



(12) **PATENT**

(19) NO

(11) **315887**

(13) B1

(51) Int Cl<sup>7</sup>

H 04 L 29/06, 12/56

## Patentstyret

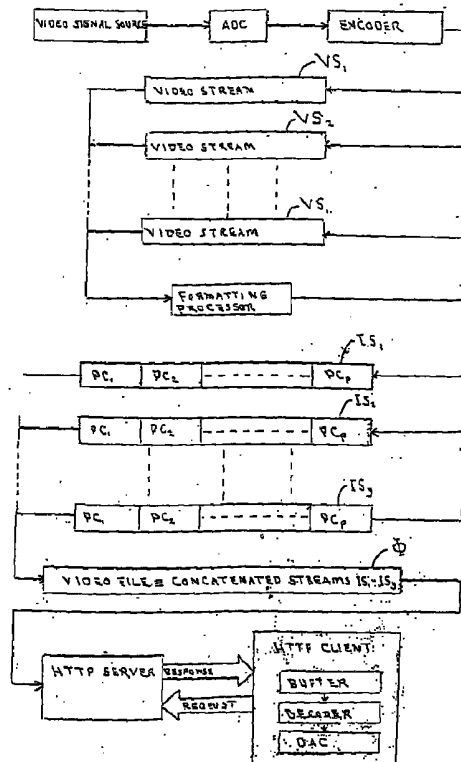
(21) Søknadsnr	20010056	(86) Int. inng. dag og søknadsnummer	
(22) Inng. dag	2001.01.04	(85) Videreføringdag	
(24) Løpedag	2001.01.04	(30) Prioritet	Ingen
(41) Alm. tilg.	2002.07.05		
(45) Meddelt dato	2003.11.03		
(71) Patenthaver	Fast Search & Transfer ASA, Postboks 1677 Vika, 0120 Oslo, NO		
(72) Oppfinner	Harald Dankworth, 0468 Oslo, NO Geirr I Leistad, 1337 Sandvika, NO		
(74) Fullmektig	Geirr I Leistad - Thin Film Electronics ASA, 0124 Oslo		

(54) Benevnelse **Fremgangsmåter ved overføring og søking av videoinformasjon**

(56) Anførte publikasjoner WO 0035162, WO 0119066, US 5898692, US 6128653

(57) Sammendrag

I en fremgangsmåte ved overføring av videoinformasjon mellom HTTP-tjenere og klienter i en delt nettverkressurs, spesielt Internett, er videoinformasjonen lagret som en videofil bestående av pakkedelte videostreamer kompresjonskodet med gjennomsnittlige bitrater  $t[c]$  som dekker klientenes forventede kanalbitrater  $\sigma$ . Hver pakke samt videofilen er forsynt med en topptekst som rommer informasjon til å realisere en båndbreddeskalerbar videooverføring over en egnet versjon av HTTP. Under overføringen svitsjes det mellom videostreamene på basis av en estimering av kanalbitraten  $\sigma$  og informasjonen i pakkeopptekstene, slik at bitraten  $t[c]$  avstemmes til klientens aktuelle kanalbitrate  $\sigma$ . I en fremgangsmåte til klienteksekvert søking og gjenfinning av videoinformasjon i en delt nettverkressurs, spesielt søking av et bilde  $F_x$  i en pakkedelt videostream, deles pakkene i en videostream i grupper spesifisert av informasjon i toppteksten til den første pakke i hver gruppe. På basis av gitte søkekriterier og søkespesifikk informasjon i pakkeopptekstene finnes pakken med bildet  $F_x$ , slik at det fås en pseudoslumpmessig søking og gjenfinning med bruk av en egnet versjon av HTTP som transportprotokoll.



Oppfinnelsen angår en fremgangsmåte ved overføring på anmodning av videoinformasjon i en delt nettverkressurs, hvor den delte nettverkressurs spesielt er Internett, et intranett eller ekstranett, hvor videoinformasjonen er lagret i form av en kodet videofil på HTTP-tjenere i den delte nettverkressurs og aksesseres av klienter via HTTP (Hypertext Transfer Protocol), hvor hver klient har en videodekoder, og hvor fremgangsmåten tar utgangspunkt i initial videoinformasjon i form av digitaliserte videosignaler.

Oppfinnelsen angår også en fremgangsmåte til klienteksekvert søking og gjenfinning av videoinformasjon i en delt nettverkressurs, spesielt søking og gjenfinning av et ønsket bilde i en videostrøm, hvor den delte nettverkressurs spesielt er Internett, et intranett eller ekstranett, hvor videoinformasjonen er lagret i form av en kodet videofil på HTTP-tjenere i den delte nettverkressurs og aksesseres av klienter via HTTP (Hypertext Transfer Protocol), hvor hver klient har en videodekoder, hvor den kodede videofil er sammenkjedet av multiple, kodede videostrømmer som rommer videoinformasjonens videosignaler komprimert ved gjennomsnittlig bitrater  $t[c]$  som dekker klientenes forventede kanalbitrater  $\sigma$ , hvor hver kodet videostrøm er delt i  $p$  pakker med varierende lengde  $q$ , hvor hver pakke omfatter en topptekst og nyttelast, hvor pakkene i en strøm er anordnet i ikke-overlappende, suksessive grupper av to eller flere påfølgende pakker, slik at hver strøm deles i  $m$  slike grupper, hvor toppteksten til den første pakke i hver gruppe i tillegg til informasjon om antall  $n$  videobilder som pakken inneholder og referanser til andre pakker og strømmer, forsynes med informasjon om en hoppdistanse  $d_j$  som svarer til den samlede lengde av pakkene i gruppen og antall  $j$  bilder som hoppdistansen  $d_j$  og den første påfølgende pakke i den påfølgende gruppe omfatter, hvor videofilen dessuten omfatter en topptekst som rommer informasjon om parametrene til strømmene, hvor informasjonen om strømmenes parametre inkluderer distansene  $d_k$  og  $d_l$  fra videofilens begynnelse til henholdsvis begynnelsen av hver strøm og til slutten av den første pakke i hver strøm, og hvor overføringen av videoinformasjon skjer båndbreddeskalert.

I en delt nettverkressurs vil de enkelte ressurser ha en varierende kvalitet og varierende operative parametre, slik at en delt nettverkressurs fremstår som et heterogent kommunikasjonsnettverk uten garantert tjenestekvalitet. Selv om oppfinnelsen generelt angår tjenester i delte nettverkressurser, vil i det følgende omtalen spesifikt være rettet mot Internett som er den mest kjente

og utbredte eksempel på en offentlig tilgjengelig delt nettverkressurs. Som vel kjent er båndbredden til nettverkforbindelsene til Internett meget variable. Typiske forbindelsesbåndbredder kan variere fra 20-500 kbit/s. Da tjenestekvaliteten på Internett ikke kan garanteres, vil båndbredden og pakkeforsinkelsen for en gitt forbindelse kunne fluktuere på grunn av opphopning i nettverket. Dette er et alvorlig hinder for å overføre båndbredeintensive og tidsfølsomme data som videoinformasjon over Internett.

Et videosignal må komprimeres for å redusere den nødvendige båndbredde for overføring over Internett. For overføring på anmodning komprimeres signalet en gang med en gjennomsnittlig målbitrate. Når det benyttes tapsbeheftet kompresjon, fås det forvrengning i det dekomprimerte signal. Kvaliteten på det dekomprimerte signal er proporsjonal med målbitraten. Internetts iboende heterogenitet er et dilemma når målbitraten skal bestemmes. På den ene side bør målbitraten være så høy at klienter med stor målbitbredde mottar et høykvalitetssignal, men da vil ikke klienter med lav båndbredde motta det samme signal i sanntid. På den annen side bør målbitraten være så lav at klienter med liten båndbredde mottar signalet i sanntid, men da vil klienter med stor båndbredde få et lavkvalitetssignal. Løsningen på dette vil være å benytte båndbreddeskalerbar kompresjon. Båndbreddeskalerbar kompresjon betyr at en rekke undermengder med forskjellig gjennomsnittlig målbitrate og tilsvarende kvalitet kan utvinnes fra det kodede signal. Når det komprimerte signal overføres til klienten, er følgelig signalet tilpasset klientenes tilgjengelige kanalbåndbredde.

Det skal nå gis en omtale av teknikken stand. Eksisterende båndbreddeskalerbare videostrømteknikker for videostrømmer på forespørsel krever en dedisert videotjener (se J. Hunter, V. Witana, M. Antoniades, "A Review of Video Streaming over the Internet", DSTC Technical Report TR97-10, august 1997). Klienten kobler seg opp mot videotjeneren og tjeneren utfører båndbreddeskalering i henhold til en eller annen fremgangsmåte. Den mest vanlig metode er å kode en rekke strømmer med forskjellige gjennomsnittlige bitrater på én fil. Tjeneren svitsjer deretter mellom strømmene avhengig av klientenes kanalbitrate. Denne løsningen har to ulemper. Den første er at en dedisert videotjener er nødvendig for å levere båndbreddeskalerbare videostrømmer på forespørsel og den andre er at

brannvegger mellom videotjeneren og klientene må konfigureres spesielt slik at videostrømmene kan gå gjennom dem.

5 Det er for tiden også kjent videostrømteknikker som benytter HTTP som transportprotokoll. Slike arkitekturer er implementert som følger. Klienten anmoder om en videofil. Når HTTP-tjeneren mottar anmodningen, begynner den å overføre videofil i en HTTP-respons til klienten. Ved å benytte HTTP som transportprotokoll er det ikke nødvendig med noen dedisert videotjener, da en HTTP-tjener er tilstrekkelig. Videre vil strømmene, da de overføres med HTTP, normalt ikke blokkeres av brannvegger og det blir derfor mulig med webtitting, noe som øker antallet klienter som er i stand til å motta strømmen. De eksisterende videostrømteknikker basert på HTTP har imidlertid to ulemper (se RealNetworks Inc., "Delivering RealAudio or RealVideo from a Web Server", RealNetworks Technical Blueprint Series, 1998). Den første er at de ikke er båndbreddeskalerbare og den andre er at det ikke er mulig å foreta en slumpmessig søking i videostrømmen.

En egnet versjon av HTTP som versjon 1.1 har to interessante egenskaper, nemlig vedvarende oppkobling og spesifisering av byteområde (se R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1", RFC 2068, UC Irvine, DEC, and MIT/LCS, 20 januar 1997). Vedvarende oppkobling betyr at en rekke HTTP-forespørsler kan overføres over en såkalt socket-forbindelse, dvs. en identifikator for en bestemt tjeneste på en bestemt node på et nettverk. Det er dermed ikke nødvendig å omkoble til tjeneren for hver HTTP-forespørsel.

Byteområdespesifisering åpner for muligheten av å anmode om en 25 undermengde av filen på HTTP-tjeneren. Et koblingsur lukker forbindelsen hvis forespørselen ikke er mottatt av HTTP-tjeneren innenfor et forhåndsbestemt intervall. Det vil dessuten være mulig med et stort antall forespørsler under én oppkobling, f.eks. 100. Disse to egenskaper ved eksempelvis HTTP-versjon 1.1 er grunnlaget for den foreliggende 30 oppfinnelse.

For å overvinne ulempene med den kjente teknikk er en første hensikt med den foreliggende oppfinnelse derfor å realisere båndbreddeskalerbar videooverføring med bruk av HTTP som transportprotokoll, og dernest en annen hensikt å muliggjøre pseudoslumpmessig søking av videoinformasjon med bruk av HTTP som transportprotokoll. 35

Den ovennevnte første hensikt og andre trekk og fordeler oppnås i henhold til oppfinnelsen med en fremgangsmåte som er kjennetegnet ved at den omfatter trinn for å generere ved hjelp av en videokoder  $y$  multiple, kodede videostrømmer som hver rommer den initiale videoinformasjons

5 videosignaler komprimert ved gjennomsnittlige bitrater  $t[c]$  som dekker klientenes forventede kanalbitrater  $\sigma$ , idet videokoderen genererer uavhengig dekodbare videobilder med gitte tidsintervaller; å generere  $y$  kodede mellomstrømmer fra de tilsvarende kodede videostrømmer ved å dele en kodet videostrøm i  $p$  pakker med varierende lengde  $q$ , idet hver pakke

10 omfatter en topp tekst og en nyttelast som rommer de kodede videosignaler for et tidssegment som tilsvarer nyttelasten; å forsyne topp teksten med følgende informasjon: (i) distansene  $d_1$  og  $d_2$  henholdsvis til begynnelsen og slutten av den nærmest påfølgende pakke, (ii) antall  $n$  videobilder som pakken inneholder, samt (iii) en referanse til den tilsvarende pakke i

15 henholdsvis den kodede mellomstrøm med den nærmestliggende lavere gjennomsnittlig bitrate  $t[c_{k-1}]$  og den kodede mellomstrøm med den nærmestliggende høyere gjennomsnittlig bitrate  $t[c_{k+1}]$ , idet  $t[c_k]$  er bitraten for den foreliggende mellomstrøm og  $k, b, a, \in y$ ; å anordne et uavhengig dekodbart videobilde ved begynnelsen av nyttelasten til en pakke; å

20 sammenkjede mellomstrømmene til en slutt fil som lagres på én eller flere HTTP-tjenere; og å forsyne slutt filen med en topp tekst som rommer informasjon om parametrene til strømmene; og dessuten ved følgende trinn iverksatt av klienten: å generere en anmodning om topp teksten og begynnelsen på den første strøm i slutt filen; å estimere kanalbitraten  $\sigma$  og å

25 velge den strøm hvis bitrate  $t[c_k]$  er den i forhold til den estimerte kanalbitrate  $\hat{\sigma}$  nærmestliggende bitrate, som den initiale strøm i overføringen; og deretter å estimere kanalbitraten  $\sigma$  under overføringen og dersom estimatet  $\hat{\sigma}$  er lavere enn den gjennomsnittlig bitrate  $t[c_k]$  til den foreliggende strøm, å svitsje til strømmen med den nærmestliggende lavere

30 gjennomsnittlige bitrate  $t[c_{k-1}]$ , eller dersom estimatet er høyere enn gjennomsnittlige bitrate  $t[c_k]$  til den foreliggende strøm, å svitsje til strømmen med den nærmestliggende høyere gjennomsnittlige bitrate  $t[c_{k+1}]$ , idet svitsjingen av strømmene skjer på basis av pakkereferansene og realiserer en båndbreddeskalierbar videooverføring over HTTP.

I fremgangsmåten ved overføring i henhold til oppfinnelsen er det spesielt foretrukket at den båndbreddeskalerbare videoovervåking skjer over en versjon av HTTP som tillater vedvarende oppkobling og spesifisering av byteområde.

- 5 I utførelsen av fremgangsmåten ved overføring i henhold til oppfinnelsen estimeres enten kanalbitraten  $\sigma$  på basis av en estimator  $\hat{\sigma} = x/\tau$ , hvor  $\hat{\sigma}$  er den estimerte kanalbitrate og  $x$  antall bufrede bit i tidsintervallet  $\tau$ , slik at dersom  $\hat{\sigma} > t[c_k]$  og bufferlengde mindre enn to ganger minimum pakkelenge, svitsjes det til strømmen med en nærmestliggende høyere
- 10 målbitrate  $t[c_a]$ , eller dersom  $\hat{\sigma} < t[c_k]$  og bufferlengde mindre enn minimum pakkelenge, svitsjes det til strømmen med en nærmestliggende lavere målbitrate  $t[c_b]$ ; eller på basis av en estimator  $d\sigma = \frac{d^2x}{dt^2}$ , slik at dersom  $|\dot{\sigma}| \geq 0$ , svitsjes det til strømmen med en nærmestliggende høyere målbitrate  $t[c_a]$ , eller dersom  $|\dot{\sigma}| < 0$ , svitsjes det til strømmen med en
- 15 nærmestliggende lavere målbitrate  $t[c_b]$ . I det sistnevnte tilfelle fastsettes en grenseverdi  $\Delta\dot{\sigma}$ , slik at svitsjing finner sted dersom  $|\dot{\sigma}| \geq \Delta\dot{\sigma}$ .

Alternativt kan i henhold til oppfinnelsen kanalbitraten  $\sigma$  estimeres ved gjentatt å integrere kanalbitraten  $\sigma$  over påfølgende tidsintervaller  $\tau = t_a - t_b$ , idet  $t_a$  og  $t_b$  henholdsvis er øvre og nedre grense for intervallet  $\tau$  og et

- 20 integrasjonsresultat  $\Sigma$  gitt ved  $\Sigma = \int_{t_b}^{t_a} \sigma dt$ , og å sammenligne integrasjonsresultatene  $\Sigma_2, \Sigma_1$  for respektive påfølgende tidsintervall  $\tau_2, \tau_1$ , slik at dersom  $\Sigma_2 - \Sigma_1 > 0$ , svitsjes det til strømmen med en nærmestliggende høyere målbitrate  $t[c_a]$  eller dersom  $\Sigma_2 - \Sigma_1 < 0$ , svitsjes det til strømmen med nærmestliggende lavere målbitrate  $t[c_b]$ , idet det fastsettes en
- 25 grenseverdi  $\Delta\Sigma$ , slik at svitsjing finner sted dersom  $|\Sigma_2 - \Sigma_1| \geq \Delta\Sigma$ .

I en fordelaktig utførelse av fremgangsmåten ved overføring i henhold til oppfinnelsen benyttes en HTTP-tjener tilpasset HTTP-versjon 1.1 og videooverføringen finner da sted over samme versjon av HTTP.

I en annen fordelaktig utførelse av fremgangsmåten ved overføring i henhold til oppfinnelsen anordnes pakkene i strømmene i ikke overlappende suksessive grupper av to eller flere påfølgende pakker, slik hver strøm deles i  $m$  slike grupper, og toppteksten til den første pakke i en gruppe forsynes med informasjon om en hoppdistanse  $d_j$  som svarer til den samlede lengde av pakkene i gruppen og antall  $j$  bilder som hoppdistansen  $d_j$  og den første pakke i den påfølgende gruppe omfatter.

I en tredje fordelaktig utførelse av fremgangsmåten ved overføring i henhold til oppfinnelsen anordnes strømmene slumpmessig i sluttfilen, idet strømmen med lavest bitrate  $t[c_L]$  bare refererer til pakker i strømmen med den nærmestliggende høyere bitrate og strømmen med høyest bitrate  $t[c_H]$  bare refererer til pakker i strømmen med den nærmestliggende lavere bitrate.

I en fjerde fordelaktig utførelse av fremgangsmåten ved overføring i henhold til oppfinnelsen anordnes strømmene suksessivt med stigende bitrate i sluttfilen, slik at strømmen med lavest bitrate  $t[c_L]$  utgjør den første videostrøm i filen og strømmen med høyest bitrate  $t[c_H]$  utgjør den siste videostrøm i filen.

Fortrinnsvis rommer strømmene et uavhengig dekodbart videobilde ved posisjoner svarende til begynnelsen av hver pakke i en foreliggende strøm.

Fortrinnsvis rommer de kodede strømmer det samme antall videobilder.

Hvor strømmene har forskjellig billedrate, er det i henhold til oppfinnelsen fordelaktig å definere et justeringsbilde som en egen bildetype, og å justere bilderatene i hver strøm til samme rate ved å innsette et passende antall justeringsbilder i de respektive strømmer.

Fortrinnsvis er pakkelengden  $q$  høyst lik en konfigurert anmodningspause for HTTP-tjeneren, og fortrinnsvis prosesseres de kodede strømmer i parallell bilde for bilde.

Endelig er det ved fremgangsmåten ved overføring i henhold til oppfinnelsen fordelaktig at toppteksten til sluttfilen inneholder informasjon om antall  $y$  strømmer i filen, den gjennomsnittlige bitrate  $t[c]$  for hver strøm, og distansene  $d_k$  og  $d_l$  fra sluttfilens begynnelse til henholdsvis begynnelsen av hver strøm og til slutten av den første pakke i hver strøm, og dessuten at klienten på basis av videostrømmenes parametre genererer undermengder av

strømmer, idet svitsjing mellom strømmene bare kan finne sted i disse undermengder.

Den ovennevnte annen hensikt oppnås med en fremgangsmåte til klienteksekvert søking og gjenfinning av videoinformasjon i en delt nettkressurs, idet denne fremgangsmåte i henhold til oppfinnelsen er kjennetegnet ved å generere en anmodning til HTTP-tjeneren om nedlasting av toppteksten til den første pakke i en gruppe i en videostrøm, å sammenligne nummeret  $x$  til det ønskede bilde med  $j$ , og dersom  $x \in j$ , å gjenoppta overføringen og dekodingen av videostrømmen fra og med første bilde i pakken hvor bildet befinner seg, idet denne pakke er en av pakkene i gruppen samt den første pakke i den påfølgende gruppe, eller dersom  $x \notin j$ , å anmode om toppteksten til den første pakke i den påfølgende gruppe, og eventuelt å fortsette prosessen inntil det ønskede bilde er funnet, hvorefter nedlasting og dekoding av strømmen gjenopptas fra og med første bilde i pakken hvor det ønskede bilde befinner seg, slik at det realiseres en bildebasert og pakkeformatert søking og gjenfinning av videoinformasjon med bruk av HTTP som transportprotokoll.

Ved fremgangsmåten til klienteksekvert søking og gjenfinning av videoinformasjon i henhold til oppfinnelsen er det spesielt foretrukket at det benyttes en versjon av HTTP som tillater vedvarende oppkobling og spesifisering av byteområde.

Ved framgangsmåten til klienteksekvert søking og gjenfinning av videoinformasjon benyttes fordelaktig en HTTP-tjener tilpasset HTTP-versjon 1.1 og HTTP-versjon 1.1 benyttes som transportprotokoll.

Både i fremgangsmåten ved overføring og i fremgangsmåten til klienteksekvert søking og gjenfinning av informasjon er det fordelaktig at det genereres og vedlikeholdes en stakkliste for HTTP-anmodninger, idet en anmodning legges i stakklisten ved sending og fjernes derfra etter prosessering av den fra HTTP-tjeneren mottatte respons, og at anmodningene i stakklisten sendes på ny ved gjenopprettet forbindelse etter et eventuelt brudd i forbindelsen mellom HTTP-tjener og klient under mottak av responsen. I den forbindelse vil fortrinnsvis den første anmodning oppdateres dersom et brudd i forbindelsen finner sted under mottak av en pakke, idet begynnelsen av pakken justeres for de eventuelt allerede mottatte data.



Oppfinnelsen skal nå forklares mer detaljert ved hjelp av utførelseseksempler og med henvisning til den vedføyde tegning hvor

- fig. 1 viser en skjematisk oversikt over genereringen av videofilen samt det ved den foreliggende oppfinnelsen benyttede tjener-klient-system,
- 5 fig. 2a strukturen til en videofil som benyttet ved den foreliggende oppfinnelse,
- fig. 2b toppteksten til videofilen på fig. 2a,
- fig. 3a pakke­delingen av en videostrøm som benyttet i videofilen på fig. 2a,
- fig. 3b sammensetningen av en pakke i videostrømmen på fig. 3a,
- 10 fig. 3c toppteksten til pakken på fig. 3b,
- fig. 4 skjematisk prinsippet for å oppnå båndbreddeskalering ved svitsjing mellom videostrømmer,
- fig. 5a hvordan en videostrøm deles i grupper av pakker,
- fig. 5b strukturen til den enkelte gruppe i videostrømmen på fig. 5a,
- 15 fig. 5c sammensetningen av pakkene i videostrømmen på fig. 5b,
- fig. 5d toppteksten til den første pakke i en gruppe i videostrømmen på fig. 5a,
- fig. 6a et bufferminne anordnet foran klientens de­koder, og
- fig. 6b flytdiagrammet for en foretrukket utførelse av den
- 20 pseudoslumpmessige søking av videoinformasjon overført ved fremgangsmåten for overføring i henhold til oppfinnelsen.
- I den etterfølgende detaljerte beskrivelse av en utførelse av den foreliggende oppfinnelse er det underforstått at hvor annet ikke uttrykkelig er sagt, er den delte nettverksressurs representert ved Internett og HTTP-versjon 1.1
- 25 benyttes som transportprotokoll. Spesielt oppnår nemlig den foreliggende oppfinnelse båndbreddeskalierbar videooverføring med bruk av HTTP-versjon 1.1 som transportprotokoll ved at det genereres en bitstrøm som er optimert for dette formål. Det er spesielt hensiktsmessig å benytte HTTP-versjon 1.1 som transportprotokoll, og dette innebærer at det må benyttes en koder og en
- 30 HTTP-klient som gjør båndbreddeskalierbar videooverføring med bruk av

HTTP 1.1 mulig med et minimum av administrasjon. Dessuten vil HTTP-klienten også være i stand til å utføre rask søking i den komprimerte videoinformasjon utenfor det område som allerede er mottatt.

5 Den foreliggende oppfinnelse vil også sette videoleverandører i stand til å tilby båndbreddeskalerbar video fra en tjener som er tilpasset HTTP-versjon 1.1. Koding av en fil er tilstrekkelig til å dekke båndbreddeområdet til klientene og søkemuligheten som nå tilbys, vil muliggjøre søkbare videoarkiver hvor søkingen utføres på grupper av pakker i en videostrøm, eventuelt med bruk av merkelapper som refererer til forskjellige posisjoner i 10 videofilen, idet disse merkelapper kan være anordnet i toppteksten til den komprimerte videofil. Det vil heller ikke lenger være nødvendig med en dedisert videotjener.

Ved den foreliggende oppfinnelse konstrueres et videostrømfilformat som tillater den å utnytte den vedvarende oppkobling som tilbys av versjoner av 15 HTTP. Særlig gjelder det HTTP-NG-versjoner som versjon 1.1. Spesielt for disse versjoner gjelder at de ikke kobles ned etter hver anmodning eller forespørsel fra klienten, men opprettholder forbindelsen over et spesifikt antall anmodninger. Samtidig er det også mulig for klienten å spesifisere et bestemt byteområde som skal overføres.

20 Fig. 1 viser skjematisk generering av en videofil som tillater båndbreddeskalerbar overføring, og systemoppbygget for overføringen med bruk av HTTP-versjon 1.1 som transportprotokoll. En videosignalkilde genererer videosignaler som leveres til en analog/digitalomformer ADC som digitaliserer videosignalene og gir dem til en koder som genererer  $y$  multiple, 25 kodete videostrømmer  $VS_1$ - $VS_y$ . Koderen kompresjonskoder videosignalene med respektive gjennomsnittlige bitrater  $t[c]$ , slik at disse dekker de forventede kanalbitrater  $\sigma$  for overføring til klientene. Eksempelvis kan det genereres seks videostrømmer  $VS_1$ - $VS_6$  som rommer den samme videoinformasjon, men som henholdsvis er kodet med gjennomsnittlige 30 bitrater på 22, 45, 60, 120, 256 og 512 Kb/s. Dermed blir det mulig med en båndbreddeskalerbar videooverføring i området 20-500 Kb/s. Det er nødvendig at videokoderen genererer uavhengig dekodbare videobilder IF ved gitte tidsintervaller. Slike videobilder IF er kjent som intravideobilder for hybride videokodeker. De kodete videostrømmer VS formatteres nå til  $y$  35 kodete mellomstrømmer IS, eksempelvis ved hjelp av en

formateringsprosessor, idet en kodet videostrøm VS deles i  $p$  pakker PC. Disse pakkene PC kan ha varierende lengde  $q$  og omfatter hver en toppstekst  $H_{PC}$  og en nyttelast PL som rommer de kodete videosignaler for et tidssegment som tilsvarer nyttelasten. De nå kodete og formattede

5 videostrømmer IS blir deretter sammenkjedet til en videofil  $\Phi$  for lagring på en HTTP-tjener, spesielt en tjener for HTTP-versjon 1.1 og kan på anmodning fra en HTTP-klient lastes ned til denne. Som vist på fig. 1, bufres den nedlastede videofil på et bufferminne av passende størrelse i HTTP-klienten og dekodes deretter ved hjelp av klientens dekode, idet

10 dekodingen naturligvis finner sted med den gjennomsnittlige bitrate  $t[c]$  som svarer til kodingsbitraten for den aktuelle strøm. Etter dekodingen finner det sted en digital/analog-omforming i klientens digital/analog-omformer DAC, hvorefter den mottatte videoinformasjon kan lagres på et analogt medium eller reproduseres på en egnet avspillingsinnretning. Dette er trivielt og

15 derfor ikke nærmere vist på fig. 1. – Det skal bemerkes at det ikke er nødvendig med et særskilt, ekstra bufferminne for dekode, da det i fremgangsmåten til overføring løpende finner sted en avstemning mellom bitraten  $t[c]$  og kanalbitraten  $\sigma$ . Derimot vil det i klienten være anordnet et ikke vist bufferminne for en eventuell avspilling etter dekoding, noe som vil

20 være innlysende for fagfolk.

Fig. 2a viser den sammenkjedede videofil  $\Phi$  slik den ligger lagret på HTTP-tjeneren. Filen  $\Phi$  er som nevnt sammenkjedet av  $y$  strømmer  $IS_1$ - $IS_y$  og omfatter i tillegg en toppstekst  $H_\Phi$  som er formatert i  $4y+1$  blokker. Den første blokken  $Y$  angir antall strømmer  $y$  i filen  $\Phi$ . Deretter følger  $y$  blokker

25  $T_1$ - $T_y$  som henholdsvis angir den gjennomsnittlige kodingsbitrate  $t[c]$  for de respektive  $y$  strømmer  $IS$ . Nå følger det i toppsteksten to blokker  $D_k$ ,  $D_l$  for hver strøm  $IS$ . Disse blokkene inneholder informasjon om distansene  $d_k$  og  $d_l$  henholdsvis fra begynnelsen av filen  $\Phi$  til begynnelsen av hver strøm  $IS$  og til slutten av den første pakke  $PC_1$  i hver strøm  $IS$ . I tillegg er det for hver

30 strøm  $IS$  én blokk  $\Pi$  som inneholder parametre for strømmen  $IS$ , idet blokken  $\Pi$  eventuelt kan være segmentert i flere underblokker avhengig av antall innlagte parametre. Disse parametre kan f.eks. angå billedimensjon og parametre for en audiokoding. Dette forhindrer at det ved

35 båndbreddeskaleringen svitsjes mellom strømmer med forskjellige billedimensjon, og dersom ikke-båndbreddeskalbar audioinformasjon er

interfoliert sammen med videoinformasjon, forhindres svitsjing mellom strømmer som rommer forskjellige audiokoding. Spesifikt angir på fig. 2b blokken  $D_{k,1}$  distansen til begynnelsen av strømmen  $IS_1$  og blokken  $D_{1,1}$  distansen til slutten av den første pakke  $PC_1$  i strømmen  $IS_1$ , mens  $\Pi_1$  5 rommer parameterinformasjon om strømmen  $IS_1$ . Tilsvarende angir blokken  $D_{k,y}$  distansen til begynnelsen av den siste strøm  $IS_y$  i filen  $\Phi$ , blokken  $D_{1,y}$  distansen til slutten av den første pakke  $PC_1$  i strømmen  $IS_y$  og blokken  $\Pi_y$  da naturligvis parameterinformasjon om strømmen  $IS_y$ .

Fig. 3 viser hvordan hver strøm  $IS$  er delt i  $p$  pakker  $PC_1$ - $PC_p$ . Hver pakke  $PC$  10 har lengden  $q$  og fortrinnsvis er pakkelengden høyst lik en konfigurert anmodningspause på HTTP-tjeneren. Dessuten bestemmer pakkelengden  $q$  balansen mellom pakkeadministrasjonen, oppløsningen ved en eventuell søking av videoinformasjon og tidsintervallet, slik det skal bli omtalt senere, og tidsintervallet mellom en eventuell svitsjing av strømmene  $IS$ . Dersom 15 disse betingelser er oppfylt, kan pakkelengden  $q$  ellers godt variere.

Hver pakke  $PC$  rommer en topptekst  $H_{PC}$ , og en nyttelast  $PL$  som omfatter de kodede videosignaler for et tidssegment som tilsvarer nyttelasten. Nyttelasten  $PL$  til hver pakke  $PC$  rommer et antall  $n$  videobilder  $F$  som kan være forskjellig fra pakke til pakke. Videre er det nødvendig at nyttelasten  $PL$  20 starter med et uavhengig dekodbart videobilde, slik at dekoderen i klienten kan dekode hver pakke uavhengig. Toppteksten  $H_{PC}$  til en pakke  $PC$  er i utgangspunktet formatert i fem blokker, hvor en første blokk  $D_1$  rommer informasjon om distansen  $d_1$  til begynnelsen av den påfølgende pakke, en annen blokk  $D_2$  informasjon om distansen  $d_2$  til slutten av den påfølgende 25 pakke, en tredje blokk  $N$  informasjon om antall videobilder  $n$  i pakken og en fjerde blokk  $B$  en referanse til den tilsvarende pakke i den kodede mellomstrøm  $IS_b$  med en nærmestliggende lavere gjennomsnittlig bitrate  $t[c_{k-1}]$  og en femte blokk  $A$  en referanse til den tilsvarende pakke i den kodede mellomstrøm  $IS_a$  med en nærmestliggende høyere gjennomsnittlig 30 bitrate  $t[c_{k+1}]$ , idet  $t[c_k]$  er bitraten for den foreliggende strøm  $IS_k$ ,  $k, b, a \in y$ .

I tillegg kan toppteksten  $H_{PC}$  til en pakke  $PC$  dessuten romme informasjon som benyttes dersom det implementeres en fremgangsmåte til søking i videoinformasjon, slik dette skal omtales nærmere nedenfor.

Det skal forstås at strømmene IS kan være anordnet slumpmessig i filen  $\Phi$ . Strømmen  $IS_L$  som har den laveste bitrate  $t[c_L]$ , vil bare referere til pakker i strømmen med den nærmestliggende høyere bitrate og tilsvarende vil strømmen  $IS_H$  med en høyeste bitrate  $t[c_H]$  bare referere til pakker i strømmen med den nærmestliggende lavere bitrate.

Det er imidlertid foretrukket og hensiktsmessig at strømmene IS anordnes suksessivt med stigende bitrate  $t[c]$  i filen  $\Phi$  slik at strømmen  $IS_L$  med lavest bitrate  $t[c_L]$  blir den første videostrøm  $IS_1$  i filen  $\Phi$  og strømmen  $IS_H$  med høyest bitrate  $t[c_H]$  den siste videostrøm  $IS_y$  i filen  $\Phi$ . Fortrinnsvis rommer de kodete strømmer VS;IS samme antall videobilder F. Dette forhindrer ikke at det kan benyttes en varierbar videobilderate, da det vil være mulig å definere et justeringsbilde  $F_S$  som en egen bildetype og justere bilderatene i hver strøm IS til samme rate ved at det innsettes et passende antall justeringsbilder  $F_S$  i de respektive strømmer IS. Klienten vil da tolke et justeringsbilde  $F_S$  som en gjentakelse av det forutgående, dekodete videobilde. Strømmen  $IS_B$  som har den nærmestliggende lavere bitrate  $t[c_{k-1}]$  i forhold til bitraten  $t[c_k]$  for en foreliggende strøm  $IS_k$ , skal ha et uavhengig dekodbart videobilde IF ved posisjoner svarende til begynnelsen av hver pakke i den foreliggende strøm  $IS_k$ . Tilsvarende skal strømmen  $IS_a$  som har den nærmestliggende høyere bitrate  $t[c_{k+1}]$  i forhold til bitraten  $t[c_k]$  for den foreliggende strøm  $IS_k$ , også ha et uavhengig dekodbart videobilde IF ved posisjoner svarende til begynnelsen av hver pakke i den foreliggende strøm  $IS_k$ . Endelig vil hver strøm IS være avhengig av strømmene med henholdsvis den nærmestliggende lavere bitrate og den nærmestliggende høyere bitrate og dette gjør at de kodete strømmer må prosesseres i parallell, bilde for bilde.

Videofilen  $\Phi$  med de kodete strømmer IS er plassert på en HTTP-tjener som er tilpasset en egnet versjon av HTTP, spesifikt HTTP-versjon HTTP 1.1. Generelt starter overføringen med at klienten først anmoder HTTP-tjeneren om et fast antall byte. Dette faste antall byte skal inkludere toppteksten  $H_\Phi$  til videofilen  $\Phi$  og starten på den første strøm  $VS_1$  i videofilen  $\Phi$ . Under visse omstendigheter vil det imidlertid ikke være mulig å svitsje mellom samtlige videostrømmer IS i filen  $\Phi$ . Dersom eksempelvis videostrømmene har forskjellige rammedimensjon eller audioinformasjonen som er interfoliert med videoinformasjonen ikke er båndbreddeskalerbar og kodet med forskjellige verdier for hver videostrøm, vil det ikke være mulig å svitsje

mellom slike videostrømmer, da de ikke er båndbreddeskalerbare. For å forhindre svitsjing mellom videostrømmer som ikke er båndbreddeskalerbare, må klienten tolke parameterne i toppteksten  $H_0$  og avgjøre hvilke videostrømmer det kan svitsjes mellom. Det er imidlertid i noen tilfeller bare  
 5 mulig å svitsje mellom en undermengde av det totale antall  $y$  videostrømmer, og klienten vil da, som ovenfor nevnt, generere undermengder av strømmene IS, slik at svitsjing mellom strømmene IS bare kan finne sted i disse undermengder.

Det er nødvendig at klienten estimerer kanalbitraten  $\sigma$  under overføringen.

10 Hvis estimatet  $\hat{\sigma}$  av kanalbitraten  $\sigma$  er lavere enn den gjennomsnittlige bitrate  $t[c_k]$  for den foreliggende strøm  $IS_k$ , svitsjes det til strømmen  $IS_b$  med den nærmestliggende lavere gjennomsnittlig bitrate  $t[c_{k-1}]$ . Hvis estimatet  $\hat{\sigma}$  av kanalbitraten  $\sigma$  er høyere enn den gjennomsnittlige bitrate  $t[c_k]$  for den foreliggende strøm  $IS_k$ , svitsjes det til strømmen  $IS_a$  med den  
 15 nærmestliggende høyere gjennomsnittlige bitrate  $t[c_{k+1}]$ . Dersom strømmene IS er anordnet suksessivt med stigende bitrate i filen  $\Phi$ , vil det naturligvis være den foreliggende strøms  $IS_k$  nabostrømmer  $IS_{k-1}$ ,  $IS_{k+1}$  det alt ettersom svitsjes til. Er den foreliggende strøm  $IS_k$  strømmen med lavest bitrate  $t[c_L]$  og anordnet som den første strøm  $IS_1$  i filen  $\Phi$ , kan det naturligvis bare  
 20 svitsjes til den påfølgende strøm  $IS_2$ . Tilsvarende kan det dersom den foreliggende strøm  $IS_k$  er lik strømmen med høyest bitrate  $IS_H$ , bare svitsjes til den nærmest foregående strøm  $IS_{y-1}$ . Ved å svitsje mellom strømmene kan kodebitraten  $t[c]$  tilpasses en foreliggende kanalbitrate  $\sigma$  og det oppnås  
 25 således båndbreddeskalerbar videooverføring over eksempelvis HTTP-versjon 1.1.

Med henvisning til fig. 4 skal det nå gis en mer spesifikk omtale av hvordan båndbreddeskalerbar videooverføring oppnås ved svitsjing mellom videostrømmene IS i videofilen  $\Phi$ . På fig. 4 er det med piler antydnet de  
 distanser som avstandsblokkene  $D_1$ ,  $D_2$  i en topptekst  $H_{PC}$  refererer til.  
 30 Likeledes er den innbyrdes referanse B, A mellom korresponderende pakker i strømmen antydnet ved piler mellom blokkene. En nærmere omtale av hvordan referansene kommer til anvendelse i fremgangsmåten for overføring vil bli gitt i det følgende.

I eksempelet på fig. 4 er det vist seks strømmer  $IS_1$ - $IS_6$  som sammenkjedet og med tillegg av en filtopptekst  $H_\Phi$  utgjør videofilen  $\Phi$ . Hver strøm  $IS$  består som vist på fig. 4 av tre pakker og rommer det samme antall bilder og den samme videoinformasjon. Både antall strømmer og antall pakker i hver strøm er naturligvis et helt skjematisk eksempel, og formatering i strømmer og pakker vil i realiteten naturligvis rette seg etter de omstendigheter som en overføring av den foreliggende videoinformasjon betinger. På fig. 4 kan strømmene henholdsvis være kodet med de bitrater  $t[c]$  som er angitt i det ovenstående eksempel og således dekke et forventet kanalbitrateområde på 20-500 kbit/s. Videre er strømmene  $IS$  i filen  $\Phi$  på fig. 4 vist anordnet med stigende bitrater, slik at den første strøm  $IS_1$  i filen  $\Phi$  har den laveste bitrate  $t[c_L]$  og den siste strøm  $IS_6$  i filen  $\Phi$  den høyeste bitrate  $t[c_H]$ .

Generelt er hver strøm  $IS$  som vist på fig. 4 formatert i  $p$  pakker, idet  $p = 3$  på fig. 4, og hver strøm inneholder altså den samme videoinformasjon. I toppteksten  $H_{PC}$  til hver pakke er det anordnet to blokker  $D_1$ ,  $D_2$  som rommer byteformatert avstandsinformasjon. Blokken  $D_1$  angir distansen  $d_1$  til begynnelsen av den påfølgende pakke og blokken  $D_2$  distansen  $d_2$  til slutten av den påfølgende pakke i strømmen. Straks de to avstandsblokker  $D_1$ ,  $D_2$  er mottatt for en pakke, anmodes om den påfølgende pakke i strømmen. Sekvenseringen av pakkeanmodninger på denne måte eliminerer rundreiseforsinkelsen i nettverket. Anmodningene om pakker opphører når det mottas to distanseblokker  $D_1$ ,  $D_2$  som inneholder tallet 0.

Toppteksten  $H_{PC}$  til pakkene  $PC$  rommer ytterligere to informasjonsblokker som benyttes ved svitsjing mellom strømmer, nemlig informasjonsblokken  $B$  som refererer til den tilsvarende pakke i strømmen med den nærmestliggende lavere bitrate og informasjonsblokken  $A$  som refererer til videostrømmen med den nærmestliggende høyere bitrate. Slik strømmene  $IS$  er anordnet på fig. 4, nemlig med stigende bitrater  $t[c]$ , vil blokkene  $B$  og  $A$  henholdsvis referere til den foreliggende strøms nabostrømmer. Som vist på fig. 4, rommer ikke den første pakke i hver av strømmene  $IS$  blokkene  $B$  og  $A$ . Når den foreliggende kanalbitrate  $\sigma$  er estimert, noe som skjer når det anmodes om toppteksten  $H_\Phi$  i videofilen  $\Phi$ , vil nemlig blokkene  $T$  med bitrateinformasjon angi hvilken strøm som er ønskelig, og avstandsblokkene  $D_k$  og  $D_l$  vil angi avstanden fra begynnelsen av filen  $\Phi$  til henholdsvis begynnelsen av den ønskede videostrøm  $IS$  og til slutten av den første pakke

PC<sub>1</sub> i denne videostrøm. Det er naturligvis ingenting i veien for at blokkene B, A allikevel kunne legges inn i toppsteksten i den første pakke PC<sub>1</sub> i hver strøm IS, men dette er som det vil ses, en overflødig foranstaltning. Da strømmene som vist på fig. 4, er anordnet med stigende bitrate, vil også

5 strømmen IS<sub>1</sub>, dvs. strømmen IS<sub>L</sub> med den laveste bitrate  $t[c_L]$  bare kunne referere til strømmen med den nærmest høyereliggende bitrate og toppstekstene til pakkene i strømmen IS<sub>1</sub> vil følgelig bare romme avstandsblokken A som refererer strømmen med den nærmestliggende høyere

10 strømmen IS<sub>H</sub> med den høyeste bitrate  $t[c_L]$  bare referere til strømmen med den nærmestliggende lavere bitrate, nemlig strømmen IS<sub>5</sub>. Følgelig inneholder ikke toppsteksten til pakken PC<sub>1</sub> til strømmen IS<sub>6</sub> A-blokken, men bare B-blokken som refererer til strømmen med den nærmestliggende lavere bitrate. Det kan nevnes at blokkene B, A bare kan settes inn ved posisjoner

15 hvor både videobildet F i den foreliggende strøm IS<sub>k</sub> og i strømmene IS<sub>k-1</sub>, IS<sub>k+1</sub> (eller IS<sub>B</sub>, IS<sub>A</sub> ved slumpmessig anordnede strømmen) er intrabilder IF. Intrabilder IF er uavhengig kodede videobilder. Dessuten må de respektive pakker ha samme lengde i strømmene IS<sub>k</sub>, IS<sub>k-1</sub>, og IS<sub>k+1</sub>.

For å svitsje fra en foreliggende strøm til strømmene med nærmestliggende

20 høyere eller lavere bitrater  $t[c_{k+1}]$ ,  $t[c_{k-1}]$  alt etter som kanalbitraten  $\sigma$  forandrer seg, anmodes det om to byteformaterte distanseblokker D<sub>1</sub>, D<sub>2</sub> med fast størrelse og som starter ved den angitte avstand i den strøm det ønskes svitsjet til, idet denne strømmen, slik det er eksemplifisert på fig. 4, naturligvis vil være én av nabostrømmene. I det øyeblikket klientene mottar

25 blokkene D<sub>1</sub>, D<sub>2</sub>, brytes anmodningssekvensen, og straks det anmodes om neste pakke, gjenopprettes anmodningssekvensen. Følgelig innfører svitsjing mellom strømmen IS bare én rundreiseforsinkelse, noe som igjen reduserer den effektive kanalbitrate  $\sigma$ . Det ville være mulig å benytte bare en eneste byteformatert distanseblokk i hver pakke, idet denne blokken da angir

30 størrelsen på neste pakke. Dette ville imidlertid føre til to rundreiseforsinkelser under svitsjingen mellom strømmene IS. For å unngå ombufing under nedlasting eller avspilling i klienten, er det essensielt at svitsjingen mellom strømmene skjer raskt. Følgelig benyttes det to distanseblokker D<sub>1</sub>, D<sub>2</sub> i hver pakke, selv om dette øker

35 pakkeadministrasjonen.



- Innløsende vil administrasjonen av båndbreddeskalerbarhet være omvendt proporsjonal med pakkelengden  $q$ , idet store pakker resulterer i mindre pakkeadministrasjon. Imidlertid vil store pakker øke intervallet hvorunder en svitsjing mellom strømmene IS kan utføres. For å unngå ombufring under nedlasting eller avspilling i klienten, er det viktig at dette intervallet ikke er for langt. Hensiktsmessig er det i fremgangsmåten ved overføring i henhold til den foreliggende oppfinnelse valgt å benytte pakker med en varighet på ca. 15 s. Dette gir en typisk pakkeadministrasjon på 0,20 Kbit/s, avhengig av størrelsen på HTTP-responsen og -anmodningen.
- 5
- 10 Under den initiale oppsetting anmoder klienten om et forhåndsgitt antall byte. Responsen vil omfatte toppsteksten  $H_p$  til videofilen  $\Phi$  og den første del av den første strøm  $IS_1$ . Under responsen estimeres kanalbitraten  $\sigma$ . Basert på estimatet  $\hat{\sigma}$  velges den initiale strøm i overføringen. Hvis den initiale strøm ikke er den første strøm  $IS_1$  i videofilen  $\Phi$ , blir de initiale byter som allerede er mottatt fra strømmen  $IS_1$ , lagt vekk. Hvis den initiale strøm er strømmen  $IS_1$ , fortsetter den neste respons fra det sted den initiale anmodning opphørte. Da strømmen  $IS_1$  som vist i eksempelet, utgjør strømmen som ble kodet med den laveste bitrate  $t[c_L]$ , minimeres initial bufring for kanaler med lav bitrate  $\sigma$ .
- 15
- 20 Hva angår estimeringen av kanalbitraten, vil det være innløsende for fagfolk at den kan skje på forskjellige måter.
- Utgangspunktet for en i fremgangsmåten ved overføring foretrukket metode til estimering av kanalbitraten  $\sigma$  er at videokoderen genererer bitstrømmen med variabel bitrate. Hver videostrøm kodes med en bestemt gjennomsnittlig målbitrate  $t[c]$ . Kvantiseringen og bilderaten under videokodingen justeres av en eller annen passende valgt ratekontrollalgoritme som sikrer at målbitraten  $t[c]$  nås og dessuten at en bufring i koderen holdes på en eller annen forhåndsbestemt størrelse under kodingen av hele videosekvensen. På den annen side må klienten bufre en forhåndsbestemt datamengde før dekodningen starter.
- 25
- 30
- I tillegg vil klienten også måtte bufre dekodete data for avspilling etter en digital/analog omforming, men dette er som nevnt irrelevant i det foreliggende tilfelle. Hvis kanalbitraten  $\sigma$  nå er lik målbitraten  $t[c]$  for videostrømmen IS, unngås ombufring hos klienten før dekodningen.

- Kanalbitraten kan være forskjellig for de forskjellige klienter og dessuten fluktuere systematisk. For å unngå ombufring i klientens bufferminne forut for dekodingen, sørger fremgangsmåten ved overføring i henhold til foreliggende oppfinnelse for båndbreddeskalering ved å svitsje mellom strømmer med forskjellig målbitrate  $t[c]$ . Denne svitsjingen blir som nevnt utført på basis av en estimering av kanalbitraten  $\sigma$ . Estimatoren er ganske enkelt  $\hat{\sigma} = x/\tau$ , hvor  $x$  er antallet bufrede bit i tidsintervallet  $\tau$ . Da det bare er mulig å svitsje mellom strømmer ved grensene for pakkene PC, er oppløsningen i  $\tau$  lik pakkelengden  $q$ . Den følgende heuristikk benyttes til å avgjøre når svitsjing mellom strømmene skal foregå:
- 1) hvis  $\hat{\sigma} > t[c_k]$  og bufferlengde mindre enn to ganger minimum pakkelengde, så svitsjes det til strømmen  $IS_a$  med den nærmestliggende høyere målbitrate  $t[c_a]$ , eller
  - 2) hvis  $\hat{\sigma} < t[c_k]$  og bufferlengde mindre enn minimum pakkelengde, så svitsjes det til strømmen  $IS_b$  med nærmestliggende lavere målbitrate  $t[c_b]$ .
- Med minimum pakkelengde skal det forstås den verdi av  $q$  som benyttes når videofilen  $\Phi$  genereres. Den her angitte metode til estimering av kanalbitraten  $\sigma$  er lett å implementere, da den gjør bruk av bufferlengden og pakkelengden.
- I prinsippet kan det også benyttes en momentan estimering som ikke er betinget av bufferlengde og pakkelengde. Eksempelvis vil det være mulig å benytte den tidsderiverte av  $\sigma$  som estimator for kanalbitraten  $\sigma$ . Estimatoren  $\hat{\sigma}$  er med andre ord  $d\sigma = \frac{d^2x}{dt^2}$ . Svitsjing finner da sted under følgende forutsetninger:
- 1) Hvis  $\dot{\sigma} > 0$ , svitsjes det til strømmen  $IS_a$  med nærmestliggende høyere målbitrate  $t[c_a]$ ,
  - 2) hvis  $\dot{\sigma} < 0$ , svitsjes det til strømmen  $IS_b$  med nærmestliggende lavere målbitrate  $t[c_b]$ .
- Fortrinnsvis vil i praksis svitsjingen skje ved at  $\dot{\sigma}$  blir lik eller overskrider en grenseverdi  $\Delta\sigma$ , slik at  $|\dot{\sigma}| \geq \Delta\sigma$ . Denne grenseverdi kan da være forhåndsbestemt på basis av de valgte kodingsbitrater  $t[c]$  og klientens

bufningskapasitet i forbindelse med dekodning. På grunn av ulik avstand mellom de gjennomsnittlige valgte kodingsbitrater  $t[c]$  er det fordelaktig å skalere  $\Delta\sigma$  avhengig av fortegnet for  $\sigma$  og den aktuelle kodingsbitrate  $t[c_k]$ , slik at svitsjing mellom f.eks. 6 strømmene  $IS_1 - IS_6$  finner sted med bruk av

- 5  $2 \cdot 4 + 2 = 10$  forhåndsvalgte verdier for  $\Delta\sigma$ . For hver av strømmene  $IS_L$  og  $IS_H$  vil det bare være tilordnet en eneste respektive grenseverdi da svitsjing bare vil finne sted til strømmen med henholdsvis nærmestliggende høyere eller nærmestliggende lavere kodingsbitrate.

- Alternativt kan estimeringen av kanalbitraten  $\sigma$  også finne sted ved gjentatt å integrere kanalbitraten  $\sigma$  over påfølgende tidsintervaller  $\tau = t_a - t_b$ , hvor  $t_a$  er den øvre grense og  $t_b$  den nedre grense for intervallet  $\tau$ , og å foreta svitsjingen ved å sammenligne integrasjonsresultater  $\Sigma_1, \Sigma_2$  for respektive påfølgende tidsintervall  $\tau_1, \tau_2$ , slik at svitsjing finner sted til en strøm  $IS_a$  med nærmestliggende høyere bitrate  $t[c_a]$  finner sted dersom  $\Sigma_2 - \Sigma_1 > 0$ , og
- 15 tilsvarende til en strøm  $IS_b$  med nærmestliggende lavere bitrate  $t[c_b]$ , dersom  $\Sigma_2 - \Sigma_1 < 0$ . Fortrinnsvis vil svitsjingen også i disse tilfeller finne sted når en grenseverdi  $\Delta\Sigma$  nås eller overskrides, slik at  $|\Sigma_2 - \Sigma_1| \geq \Delta\Sigma$ . Grenseverdien kan være forhåndsbestemt som på samme grunnlag som grenseverdien  $\Delta\sigma$  for den tidsderiverte kanalbitrate  $\dot{\sigma}$ , og fordelaktig skaleres i analogi med
- 20 skaleringen av  $\Delta\sigma$ .

- Hva enten den tidsderiverte eller integrasjonsresultatet benyttes som estimator for kanalbitraten, må det ved fastsettelsen av tidsintervaller og grenseverdier tas hensyn til at det både i kanalbitraten  $\sigma$  og kodingsbitraten  $t[c]$  vil opptre kortvarige, ikke-periodiske og slumpmessige fluktuasjoner.
- 25 Tidsintervall og grenseverdier må derfor velges signifikant større enn henholdsvis den maksimale varighet og amplitude til fluktuasjoner av denne art.

- I forbindelse med fremgangsmåten ved overføring av videoinformasjon i henhold til den foreliggende oppfinnelse, vil det være ønskelig at klienten kan eksekvere søking og gjenfinning av den overførbare videoinformasjon som er lagt inn i videofilen  $\Phi$  i form av videostrømmer  $IS$  og kodet med respektive forskjellige bitrater  $t[c]$  som svarer til klientens forventede
- 30

kanalbitrater  $\sigma$ . Da strømmene IS er dessuten er formatert med tanke på at overføringen skal kunne skje med en egnet versjon av HTTP som transportprotokoll, kan søkingen benytte den informasjon som allerede foreligger i toppteksten av  $H_{PC}$  til pakkene PC i de pakkeformaterte

5 videostrømmer IS. Videoinformasjon kan omfatte sekvenser av videobilder som har en innbyrdes semantisk relasjon, in casu videofilmer, men kan også bestå av enkeltbilder eller stillbilder. Det er derfor ønskelig at søking og gjenfinning av videoinformasjon er rettet mot enkeltbilder. På den annen side er det innlysende at en eksakt søking og gjenfinning av et enkeltbilde og

10 nedlasting av dette vil være svært ressurskrevende. Fremgangsmåten til klienteksekvert søking og gjenfinning av informasjon i henhold til oppfinnelsen er derfor utgangspunktet i søking etter enkeltbilder, men gjenfinningen finner sted ved å lokalisere dette enkeltbilde i en sekvens av bilder som kan omfatte opptil flere pakker PC i en videostrøm IS og deretter

15 å finne og laste ned pakken hvor enkeltbildet forekommer. I praksis innebærer dette at søkingen og gjenfinningen i henhold til oppfinnelsen realiseres som et pseudoslumpmessig søk. Dette oppnås ved å legge inn ytterligere informasjon i toppteksten til noen av pakkene i videostrømmen, slik dette skal forklares nærmere med henvisning til fig. 5a-5d.

20 Som vist på fig. 5a, deles en videofil i  $m$  grupper  $G_1 \dots G_m$ , og hver av disse gruppene omfatter et antall  $r$  pakker PC slik det skal forklares nærmere med henvisning til fig. 5b. Det skal forstås at antallet  $r$  godt kan være innbyrdes forskjellig i de forskjellige gruppene, men det vil være naturlig at  $r$  er et fast tall. Dersom antall pakker i strømmen da ikke er et multiplum av  $r$ , vil

25 naturligvis den siste gruppe i strømmen ha et antall pakker som er mindre enn det fast valgte tall  $r$ . En vilkårlig valgt gruppe  $G_k$ ,  $k \in m$ , omfatter derfor pakker  $PC_{k,1}, PC_{k,2}, \dots, PC_{k,r}$ . Den første påfølgende gruppe  $G_{k+1}$  omfatter da naturligvis tilsvarende pakker  $PC_{k+1}, PC_{k+1,2}, \dots, PC_{k+1,r}$ . Toppteksten  $H_{PC}$  i den første pakke  $PC_{k,1}$  i en vilkårlig gruppe  $G_k$ ,  $k \in m$ , må for å realisere

30 fremgangsmåten til søking omfatte ytterligere to informasjonsblokker utover de som er vist på fig. 3c, nemlig en informasjonsblokk  $D_j$  og en informasjonsblokk  $J$ . Informasjonsblokken  $D_j$  rommer informasjon om en hoppdistanse  $d_j$  som svarer til den samlede lengde av pakken i gruppen  $G_k$  og informasjonsblokken  $J$  rommer informasjon om antall bilder som er omfattet

35 av hoppdistansen  $d_j$  og den første pakke  $PC_{k+1,1}$  i den påfølgende gruppe

$G_{k+1}$ . Det vil nå ses at delingen i grupper  $G$  er konvensjonelt gitt ved  $d_j$ , og ikke fysisk.

- Ved fremgangsmåten til søking og gjenfinning i henhold til oppfinnelsen genereres først en anmodning til HTTP-tjeneren om nedlasting av
- 5 toppteksten  $H_P$  til den første pakke  $PC_{k,l}$  i en eksempelvis inntreffende gruppe  $G_k$ ,  $k \in m$ , i en videostrøm  $IS_k$ ,  $k \in y$ . Blokken  $D_j$  gir distansen  $d_j$  til begynnelsen av pakken  $PC_{k+1,1}$  som allerede nevnt og vist på fig. 5c, mens blokken  $J$  angir naturligvis da antall  $j$  videobilder omfattet av hoppdistansen og i den første pakke  $PC_{k+1,1}$  i påfølgende gruppe  $PC_{k+1}$ . Dersom nå det
- 10 ønskede bilde er  $F_x$ , sammenlignes nummeret  $x$  til dette bildet med  $j$ , og dersom  $x \in j$ , gjenopptas overføringen av dekodningen av videostrømmen fra og med første bilde i pakken hvor bildet  $F_x$  befinner seg, idet denne pakke må være en av pakkene i gruppen  $G_k$  samt den første pakke  $PC_{k+1,1}$  i den påfølgende gruppe  $G_{k+1}$ . Antall bilder  $n$  i hver pakke er nemlig gitt ved
- 15 blokken  $N$  i toppteksten til pakkene, og da det ses at
- $$j = n_{k,1} + n_{k,2} + \dots + n_{k,r} + n_{k+1,1}$$
- dvs. summen av alle bildene i gruppen  $G_k$  pluss antall bilder i den første påfølgende pakke  $PC_{k+1,1}$  i neste gruppe  $G_{k+1}$ , vil nå pakken hvor bildet  $F_x$  befinner seg, finnes og overføringen og dekodningen av videostrømmen skjer da fra første bilde i denne pakke.
- 20 Dersom nå  $x \neq j$  anmodes det om toppteksten  $HP$  til den første pakke  $PC_{k+1,1}$  i neste gruppe  $G_{k+1}$ , idet toppteksten til denne pakke likeledes inneholder blokken  $D_j$  med informasjon om hoppdistansen  $d_j$  for gruppen  $G_{k+1}$  og blokken  $J$  som angir antall bilder  $j$  til pakkene i denne gruppe og den første pakke  $G_{k+2,1}$  i den neste gruppe. Dersom nå  $x \in j$ , overføres og dekodes
- 25 videostrømmen fra og med første bilde i pakken hvor det søkte bilde befinner seg, og dersom  $x \notin j$ , gjentas prosessen på pakkene i den påfølgende gruppe inntil det ønskede bilde  $F_x$  er funnet. Dermed realiseres en bilderettet, men pakkeformatert søking og gjenfinning av videoinformasjon med bruk av en egnet versjon av HTTP som er transportprotokoll. Søkingen foregår i hopp
- 30 over en sekvens av hoppavstander  $d_j$ , noe som bidrar til å redusere søketiden. Når det ønskede videobildet befinner seg innenfor angitte bildenummer  $j$ , blir søkingen begrenset til å finne den pakke som inneholder det ønskede bilde, slik denne vil være gitt i toppteksten til den første pakke i gruppen ved blokkene  $D_j$  og  $J$ , samt blokken  $N$  i toppteksten til hver av pakkene som er
- 35 inkludert i gruppen og den første påfølgende pakke etter gruppen. Da søkingen er begrenset til å finne bare pakken som inneholder et ønsket bilde,

kan den betegnes som en pseudoslumpmessig søking. Det skal bemerkes at hoppdistansen  $d_j$  påvirker både administrasjonen og responstiden ved pseudoslumpmessig søking. Administrasjonen ved pseudoslumpmessig søking er nemlig omvendt proporsjonal med hoppdistansen  $d_j$ , idet en større hoppdistanse  $d_j$  resulterer i mindre pakkeadministrasjon. Bruk av en hoppdistanse på eksempelvis 25 pakker PC, dvs.  $r = 25$ , gir typisk en pakkeadministrasjon på 0,02 Kbit/s, noe som er ubetydelig. Søketiden domineres av rundreiseforsinkelsen i nettverket, da det ikke skjer noe pipelining av anmodningene og bare en del av pakketoppteksten  $H_{PC}$  strengt tatt er nødvendig. Det vil også ses at oppløsningen ved en slik pseudoslumpmessig søking bestemmes av pakkelengden  $q$ , som igjen hovedsakelig er gitt ved antall bilder  $n$  i hver pakke.

Søking og gjenfinning av videoinformasjonen kan finne sted allerede fra initieringen av overføringen, idet det da er naturlig å anmode om toppteksten til den første pakke  $PC_{1,1}$  i den første gruppe  $G_1$  i strømmen i den initiale videostrøm  $IS_k$ . Dersom en del av videostrømmen  $IS$  allerede er lastet ned og dekodet, kan naturligvis søking og gjenfinning starte fra en inntreffende pakke  $G_k$ , idet sekvensen av bilder  $F_a, \dots, F_b$  som ligger i klientens buffer da er bilder inneholdt i den foreliggende gruppes pakker. Det skal i den forbindelse bemerkes at bufferen i høyden lager et fåtall pakker, men bildene som foreligger på bufferen eksempelvis kan godt være fra den siste pakke i en gruppe  $G_{k-1}$  og fra den påfølgende gruppe  $G_k$ . Informasjonen om antall bilder og hoppdistanser  $d_j$  må derfor hensiktsmessig kunne relateres til posisjonen til henholdsvis det første bilde  $F_a$ , dvs. bildet med lavest nummer i bufferen, og det siste bilde  $F_b$ , dvs. bildet med høyest nummer i bufferen, som vist på fig. 6a. Dette skal forklares nærmere med henvisning til flytskjemaet på fig. 6b.

I henhold til flytskjemaet på fig. 6b spørres det i trinn 601 om  $x < F_a$ . Dette betyr at det søkte bildet  $F_x$  i en strøm  $IS$  har lavere nummer enn  $F_a$  og i trinn 602 anmodes det derfor om toppteksten  $H_{PC}$  til den aller første pakke  $PC_{1,1}$  i strømmen. I trinn 603 spørres det om  $x$  finnes blant bildene i denne pakke og er svaret ja, anmodes det om nyttelasten PL til pakken  $PL_{1,1}$  i trinn 604 og dekoding av nyttelasten i trinn 605, hvoretter bildet  $F_x$  vil være funnet og prosessen stopper. Er svaret nei, spørres det i trinn 606 om  $x$  finnes blant bildene  $j$  som er omfattet av hoppdistansen  $d_j$  som angitt i første pakke  $PC_{1,1}$  og bildene i den påfølgende gruppes første pakke  $PC_{2,1}$ . Et svaret ja, anmodes

det i trinn 607 om toppteksten  $H_{PC}$  til neste pakke, altså den annen pakke  $PC_{1,2}$  både i strømmen IS og den første gruppe  $G_1$  i strømmen IS. Det spørres så i trinn 608 om  $x$  er å finne blant bildenummerne  $n$  til denne pakken og dersom svaret er nei, skjer det en iterasjon i trinn 609 og prosessen gjentas

5 for neste pakke  $PC_{1,next}$  i trinn 608. Er svaret i ethvert tilfelle ja, anmodes det om nyttelasten til denne pakke i trinn 610 og i trinn 611 dekodes nyttelasten, hvoretter prosessen stopper. Er svaret i trinn 606 derimot at  $x$  ikke er inneholdt i  $j$ , fortsettes det til trinn 612, hvor det anmodes om toppteksten til den første pakke  $PC_{next,1}$  i den påfølgende gruppe  $G_{next}$ , altså den annen

10 gruppe i strømmen IS. Svares det i trinn 613 nei på spørsmålet om  $x$  er inneholdt i bildenummerne som er omfattet av hoppdistansen  $d_j$  og bildene til den første pakke i den påfølgende gruppe  $G_{next+1}$ , i dette tilfelle altså den tredje gruppe i strømmen IS, går man i trinn 614 tilbake til trinn 612, idet det nå iterativt anmodes om den første pakke i den neste gruppe  $G_{next+1}$ , altså i

15 det angjeldende tilfelle den tredje gruppe 3. Er svaret i 613 derimot ja, anmodes det om toppteksten  $H_{PC}$  til neste pakke  $PC_{next}$  i gruppen og er  $x$  ikke å finne blant bildenummerne  $n_{next}$  til denne pakke, går det via trinn 617 tilbake til trinn 615 og det anmodes om den neste påfølgende pakke, hvoretter prosessen fortsetter. Er derimot svaret i 616 ja, anmodes det om

20 nyttelasten til den angjeldende pakke og i trinn 619 dekodes nyttelasten, hvoretter prosessen stopper.

Er svaret i trinn 601 nei, går prosessen til trinn 621 hvor det spørres om  $x$  er større enn nummeret til det siste bildet  $F_b$  i bufferen. Er svaret nei, betyr det at det søkte bildet  $F_x$  er å finne blant bildene i bufferen og derfor umiddelbart

25 vil dekodes, hvorfor prosessen stopper. Er svaret derimot ja, dvs. at nummeret  $x$  til det søkte bildet  $F_x$  er større enn nummeret til  $F_b$ , anmodes det i trinn 622 om toppteksten til den påfølgende pakke  $PC_{k,\gamma}$  og det spørres i trinn 623 om  $x$  finnes blant bildenummerne i denne pakke. Er svaret ja, anmodes det i trinn 624 om nyttelasten  $H_{PC}$  til pakken  $PC_{k,\gamma}$ , og i trinn 625 dekodes

30 nyttelasten  $PL_{k,\gamma}$ , hvoretter prosessen stopper. Er svaret derimot nei i trinn 623, spørres det i trinn 626 om blokken  $J$  er å finne i toppteksten  $H_{PC}$  til pakken  $PC_{k,\gamma}$ . Er svaret nei, går det via trinn 627 tilbake til trinn 622 og det anmodes nå om toppteksten  $H_{PC}$  til en neste påfølgende pakke  $PC_{k,\gamma+1}$ , hvoretter prosessen fortsetter. Er svaret i trinn 626 derimot ja, spørres det i

35 trinn 628 om nummeret  $x$  er inneholdt i  $j$  som angir bildenummerne omfattet av hoppdistansen  $d_j$  og nyttelasten til første pakke  $PC_{k+1,1}$  i en påfølgende

- gruppe  $G_{k+1}$ . Er svaret i trinn 628 ja, anmodes det om toppteksten  $H_{PC}$  til den påfølgende pakke  $PC_{k,y+1}$  i trinn 629 og i trinn 630 spørres det om  $x$  er å finne blant bildenummerne i denne pakke. Dersom dette ikke er tilfellet, itereres det via trinn 631 og det anmodes om toppteksten til den neste påfølgende
- 5 pakke, hvoretter prosessen fortsetter. Er svaret i trinn 630 ja, anmodes det om nyttelasten  $PL_{k,y+1}$  til pakken  $PC_{k,y+1}$  i trinn 632 og i trinn 633 dekodes nyttelasten, hvoretter prosessen stopper. Er svaret i trinn 628 nei, anmodes det om toppteksten  $H_{PC}$  til den første pakke  $PC_{k+1,1}$  i den påfølgende gruppe  $G_{k+1}$  i trinn 634, og det spørres i trinn 635 om  $x$  er inneholdt i bildenummerne
- 10 omfattet av hoppdistansen  $d_j$  for gruppen  $G_{k+1}$  og den første pakke  $PC_{k+1,1}$  i den påfølgende gruppe  $G_{k+2}$ . Er svaret nei, itereres det i trinn 636 tilbake til trinn 634 og det anmodes nå om toppteksten til den første pakke i den neste påfølgende gruppe, hvoretter prosessen fortsetter. Er svaret i trinn 635 ja, anmodes det om toppteksten  $H_{PC}$  til neste pakke  $PC_{k+y,next}$ , altså om den annen
- 15 pakke i gruppen  $G_{k+y}$  i trinn 637, og er  $x$  ikke å finne blant bildenummerne i denne pakke, går det via trinn 639 videre til neste pakke osv., hvoretter prosessen fortsetter. Er svaret i trinn 638 ja, anmodes det om nyttelasten  $PL_{k+1,next}$  til pakken  $PC_{k+1,next}$  i trinn 640 og nyttelasten dekodes i trinn 641, hvoretter prosessen stopper.
- 20 På den her angitte måte blir det mulig å finne et ønsket bilde  $F_x$  ved at pakken som rommer  $F_x$ , finnes og dekodes. Søkingen etter  $F_x$  kan initieres under overføringen av en strøm som vist på fig. 6b, men selvsagt også initieres allerede når selve overføringen av strømmen starter. Det er underforstått at det under søkingen vil være mulig å svitsje mellom
- 25 strømmene IS ettersom kanalbitraten  $\sigma$  varierer med bruk av fremgangsmåten til overføring i henhold til oppfinnelsen.

- I forbindelse med både overføringen og søkingen genereres det og vedlikeholdes en stakklister for HTTP-anmodninger. En anmodning kan da legges i stakklister når den sendes fra klienten og fjernes først derfra etter at
- 30 den mottatte respons fra HTTP-tjeneren er prosessert. Dersom det skjer et brudd i forbindelsen mellom HTTP-tjener og klient under overføringen av videoinformasjon, under mottak av responsen, kan anmodningen i stakklister sendes på ny ved gjenopprettet forbindelse. Dersom det finner sted et brudd i forbindelse med mottaket av pakken, kan den første anmodning i listen
- 35 oppdateres og begynnelsen av pakken da justeres for de eventuelt allerede mottatte data.



## PATENTKRAV

1. Fremgangsmåte ved overføring på anmodning av videoinformasjon i en delt nettverkssjurs, hvor den delte nettverkssjurs spesielt er Internett, et intranett eller ekstranett, hvor videoinformasjonen er lagret i form av en kodet videofil på HTTP-tjenere i den delte nettverkssjurs og aksesserer av klienter via HTTP (Hypertext Transfer Protocol), hvor hver klient har en videodekoder, og hvor fremgangsmåten med utgangspunkt i initial videoinformasjon i form av digitaliserte videosignaler er karakterisert ved at den omfatter trinn for å generere ved hjelp av en videokoder  $y$  multiple, kodede videostrømmer (VS) som hver rommer den initiale videoinformasjons videosignaler komprimert med gjennomsnittlige bitrater  $t[c]$  som dekker klientenes forventede kanalbitrater  $\sigma$ , idet videokoderen genererer uavhengig dekodbare videobilder med gitte tidsintervaller;
- å generere  $y$  kodede mellomstrømmer (IS) fra de tilsvarende kodede videostrømmer (VS) ved å dele en kodet videostrøm i  $p$  pakker (PC) med varierende lengde  $q$ , idet hver pakke (PC) omfatter en topp tekst ( $H_{PC}$ ) og en nyttelast (PL) som rommer de kodede videosignaler for et tidssegment som tilsvarende nyttelasten;
- å forsyne topp teksten ( $H_{PC}$ ) med følgende informasjon:
- (i) distansene  $d_1$  og  $d_2$  henholdsvis til begynnelsen og slutten av den nærmest påfølgende pakke,
  - (ii) antall  $n$  videobilder (F) som pakken inneholder, samt
  - (iii) en referanse til den tilsvarende pakke i henholdsvis den kodede mellomstrøm ( $IS_b$ ) med den nærmestliggende lavere gjennomsnittlig bitrate  $t[c_{k-1}]$  og den kodede mellomstrøm ( $IS_a$ ) med den nærmestliggende høyere gjennomsnittlig bitrate  $t[c_{k+1}]$ , idet  $t[c_k]$  er bitraten for den foreliggende mellomstrøm ( $IS_k$ ) og  $k, b, a \in y$ ;
- å anordne et uavhengig dekodbart videobilde (IF) ved begynnelsen av nyttelasten (PL) til en pakke (PC);
- å sammenkjedde mellomstrømmene (IS) til en sluttfil ( $\Phi$ ) som lagres på en eller flere HTTP-tjenere; og å forsyne sluttfilen ( $\Phi$ ) med en topp tekst ( $H_\Phi$ ) som rommer informasjon om parametrene til strømmene (IS); og dessuten ved følgende trinn iverksatt av klienten:
- å generere en anmodning om topp teksten ( $H_\Phi$ ) og begynnelsen på den første strøm ( $IS_1$ ) i sluttfilen ( $\Phi$ );

å estimere kanalbitraten  $\sigma$  og å velge den strøm ( $IS_k$ ) hvis bitrate  $t[c_k]$  er den i forhold til den estimerte kanalbitrate  $\hat{\sigma}$  nærmestliggende bitrate, som den initiale strøm i overføringen, slik  $t[c_k] \approx \sigma$ , og deretter å estimere kanalbitraten  $\sigma$  under overføringen, og dersom estimatet  $\hat{\sigma}$  er lavere enn den gjennomsnittlig bitrate  $t[c_k]$  til den foreliggende strøm ( $IS_k$ ),  
 5 å svitsje til strømmen ( $IS_b$ ) med den nærmestliggende lavere gjennomsnittlige bitrate  $t[c_{k-1}]$ , eller dersom estimatet  $\hat{\sigma}$  er høyere enn gjennomsnittlige bitrate  $t[c_k]$  til den foreliggende strøm ( $IS_k$ ),  
 10 å svitsje til strømmen ( $IS_a$ ) med den nærmestliggende høyere gjennomsnittlige bitrate  $t[c_{k+1}]$ , idet svitsjingen av strømmene skjer på basis av pakkereferansene og realiserer en båndbreddeskalerbar videooverføring over HTTP.

2. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d at den båndbreddeskalerbare videooverføring  
 15 skjer over en versjon av HTTP som tillater vedvarende oppkobling og spesifisering av byteområde.

3. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d å estimere kanalbitraten  $\sigma$  på basis av en  
 estimator  $\hat{\sigma} = x/\tau$ , hvor  $\hat{\sigma}$  er den estimerte kanalbitrate og  $x$  antall bufrede  
 20 bit i tidsintervallet  $\tau$ , slik at dersom  $\hat{\sigma} > t[c_k]$  og bufferlengde mindre enn to ganger minimum pakkelenge, svitsjes det til strømmen med en  
 nærmestliggende høyere målbitrate  $t[c_a]$ , eller dersom  $\hat{\sigma} < t[c_k]$  og bufferlengde mindre enn minimum pakkelenge, svitsjes det til strømmen med en nærmestliggende lavere målbitrate  $t[c_b]$ .

25 4. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d estimere kanalbitraten  $\sigma$  på basis av en estimator  

$$d\sigma = \frac{d^2x}{dt^2}$$
, slik at dersom  $|\dot{\sigma}| \geq 0$ , svitsjes det til strømmen med en  
 nærmestliggende høyere målbitrate  $t[c_a]$ , eller dersom  $|\dot{\sigma}| < 0$  svitsjes det til strømmen med en nærmestliggende lavere målbitrate  $t[c_b]$ .

5. Fremgangsmåte i henhold til krav 4,  
 k a r a k t e r i s e r t v e d at det fastsettes en grenseverdi  $\Delta\dot{\sigma}$ , slik at  
 svitsjing finner sted dersom  $|\dot{\sigma}| \geq \Delta\dot{\sigma}$ .
6. Fremgangsmåte i henhold til krav 1,  
 5 k a r a k t e r i s e r t v e d å estimere kanalbitraten  $\sigma$  ved gjentatt å  
 integrere kanalbitraten  $\sigma$  over påfølgende tidsintervaller  $\tau = t_a - t_b$ , idet  $t_a$  og  
 $t_b$  henholdsvis er øvre og nedre grense for intervallet  $\tau$  og et  
 integrasjonsresultat  $\Sigma$  gitt ved  $\Sigma = \int_{t_b}^{t_a} \sigma dt$ , og å sammenligne  
 integrasjonsresultatene  $\Sigma_2, \Sigma_1$  for respektive påfølgende tidsintervall  $\tau_2, \tau_1$ ,  
 10 slik at dersom  $\Sigma_2 - \Sigma_1 > 0$ , svitsjes det til strømmen med en nærmestliggende  
 høyere målbitrate  $t[c_a]$ , eller dersom  $\Sigma_2 - \Sigma_1 < 0$ , svitsjes det til strømmen  
 med nærmestliggende lavere målbitrate  $t[c_b]$ .
7. Fremgangsmåte i henhold til krav 6,  
 k a r a k t e r i s e r t v e d at det fastsettes en grenseverdi  $\Delta\Sigma$ , slik at  
 15 svitsjing finner sted dersom  $|\Sigma_2 - \Sigma_1| \geq \Delta\Sigma$ .
8. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d å benytte en HTTP-tjener tilpasset  
 HTTP-versjon 1.1.
9. Fremgangsmåte i henhold til krav 8,  
 20 k a r a k t e r i s e r t v e d at videooverføringen finner sted over  
 HTTP-versjon 1.1.
10. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d å anordne pakkene (PC) i strømmene (IS) i ikke  
 overlappende suksessive grupper (G) av to eller flere påfølgende pakker, slik  
 25 hver strøm (IS) deles i  $m$  slike grupper, og å forsyne toppteksten til den første  
 pakke (PC<sub>k,1</sub>) i en gruppe (G<sub>k</sub>,  $k \in m$ ) med informasjon om en hoppdistanse  $d_j$   
 som svarer til den samlede lengde av pakkene i gruppen (G<sub>k</sub>) og antall  $j$   
 bilder (F) som hoppdistansen  $d_j$  og den første pakke i den påfølgende gruppe  
 (G<sub>k+1</sub>) omfatter.
- 30 11. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d å anordne strømmene (IS) slumpmessig i

sluttfilen ( $\Phi$ ), og at strømmen ( $IS_L$ ) med lavest bitrate  $t[c_L]$  bare refererer til pakker i strømmen med den nærmestliggende høyere bitrate og strømmen ( $IS_H$ ) med høyest bitrate  $t[c_H]$  bare refererer til pakker i strømmen med den nærmestliggende lavere bitrate.

- 5 12. Fremgangsmåte i henhold til krav 1, karakterisert ved å anordne strømmene ( $IS$ ) suksessivt med stigende bitrate  $t[c]$  i sluttfilen ( $\Phi$ ), slik at strømmen ( $IS_L$ ) med lavest bitrate  $t[c_L]$  utgjør den første videostrøm ( $IS_1$ ) i filen ( $\Phi$ ) og strømmen ( $IS_H$ ) med høyest bitrate  $t[c_H]$  utgjør den siste videostrøm ( $IS_y$ ) i filen ( $\Phi$ ).
- 10 13. Fremgangsmåte i henhold til krav 1, karakterisert ved at strømmene ( $IS_b$ ) og ( $IS_a$ ) rommer et uavhengig dekodebart videobilde ( $IF$ ) ved posisjoner svarende til begynnelsen av hver pakke ( $PC$ ) i en foreliggende strøm ( $IS_k$ ).
14. Fremgangsmåte i henhold til krav 1,
- 15 karakterisert ved at de kodede strømmen ( $VS;IS$ ) rommer det samme antall videobilder ( $F$ ).
15. Fremgangsmåte i henhold til krav 1, hvor strømmene ( $IS$ ) har forskjellig bilderate, karakterisert ved å definere et justeringsbilde ( $F_s$ ) som en egen bildetype, og å justere bilderatene i hver
- 20 strøm ( $IS$ ) til samme rate ved å innsette et passende antall justeringsbilder ( $F_s$ ) i de respektive strømmen.
16. Fremgangsmåte i henhold til krav 1, karakterisert ved at pakkelengden  $q$  er høyst lik en konfigurert anmodningspause for HTTP-tjeneren.
- 25 17. Fremgangsmåte i henhold til krav 1, karakterisert ved å prosessere de kodede strømmen ( $IS$ ) i parallell bilde for bilde.
18. Fremgangsmåte i henhold til krav 1, karakterisert ved at topp teksten ( $H_b$ ) til sluttfilen ( $\Phi$ ) inneholder
- 30 informasjon om antall  $y$  strømmen ( $IS$ ) i filen, den gjennomsnittlige bitrate  $t[c]$  for hver strøm, og distansene  $d_k$  og  $d_l$  fra sluttfilens begynnelse til henholdsvis begynnelsen av hver strøm og til slutten av den første pakke ( $PC_1$ ) i hver strøm ( $IS$ ).

19. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d at klienten på basis av videostrømmenes  
 parametre genererer undermengder av strømmene (IS), idet svitsjing mellom  
 strømmene (IS) bare kan finne sted i disse undermengder.
- 5 20. Fremgangsmåte i henhold til krav 1,  
 k a r a k t e r i s e r t v e d å generere og vedlikeholde en stakklister for  
 HTTP-anmodninger, idet en anmodning legges i stakklister ved sending og  
 fjernes derfra etter prosessering av den fra HTTP-tjeneren mottatte respons,  
 og at anmodningene i stakklister sendes på ny ved gjenopprettet forbindelse  
 10 etter et eventuelt brudd i forbindelsen mellom HTTP-tjener og klient under  
 mottak av responsen.
21. Fremgangsmåte i henhold til krav 20,  
 k a r a k t e r i s e r t v e d å oppdatere den første anmodning dersom et  
 brudd i forbindelsen finner sted under mottak av en pakke (PC), idet  
 15 begynnelsen av pakken (PC) justeres for de eventuelt allerede mottatte data.
22. Fremgangsmåte til klienteksekvert søking og gjenfinning av  
 videoinformasjon i en delt nettverkressurs, spesielt søking og gjenfinning av  
 et ønsket bilde ( $F_x$ ) i en videostrøm, hvor den delte nettverkressurs spesielt  
 er Internett, et intranett eller ekstranett, hvor videoinformasjonen er lagret i  
 20 form av en kodet videofil på HTTP-tjenere i den delte nettverkressurs og  
 aksesseres av klienter via HTTP (Hypertext Transfer Protocol), hvor hver  
 klient har en videodekoder, hvor den kodede videofil ( $\Phi$ ) er sammenkjedet av  
 multiple, kodede videostrømmer (IS) som rommer videoinformasjonens  
 videosignaler komprimert ved gjennomsnittlig bitrater  $t[c]$  som dekker  
 25 klientenes forventede kanalbitrater  $\sigma$ , hvor hver kodet videostrøm er delt i  $p$   
 pakker (PC) med varierende lengde  $q$ , hvor hver pakke (PC) omfatter en  
 topptekst ( $H_p$ ) og nyttelast (PL), hvor pakkene (PC) i en strøm (IS) er  
 anordnet i ikke-overlappende, suksessive grupper (G) av to eller flere  
 påfølgende pakker, slik at hver strøm (IS) deles i  $m$  slike grupper, hvor  
 30 toppteksten ( $H_{PC}$ ) til den første pakke ( $PC_{k,1}$ ) i hver gruppe ( $G_k$ ,  $k \in m$ ) i  
 tillegg til informasjon om antall  $n$  videobilder som pakken inneholder og  
 referanser til andre pakker og strømmer, forsynes med informasjon om en  
 hoppdistanse  $d_j$  som svarer til den samlede lengde av pakkene i gruppen og  
 antall  $j$  bilder (F) som hoppdistansen  $d_j$  og den første påfølgende pakke  
 35 ( $PC_{k+1,1}$ ) i den påfølgende gruppe ( $G_{k+1}$ ) omfatter, hvor videofilen ( $\Phi$ )

dessuten omfatter en topptekst ( $H_p$ ) som rommer informasjon om  
 parametrene til strømmene (IS), hvor informasjonen om strømmenes (IS)  
 parametre inkluderer distansene  $d_k$  og  $d_l$  fra videofilens ( $\Phi$ ) begynnelse til  
 henholdsvis begynnelsen av hver strøm (IS) og til slutten av den første pakke  
 5 (PC<sub>1</sub>) i hver strøm (IS), hvor overføringen av videoinformasjon skjer  
 båndbreddeskalert, og hvor fremgangsmåten er  
 k a r a k t e r i s e r t v e d  
 å generere en anmodning til HTTP-tjeneren om nedlasting av toppteksten  
 ( $H_p$ ) til den første pakke (PC<sub>k,1</sub>) i en gruppe ( $G_k, k \in m$ ) i en videostrøm  
 10 (IS<sub>k, k \in y</sub>) å sammenligne nummeret  $x$  til det ønskede bilde ( $F_x$ ) med  $j$ , og  
 dersom  $x \in j$ , å gjenoppta overføringen og dekodingen av videostrømmen fra  
 og med første bilde i pakken hvor bildet ( $F_x$ ) befinner seg, idet denne pakke  
 er en av pakkene i gruppen ( $G_k$ ) samt den første pakke (PC<sub>k+1,1</sub>) i den  
 påfølgende gruppe ( $G_{k+1}$ ), eller dersom  $x \notin j$ , å anmode om toppteksten ( $H_p$ )  
 15 til den første pakke (PC<sub>k+1,1</sub>) i den påfølgende gruppe ( $G_{k+1}$ ), og eventuelt å  
 fortsette prosessen inntil det ønskede bilde ( $F_x$ ) er funnet, hvorefter  
 nedlasting og dekoding av strømmen (IS) gjenopptas fra og med første bilde i  
 pakken hvor det ønskede bilde ( $F_x$ ) befinner seg, slik at det realiseres en  
 bildebasert og pakkeformatert søking og gjenfinning av videoinformasjon  
 20 med bruk av HTTP som transportprotokoll.

23. Fremgangsmåte i henhold til krav 22,  
 k a r a k t e r i s e r t v e d at det benyttes en versjon av HTTP som tillater  
 vedvarende oppkobling og spesifisering av byteområde.

24. Fremgangsmåte i henhold til krav 22,  
 25 k a r a k t e r i s e r t v e d å generere og vedlikeholde en stakklister for  
 HTTP-anmodninger, idet en anmodning legges i stakklister ved sending og  
 fjernes derfra etter prosessering av den fra HTTP-tjeneren mottatte respons,  
 og at anmodningene i stakklister sendes på ny ved gjenopprettet forbindelse  
 etter et eventuelt brudd i forbindelsen mellom HTTP-tjener og klient under  
 30 mottak av responsen.

25. Fremgangsmåte i henhold til krav 24,  
 k a r a k t e r i s e r t v e d å oppdatere den første anmodning dersom et  
 brudd i forbindelsen finner sted under mottak av en pakke (PC), idet  
 begynnelsen av pakken (PC) justeres for de eventuelt allerede mottatte data.

26. Fremgangsmåte i henhold til krav 22,  
karakterisert ved å benytte en HTTP-tjener tilpasset  
HTTP-versjon 1.1, og at HTTP-versjon 1.1 benyttes som transportprotokoll.

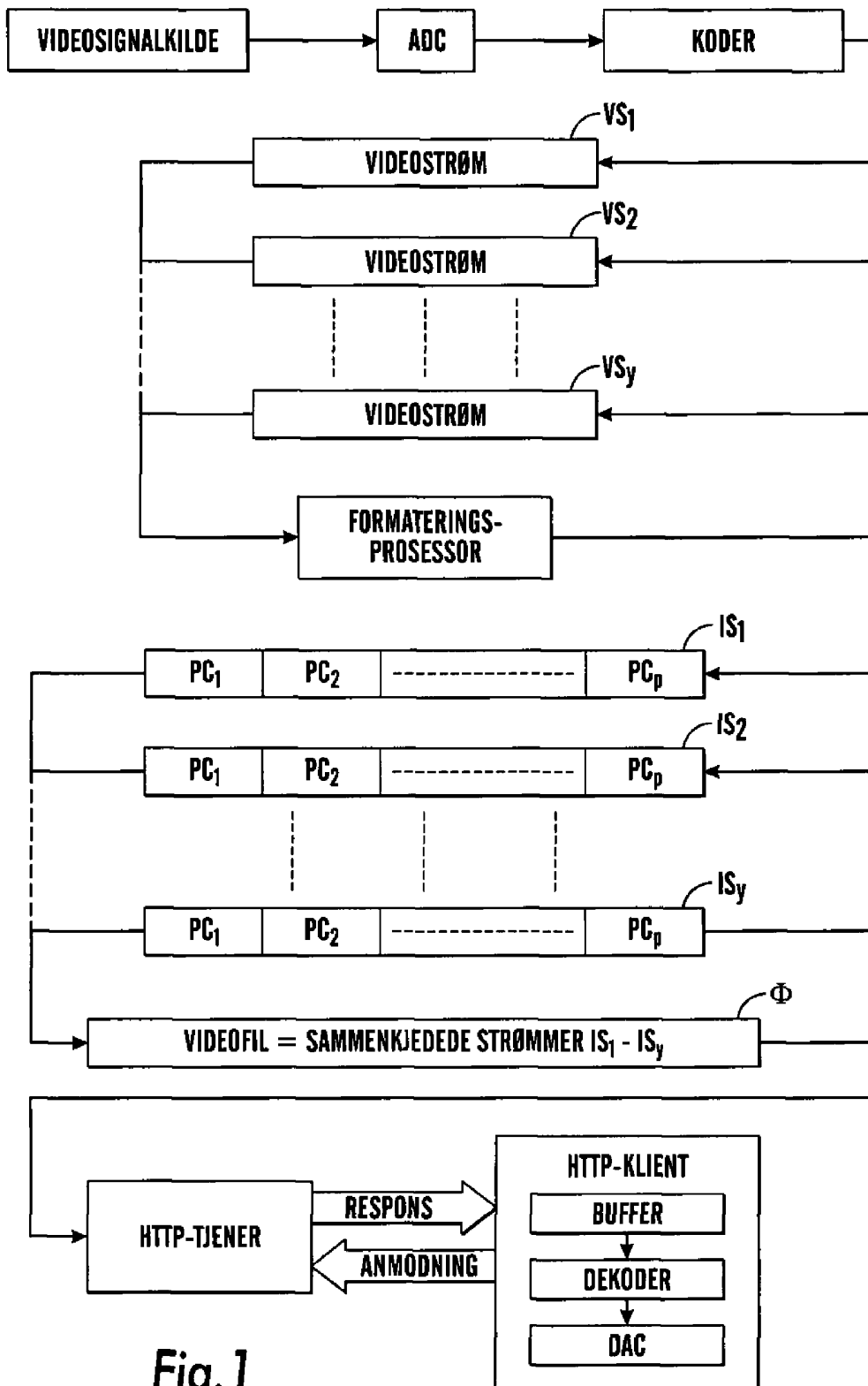
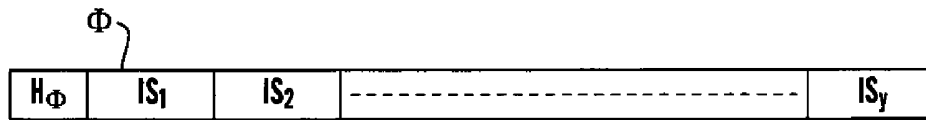
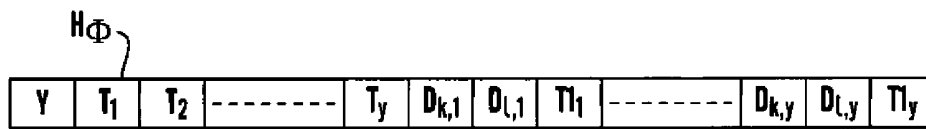


Fig. 1

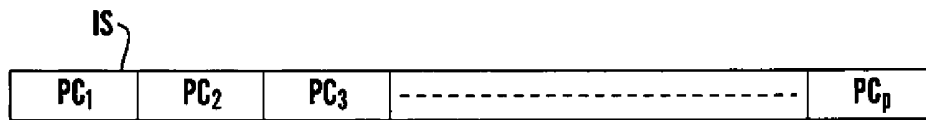




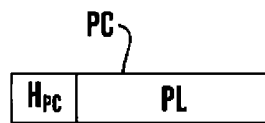
**Fig. 2a**



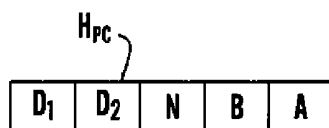
**Fig. 2b**



**Fig. 3a**



**Fig. 3b**



**Fig. 3c**

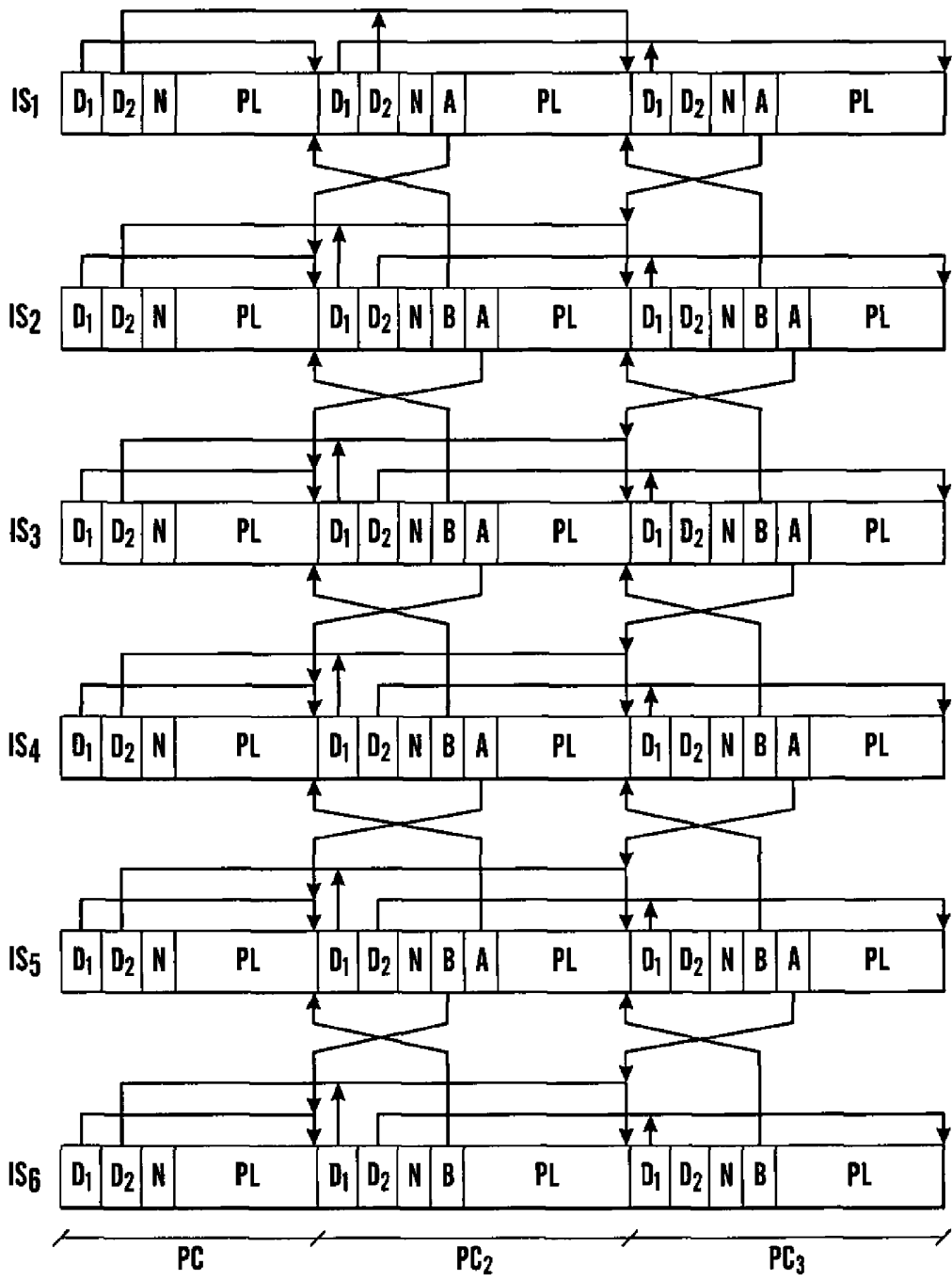


Fig.4

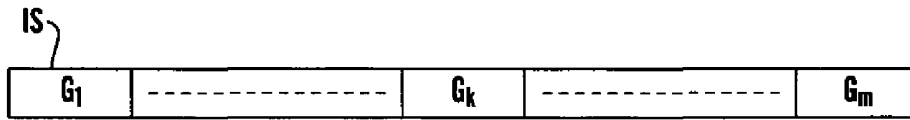


Fig. 5a

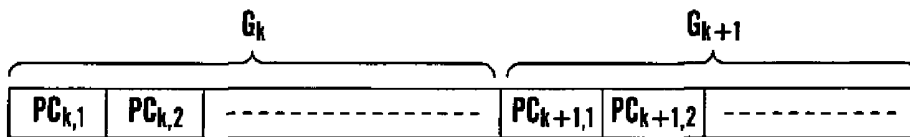


Fig. 5b

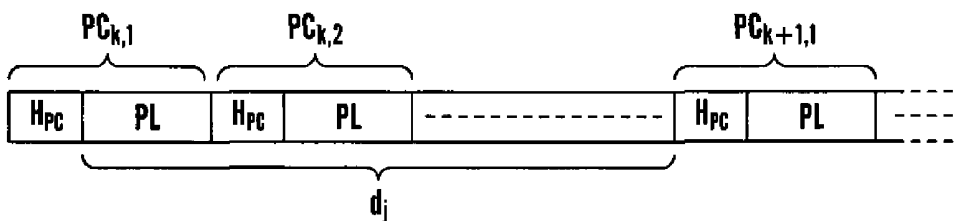


Fig. 5c

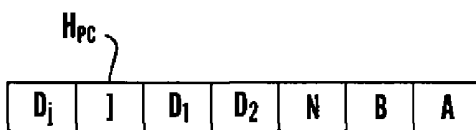


Fig. 5d

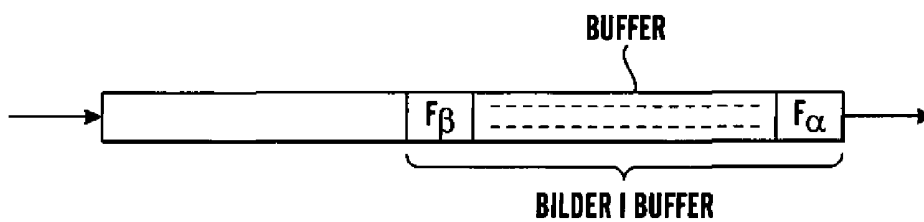


Fig. 6a

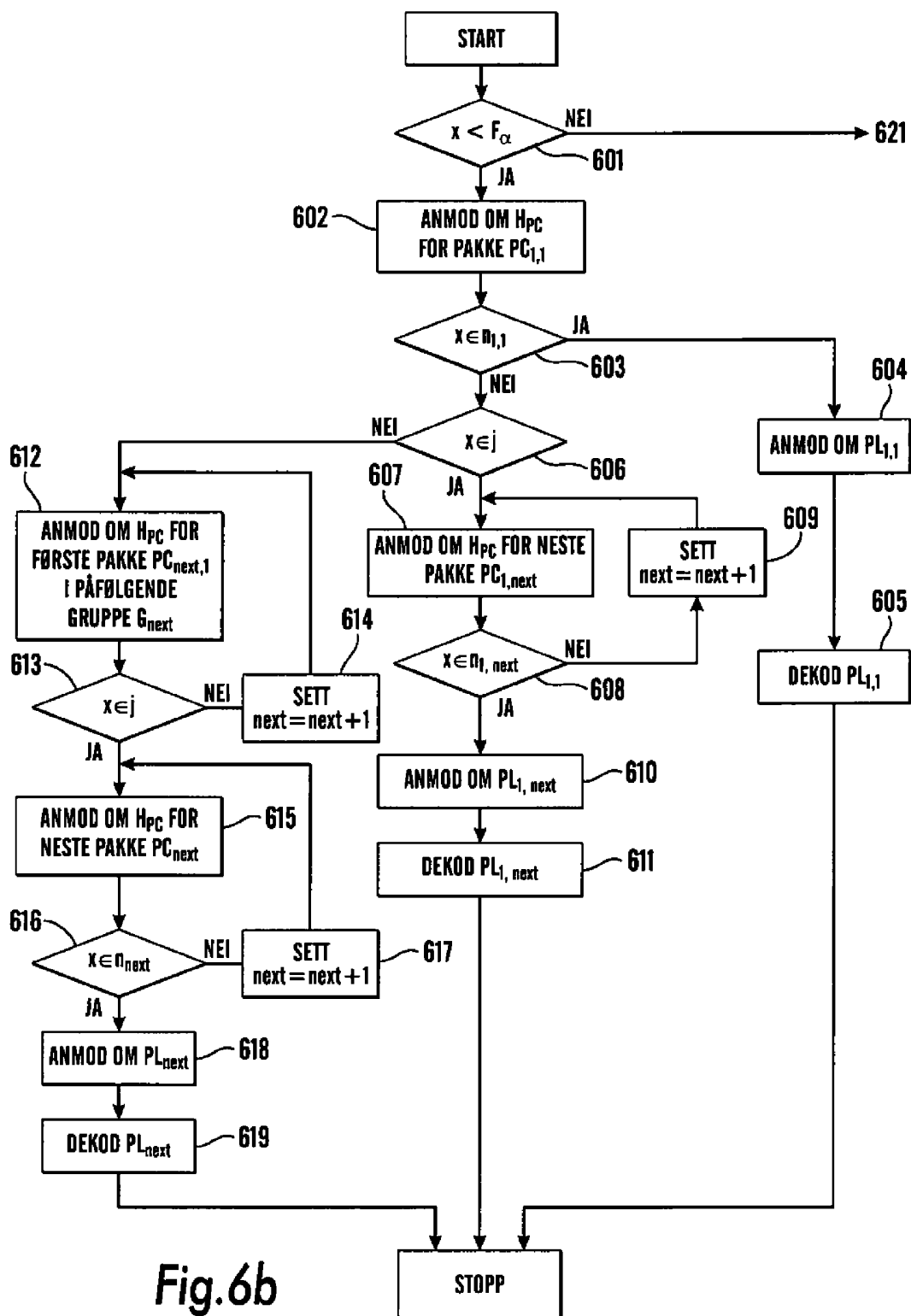


Fig. 6b

FRA TRINN 601

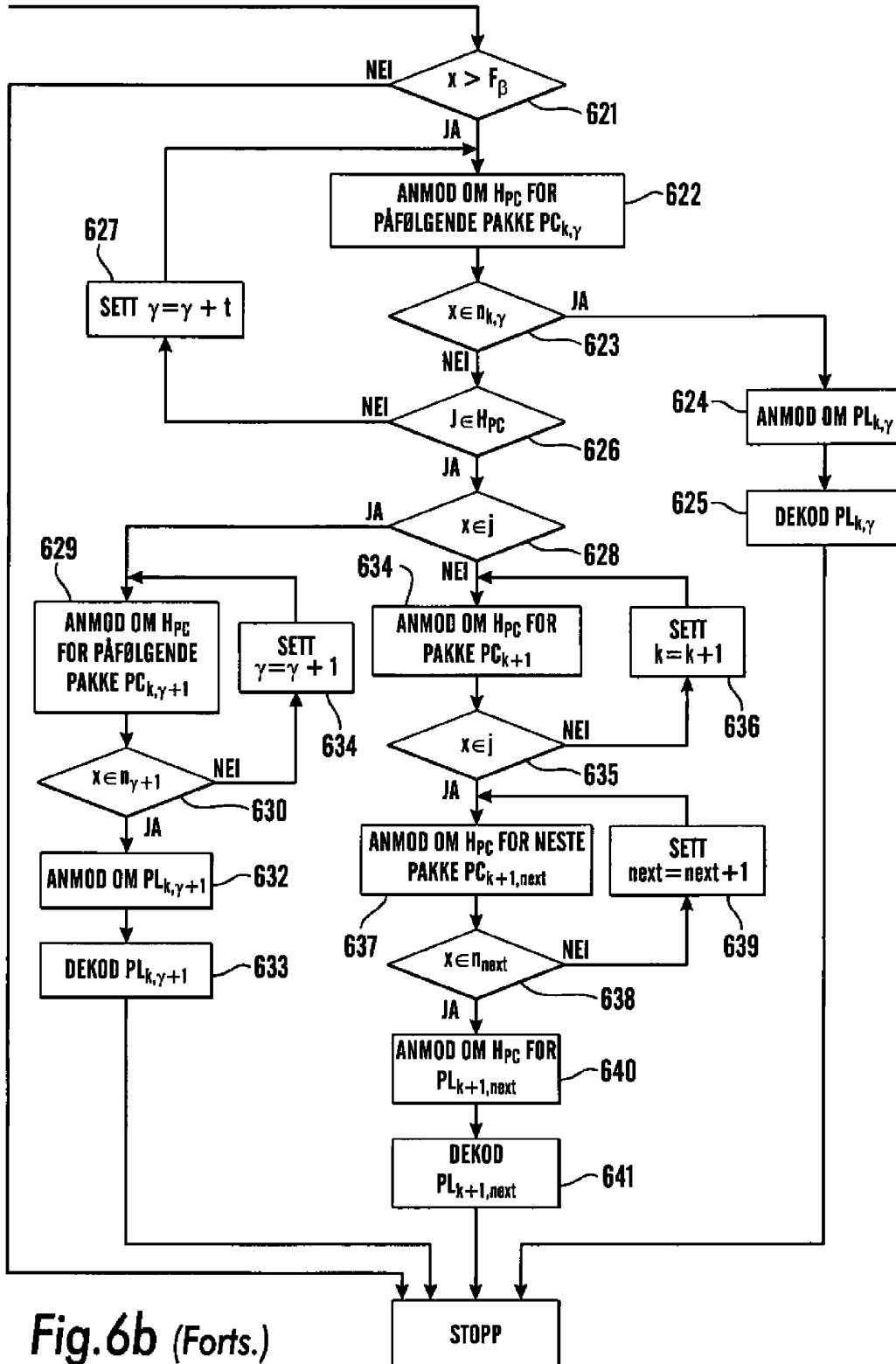


Fig. 6b (Forts.)

FRA TRINN 601

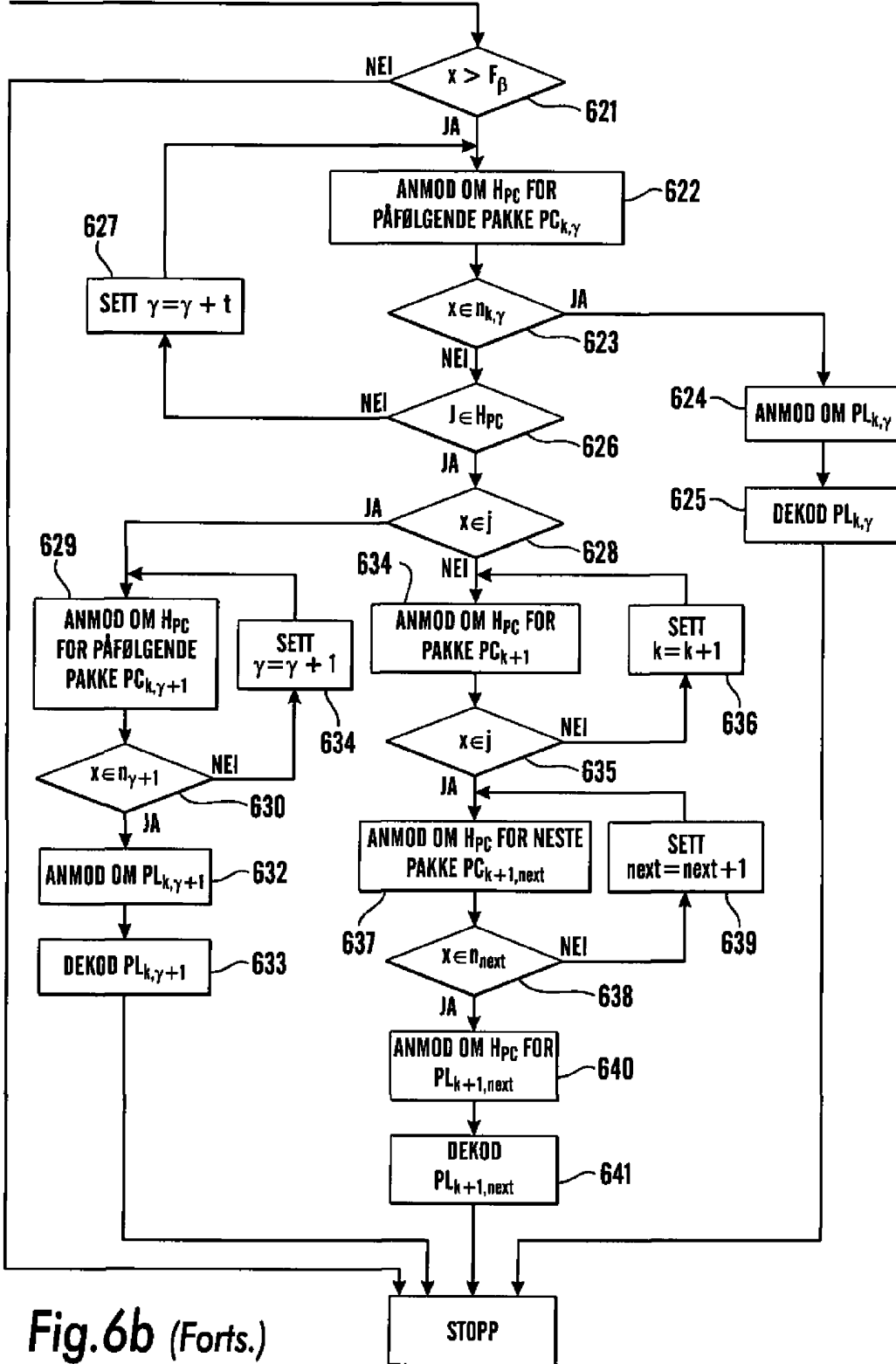


Fig.6b (Forts.)

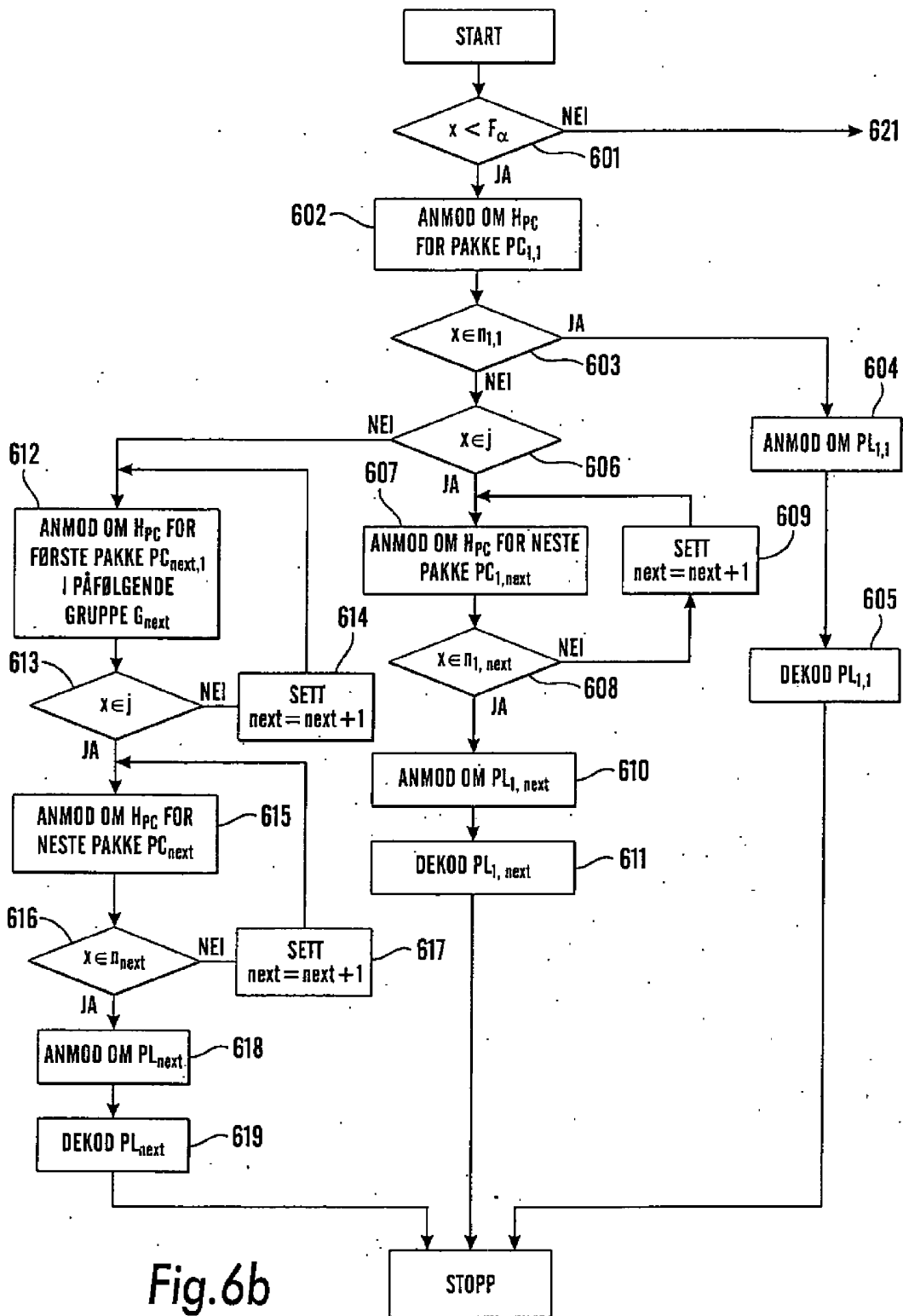


Fig.6b

FRA TRINN 601

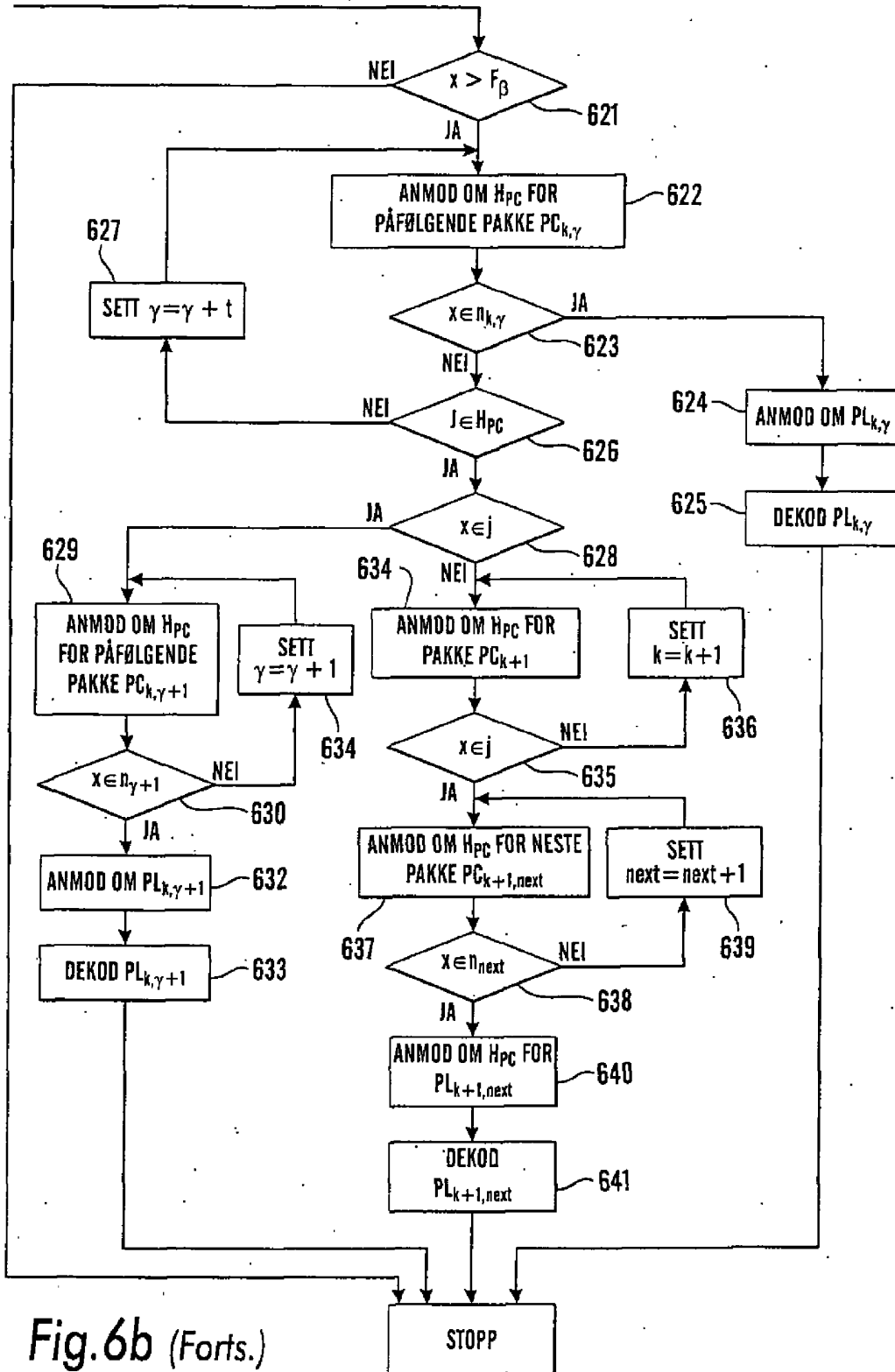


Fig. 6b (Forts.)



FRA TRINN 601

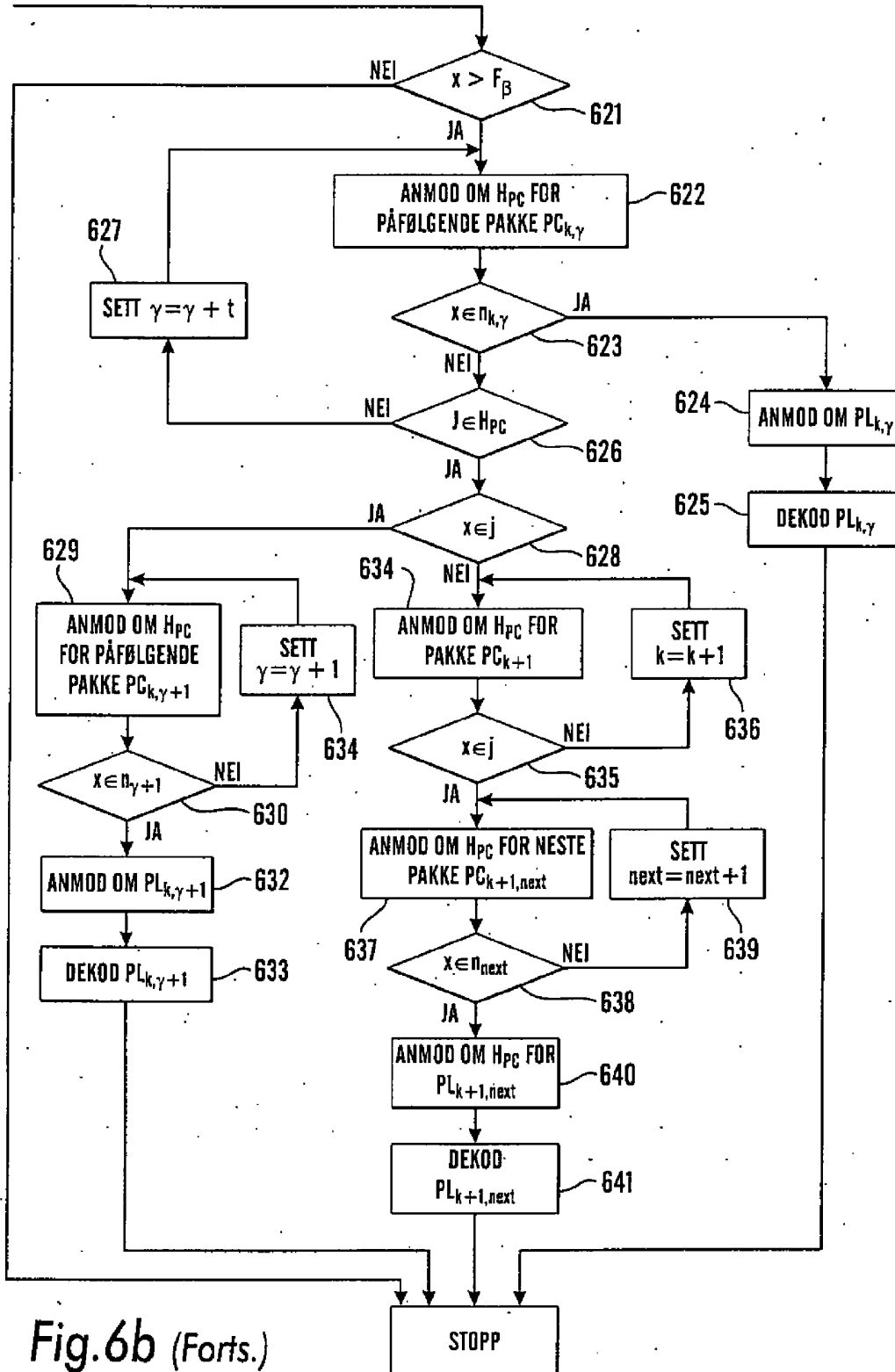


Fig. 6b (Forts.)