Ruter	Enhet som opererer på transportlaget og utveksler IP-pakker mellom forskjellige nett.
Sekvensintegritet	Sikkerhetstjeneste som gir garantier om at meldinger mottas i samme rekkefølge som mottatt og uten tap eller duplisering.
Sertifikat	Digitalt signert melding som bekrefter at en gitt offentlig nøkkel tilhører en bruker el.l. med et gitt navn, utstedt av en tiltrodd sertifiseringsautoritet.
Sikkerhetspolicy	En overordnet men fullstendig og spesifikk beskrivelse av sikkerhetsregler som gjelder i en nærmere definert sammenheng, f.eks. for en organisasjon, et nett, en maskin, en EDI-anvendelse.
Sikkerhetssyntaks	Meldingsformat for å representere digitalt signerte og/eller krypterte meldinger.
S/MIME	Standard som definerer en sikkerhetssyntaks (avledet fra PKCS#7) og hvordan denne skal transporteres i MIME-formaterte meldinger.
SMTP	Simple Mail Transfer Protocol Protokoll for overføring av epost-meldinger til Internett MTA.
SNMP	Simple Network Management Protocol Protokoll for fjernovervåkning/administrasjon av Internett-noder.

SNMP-agent — Tjener som kjører på en node som er administrert via SNMP og gjør data fra noden tilgjengelig.

SNMP trap — Signal som sendes uoppfordret fra en SNMP-agent, f.eks. alarm ved feilsituasjoner.

Socket — (Ikke noe veletablert norsk oversettelse)
Programvare-konstruksjon som brukes for å nå et transport-endepunkt (sende eller motta data) i et svært utbredt API for nettverksprogrammering. (Brukes primært for TCP og UDP, men er i prinsippet protokolluavhengig.)

SSL — Secure Socket Layer
Protokoll for å realisere transportlagssikkerhet i Internett (jfr. TLS)

Synkron — ”Samtidig”, i vår sammenheng spesielt brukt om synkron meldingsutveksling, der prosessene ved sending og mottak av meldinger løper parallelt og avhengig av hverandre, og hver interaksjon mellom sender/mottaker (klient/tjener) typisk består av en melding (forespørsel) den ene veien fulgt av en svarmelding (respons) den andre veien.

TCP — Transport Control Protocol
Protokollen som tilbyr en forbindelsesorientert transporttjeneste i Internett.

Tegn — Minste betydningsbærende enhet i et skriftspråk. (NB! Tegn er ikke det samme som grafisk representasjon. Samme tegn kan ha forskjellige grafiske representasjoner, avhengig av valgt skrifttype eller sammenhengen det står i, og omvendt kan flere tegn dele samme grafiske representasjon, som f.eks. latinsk A, kyrillisk A og gresk stor alfa.)

Tegnkoding — Character Encoding
Mekanisme for konkret representasjon av tegnkoder, hvor hvert tegn representeres enten ved et fast eller variabelt antall oktetter (8-biters byte).

Tegnsett — (Coded) Character Set
En mengde av entydig identifiserte tegn. (Evt. ”kodet tegnsett” for å presisere at hvert tegn er identifisert gjennom en tegnkode, men dette er er vanligvis underforstått. Tegnene er imidlertid ikke nødvendigvis representert direkte gjennom tegnkodene, se tegnkoding.)

Tiltrodd tredjepart
Organisasjon/tjenesteleverandør som tilbyr en tjeneste som er avhengig av tiltro fra begge parter i et kommunikasjonsforhold. (Typisk for å etablere tillit mellom opprinnelig ukjente parter, for å bevare partenes hemmeligheter for hverandre eller for å styre gjennomføringen av en transaksjon når partenes egeninteresser gjør det uakseptabelt for den ene eller andre å ta styringen. Vi bruker normalt ikke begrepet om en tjenesteleverandør som opererer på vegne av den ene parten, og som den andre ikke har noe spesielt tillitsforhold til.)

Tjener	1) Program eller prosess som tilbyr tjenester til klienter (s.d.) 2) Maskin som primært kjører tjener-programvare (tjenestemaskin). Tjenere er normalt passive, og avventer oppkoblinger fra klienter. Dvs. at tjenere må være mer eller mindre kontinuerlig tilgjengelig, og en part som opererer tjenere normalt vil ha krav på seg til oppetid og tilgjengelighet. En tjener vil ofte selv opptre som klient overfor andre tjenere.
TLS	Transport Layer Security Protokoll for å realisere transportlagssikkerhet i Internett (jfr. SSL)
Transport-endepunkt	Endepunkt (fra/til) for datatrafikk på transportlaget. Hvert program/prosess på en maskin som kommuniserer har vanligvis sine egne distinkte transport-endepunkter, og standard tjenester/protokoller har vanligvis transport-endepunkter med faste (velkjente) adresser. (Se også port.)
Transportagent	(MTA, Message Transport Agent) Tjener/agent som normalt opererer på vegne av tjenesteleverandør eller organisasjon og mottar og videresender meldinger.
Transportlag	Det protokoll-laget i et datanett som tilbyr direkte kommunikasjon bruker til bruker (prosess til prosess), og vanligvis med implisitt feildeteksjon og evt. -korrigering.
Transportnettet	I denne veiledningen har vi brukt dette som en samlebetegnelse for det fysiske datanettet og alt av programvare og protokoller under applikasjonslaget.
TSAP	Transport Service Access Point Transport-endepunkt, se også port. (TSAP er primært OSI-terminologi.)
TTP	Trusted Third Party Tiltrodd tredjepart (s.d.)
Tunnellering	Teknikk hvor trafikk transporteres i transitt gjennom et annet nett mellom en avsender og mottaker som bruker de samme protokollene. (Innebærer vanligvis at den opprinnelige protokollen eller dataformatet kapsles inn på en eller annen måte uten noen egentlig protokoll-konvertering, slik at det bare oppfattes som innhold og ikke er direkte tilgjengelig for protokoller i transitt-nettet. Eksempler: IP-tunneller i VPN. HARPOON-innkapslede MIME-meldinger i et X.400-nett.)
UA	User Agent Brukeragent.
UCS	Universal Multiple-Octet Coded Character Set. Det kodede tegnsettet som defineres i ISO 10646 (s.d.)
UCS-2	Delmengden av UCS (s.d.) med kodeverdier under 2^{16}, representert ved 2 oktetter (16 biter) pr. tegn.

UCS-4	UCS (s.d.), representert ved 4 oktetter (32 biter) pr. tegn.
UDP	User Datagram Protocol Protokollen som tilbyr en forbindelsesløs (datagramorientert) transporttjeneste i Internett.
UNICODE	Standard (definert av et industrikonsortium) som definerer et kodet tegnsett med 16-biters koder. Kodeverdiene er sammenfallende med ISO 10646 (den delmengden av tegnene derfra som har kodeverdier under 2^{16}), ISO 8859-1 (8-biters koder) og ASCII (7-biters koder).
UTF-8	UCS Transformation Format Tegnkodingsmekanisme som representerer tegnene i UCS ved et variabelt antall oktetter (1-6) pr. tegn, slik at de mest vanlige (ASCII-tegnene) representeres med en oktett. UTF-8 ser ut til å bli den mest aktuelle tegnrepresentasjonen i Internett på sikt.
UTF-16	UCS Transformation Format Tegnkodingsmekanisme som representerer tegnene i en delmengde av UCS ved en eller to par av oktetter. Tegnene i BMP representeres direkte.
Utveksling	I EDIFACT: En samling av (EDIFACT-)meldinger eller grupper som transporteres gjennom meldingsutvekslingssystemet som én enhet (melding i betydning 1).
Velkjent	En velkjent adresse er en adresse som det normalt ikke er nødvendig å slå opp i en katalogtjeneste eller lignende, fordi den er fast tildelt. Standard tjenester som SMTP, FTP, HTTP osv. i Internettet finnes vanligvis på velkjente portnumre.
Virtuelt privat nett	1) Lukket nett realisert ved kryptografiske teknikker (typisk på nettverkslaget) på basis av et mer åpent nett 2) Teknologier/protokoller for å realisere 1)
VPN	Virtual Private Network Virtuelt privat nett
Web	1) Tjenesten som ser innholdet i tjenester tilgjengelig gjennom Internettet som en samling av gjensidig hyperlenkede dokumenter. 2) Mer spesifikt: Tjenester basert på bæretjenesten HTTP.
X.25	Standard for pakkesvitsjet datanett (mye brukt som bæretjeneste for X.400).
X.21	Standard for linjesvitsjet datanett.
X.400	En samling av standarder som definerer meldingsformater og protokoller for meldingsutveksling. (Omfatter X.420 og X.435, se nedenfor. X.400 er egentlig identifikasjonen på den første, innledende standarden i serien, men brukes også som en betegnelse på hele serien.)
X.420	Standard som definerer meldingsformatet IPM (s.d.)

X.435	Standard som definerer meldingsformatet EDIM (s.d.)
X.500	En samling av standarder som definerer en katalogtjeneste og sikkerhetsmekanismer relatert til den. (Omfatter bl.a. X.509. Som for X.400 er X.500 egentlig identifikasjonen på den første, innledende standarden i serien, men brukes også som betegnelse på hele serien.)
X.509	Standard for sertifikatformat. (Aktuell versjon er X.509 versjon 3)
Åpent nett	Datanett som er alment tilgjengelig, ved at det ikke er noen (eller bare svake) restriksjoner på hvilke parter som kan knytte seg til nettet.

B Litteraturliste

B.1.1 ISO-standarder og profiler

Katalogen over ISO-standarder kan nås gjennom http://www.iso.ch

- ISO: ”ISO 9735-1:1998: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 1: Syntax rules common to all parts, together with syntax service directories for each of the parts”, 1998.

- ISO: ”ISO 9735-2:1998: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 2: Syntax rules specific to batch EDI”, 1998.

- ISO: ”ISO 9735-3:1998: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 3: Syntax rules specific to interactive EDI”, 1998.

- ISO: ”ISO 9735-4:1998: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 4: Syntax and service report message for batch EDI (message type – CONTRL)”, 1998.

- ISO: ”ISO 9735-5:1999: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin.)”, 1999.

- ISO: ”ISO 9735-6:1999: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules(Syntax version number: 4) — Part 6: Secure authentication and acknowledgement message (message type – AUTACK)”, 1999.

- ISO: ”ISO 9735-7:1999: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 7: Security rules for batch EDI (confidentiality)”, 1999.

- ISO: ”ISO 9735-8:1998: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 8: Associated data in EDI”, 1998.

- ISO: ”ISO 9735-9:1999: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) — Application Level Syntax Rules (Syntax version number: 4) — Part 9: Security key and certificate management message (message type – KEYMAN)”, 1999.

- ISO: ”ISO/IEC 10021-1:199X/PDAM2, Information Technology – MHS- Part 1: System and Service Overview – Amendment 2: Additional Security Features”, juni 1997.

- ISO: ”ISO/IEC 10021-5:199X/DAM5, Information Technology – MHS- Part 5: Message Store: Abstract Service Definition – Amendment 5: Additional Security Features”, juni 1997.

- ISO: ”ISO/IEC 10021-7:199X/PDAM2, Information Technology – MHS- Part 7: Interpersonal Messaging System – Amendment 1: Additional Security Features”, juni 1997.

B.1.2 ITU-rekommendasjoner

Katalogen over ITU-rekommentasjoner kan nås gjennom http://www.itu.int/

- ITU: ”Recommendation X.400/F.400: Message Handling System and Service Overview”, juli 1996.
- ITU: ”Recommendation X.400/F400: Amendment 1”, september 1998.
- ITU: ”Recommendation X.402: Message Handling Systems (MHS): Overall Architecture”, november 1995.
- ITU: ”Recommendation X.402: Amendment 1”, desember 1997.
- ITU: ”Recommendation X.402: Corrigendum 1”, august 1997.
- ITU: ”Recommendation X.411: Message Handling Systems (MHS): Message Transfer System: Abstract Service Definition and Procedures”, november 1995.
- ITU: ”Recommendation X.411: Amendment 1”, desember 1997.
- ITU: ”Recommendation X.411: Corrigendum 1”, august 1997.
- ITU: ”Recommendation X.411: Corrigendum 2”, desember 1997.
- ITU: ”Recommendation X.411: Corrigendum 3”, september 1998.
- ITU: ”Recommendation X.413: Message Handling Systems (MHS): Message Store: Abstract-service Definition”.
- ITU: ”Recommendation X.413: Amendment 1”, august 1997.
- ITU: ”Recommendation X.413: Amendment 2”, desember 1997.
- ITU: ”Recommendation X.413: Corrigendum 1”, august 1997.
- ITU: ”Recommendation X.413: Corrigendum 2”, desember 1997.
- ITU: ”Recommendation X.413: Corrigendum 3”, september 1998.
- ITU: ”Recommendation X.419: Message Handling Systems (MHS): Protocol specifications”.
- ITU: ”Recommendation X.419: Amendment 1”, desember 1997.
- ITU: ”Recommendation X.419: Amendment 2”, august 1998.
- ITU: ”Recommendation X.419: Corrigendum 1”, august 1997.
- ITU: ”Recommendation X.420: Message Handling Systems (MHS): Interpersonal Messaging System”, 1998.
- ITU: ”Recommendation X.435: Message Handling Systems (MHS): Electronic Data Interchange Messaging System”, juni 1999.
- ITU: ”Recommendation X.500: The Directory — Overview of Concepts, Models and Services”, august 1997.

- ITU: ”Recommendation X.509: The Directory — Authentication Framework”, august 1997.

- ITU: ”Recommendation X.520: The Directory — Selected Attribute Types”, august 1997.

B.1.3 Internet Request For Comments (RFC) og Internet Drafts

RFC-er og andre Internett-dokumenter kan hentes fra flere kilder, f.eks. ftp://ftp.ntnu.no/pub/rfc/ med filnavn rfcXXXX.txt, stdXXXX.txt etc. Internet drafts kan hentes fra ftp://ftp.nordu.net/internet-drafts/, se 1id-abstracts.txt for hvilke versjoner av drafts som er gjeldende. Internet Mail Consortium (http://www.imc.org) har en utmerket oversikt over aktuelle dokumenter angående elektronisk post.

- D. Moberg, D. Brooks, R. Drummond,: "HTTP Transport for Secure EDI", Work-in-progress Internet Draft, oktober 1999. ftp://ftp.nordu.net/internet-drafts/draft-ietf-ediint-as2-06.txt

- T. Harding, R. Drummond, C. Shih: ”MIME-based Secure EDI”, Work-in-Progress Internet draft, oktober 1999. ftp://ftp.nordu.net/internet-drafts/draft-ietf-ediint-as1-11.txt

- T. Harding, R. Drummond, C. Shih: ”Requirements for Inter-operable Internet EDI”, Work-in-Progress Internet draft, oktober 1999. ftp://ftp.nordu.net/internet-drafts/draft-ietf-ediint-req-08.txt

- J. Postel: ”RFC 821: Simple Mail Transfer Protocol”, august 1982.

- D. Crocker: ”RFC 822: Standard for the Format of ARPA Internet Text Messages”, august 1982.

- J. Linn: ”RFC 1421: Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures”, februar 1993.

- S. Kent: ”RFC 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management”, februar 1993.

- D. Balenson: ”RFC 1423: Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes and Identifiers”, februar 1993.

- B. Kalinski: ”RFC 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services”, februar 1993.

- H. Alvestrand, J.Romaguera, K. Jordan: ”RFC1496: Rules for Downgrading Messages from X.400/88 to X.400/84 When MIME Content-Types are Present in the Messages”, august 1993.

- D. Crocker: ”RFC 1767: MIME Encapsulation of EDI Objects”, mars 1995.

- J. Galvin, S. Murphy, S. Crocker, N. Freed: ”RFC 1847: Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted”, oktober 1995.

- E. Nebel, L. Masinter: ”RFC 1867: Form-based File Upload in HTML”, november 1995.

- J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker: ”RFC 1869: SMTP Service Extensions”, november 1995.

- N. Berge: ”RFC 1875: UNINETT PCA Policy Statements”, desember 1995.

- K. Moore: ”RFC 1891: SMTP Service Extension for Delivery Status Notifications”, januar 1996.

- G. Vaudreuil: "RFC 1892: The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", januar 1996.

- G. Vaudreuil: "RFC 1893: Enhanced Mail System Status Codes", januar 1996.

- K. Moore, G. Vaudreuil: "RFC 1894: An Extensible Message Format for Delivery Status Notifications", januar 1996.

- J. Myers, M. Rose: ”RFC 1939: Post Office Protocol — Version 3”, mai 1996.

- D. Atkins, W. Stallings, P. Zimmermann: ”RFC 1991: PGP Message Exchange Formats”, august 1996.

- M. Elkins: ”RFC 2015: MIME Security with Pretty Good Privacy (PGP)”, oktober 1996.

- F. Yergeau: ”RFC 2044: UTF-8, a transformation format of ISO 10646”, januar 1998.

- N. Freed, N. Borenstein: "RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", november 1996.

- N. Freed, N. Borenstein: "RFC 2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", november 1996 •

- N. Freed, J. Klensin, J. Postel: ”RFC 2048: Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures”, november 1996.

- N. Freed, N. Borenstein: "RFC 2049: Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", november 1996.

- M. Crispin: ”RFC 2060: Internet Message Access Protocol — Version 4rev1”, desember 1996.

- R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee: ”RFC 2068: Hypertext Transfer Protocol — HTTP/1.1”, januar 1997. Se også: http://w3c.org/Protocols/Specs.html

- S. Kille: ”RFC 2156: MIXER (MIME Internet X.400 Enhanced Relay) Mapping between X.400 and RFC-822/MIME”, januar 1998.

- H. Alvestrand: ”RFC 2157: Mapping between X.400 and RFC-822/MIME Message Bodies”, januar 1998.

- T. Dierks, C. Allen: ”RFC 2246: The TLS Protocol Version 1.0”, januar 1998.

- R. Fajman: "RFC 2298: Extensible Message Format for Message Disposition Notifications”, mars 1998.

- S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, L. Repka: ”RFC 2311: S/MIME Version 2 Message Specification”, mars 1998.

- S. Kent, R. Atkinson, "RFC 2401: Security Architecture for the Internet Protocol", november 1998.

- S. Kent, R. Atkinson, "RFC 2402: IP Authentication Header", november 1998.

- S. Kent, R. Atkinson, "RFC 2406: IP Encapsulating Security Payload (ESP)", RFC 2406, november 1998.

- D. Maughan, M. Schertler, M. Schneider, J. Turner: ”RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)”, november 1998.

- R. Thayer: ”RFC 2411: IP Security Roadmap”, november 1998.

- R. Housley, W. Ford, W. Polk, D. Solo: ”RFC 2459: Internet X.509 Public Key Infrastructure. Certificate and CRL Profile”, januar 1999.

- P. Hoffman: ”RFC 2487: SMTP Service Extension for Secure SMTP over TLS”, januar 1999.

- C. Adams, S. Farrell: ”RFC 2510: Internet X.509 Public Key Infrastructure. Certificate Management Protocols”, mars 1999.

- M. Myers, C. Adams, D. Solo, D. Kemp: ”RFC 2511: Internet X.509 Public Key Infrastructure. Certificate Request Format”, mars 1999.

- C. Newman: ”RFC 2595: Using TLS with IMAP, POP3 and ACAP”, juni 1999.

- R. Housley: ”RFC 2630: Cryptographic Message Syntax”, juni 1999.

- E. Rescorla: ”RFC 2631: Diffie-Hellman Key Agreement Method”, juni 1999.

- B. Ramsdell (redaktør): ”RFC 2632: S/MIME Version 3 Certificate Handling”, juni 1999.

- B. Ramsdell (redaktør): ”RFC 2633: S/MIME Version 3 Message Specification”, juni 1999.

- P.Hoffman (redaktør): ”RFC 2634: Enhanced Security Services for S/MIME”, juni 1999.

- A. O. Freier, P. Karlton, P. C. Kocher: ”The SSL Protocol. Version 3.0”, Work-in-progress Internet draft, november 1996. (Denne spesifikasjonen er utgått som Internet draft og fjernet fra offisielle arkiv, men er den siste tilgjengelige og finnes hos Netscape: http://home.netscape.com/eng/ssl3/draft302.txt)

- J. Reynolds, S. Braden (redaktør): ”RFC2500: Internet Official Protocol Standards”, Internet Engineering Task Force, juni 1999.
  Siste gjeldende utgave er tilgjengelig som STD1:
  ftp://ftp.ntnu.no/pub/rfc/std1.txt

B.1.4 Bransjestandarder og andre spesifikasjoner

- Norsk EDIPRO: ”Nasjonal infrastruktur for EDB. Kommunikasjonsbærere for EDI. Interimsløsning for bruk av MHS (X.400) og Vurdering av FTAM”, desember 1990.

- RSA Laboratories ”PKCS #1: RSA Cryptography Standard. Version 2.0” Oktober 1998. http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-1.html

- RSA Laboratories: ”PKCS #7: Cryptographic Message Syntax Standard. Version 1.6” Mai 1997. http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-7.html

RSA Laboratories: ”PKCS #9: Selected Attribute Types. Version 1.1” November 1993. http://www.rsa.com/rsalabs/pubs/PKCS/html/pkcs-9.html