



(12) **PATENT**

(19) **NO**

(11) **332162**

(13) **B1**

**NORGE**

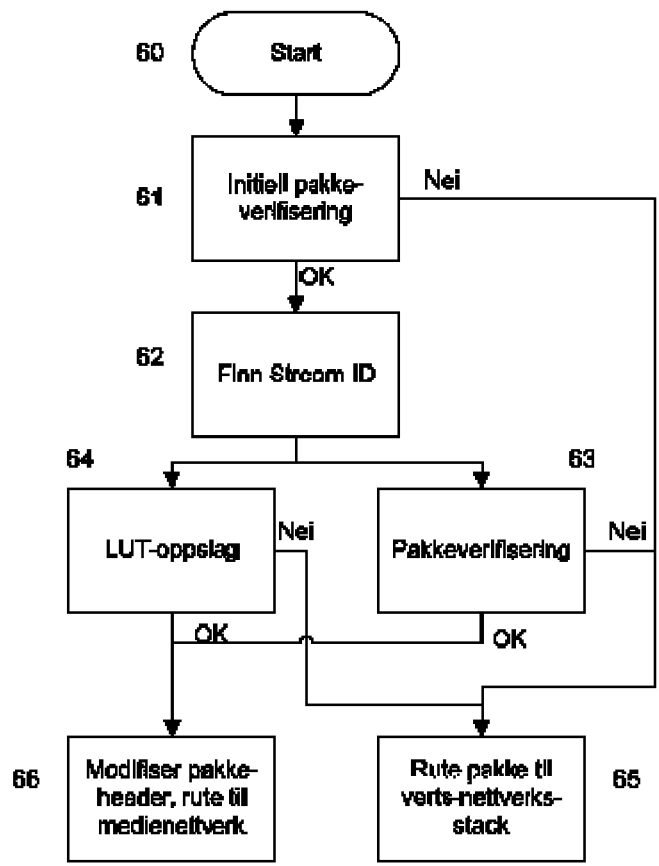
(51) Int Cl.  
*H04L 12/56 (2006.01)*

**Patentstyret**

(21)	Søknadsnr	20093566	(86)	Int.inng.dag og søknadsnr
(22)	Inng.dag	2009.12.21	(85)	Videreføringsdag
(24)	Løpedag	2009.12.21	(30)	Prioritet
(41)	Alm.tilgj	2011.06.22		
(45)	Meddelt	2012.07.09		
(73)	Innehaver	Cisco Systems International SarL, Avenue des Uttins 5, CH-1180 ROLLE, Sveits		
(72)	Oppfinner	Simon Robbins, 14, Mendip Close, GB-RG109QU CHARVIL, BERKSHIRE, Storbritannia		
(74)	Fullmektig	Onsagers AS, Postboks 1813 Vika, 0123 OSLO, Norge		

(54)	Benevnelse	<b>Anordning og fremgangsmåte for å filtrere mediapakker</b>
(56)	Anførte publikasjoner	US 6687247 B1, XP002574536 - tidsfestet Wikipedia-artikkel "Forwarding plane", EP 1528709 A2
(57)	Sammendrag	

En fremgangsmåte i en videokonferanseinnretning for å filtrere mediapakker fra en videokonferansedatastrøm. Videokonferanseinnretningen omfatter et medieprosesseringsnettverk, et generelt nettverk, en soft-core-prosessor, et Look-up Table minne (LUT), og en fullnettverksstakk. Fremgangsmåten omfatter å motta ved soft-core-prosessoren, en pakke i en videokonferansemediestrøm, å bestemme hvorvidt lengden av pakken er tilstrekkelig lang til å inneholde media, å bestemme hvorvidt protokollen for pakken er støttet av soft-core-prosessoren, å finne en mediestrøm-ID for pakken, å sende en forespørsel til LUT ved bruk av strøm-ID'en som inngangsverdi mens det i parallell bestemmes hvorvidt pakken er en gyldig mediepakke, og ved mottak av en destinasjonsadresse i medieprosesseringsnettverket fra LUT, og bestemmelse av at pakken er en gyldig mediepakke, å modifisere headeren for pakken med destinasjonsadressen mottatt fra LUT, og å rute pakken til den modifiserte destinasjonsadressen. En videokonferanseinnretning er tilpasset å utføre fremgangsmåten.



### Teknisk område

Den foreliggende oppfinnelsen vedrører generelt prosessering av videokonferansedata. Mer spesifikt vedrører oppfinnelsen en fremgangsmåte for å filtrere mediepakker fra en videokonferansedatastrøm, og en

5 videokonferanseinnretning, slik som en multipoint control unit, som utfører fremgangsmåten.

### Bakgrunn

Videokonferanser og assosiert maskinvare faller grovt inn i to grupper. I den første gruppen opptrer "konferansen" mellom bare to deltakere, og deltakerne er forbundet

10 direkte til hverandre gjennom en form for datanettverk. I denne formen av nettverk er bare to endepunkter involvert, og sann konferanse opptrer bare dersom flere deltakere er til stede ved ett av de to endepunkt-stedene. Eksempler på denne typen konferanser er, ved lavteknologienden, PC-utførte endepunkter sammenbundet ved hjelp av programvare så som NetMeeting® eller Skype®, og ved den høyere enden,

15 utstyr som benytter dedikert endepunktmaskinvare som er forbundet til hverandre, f.eks. via ISDN eller IP- (Internet Protocol-) lenker.

I den andre gruppen tillater videokonferanser flere enn to endepunkter å samvirke med hverandre. Dette oppnås ved å tilveiebringe minst ett sentralisert koordinerende punkt; en såkalt "multipoint control unit (MCU)", som mottar video- og

20 audiostrømmer (eng.: video and audio streams) fra endepunktene, kombinerer disse på ønsket måte, og retransmitterer den kombinerte sammensatte video-/audiostrøm til deltakerne. Ofte er konferansebetraktningen (eng.: the conference view) som transmitteres til endepunktene den samme for hvert endepunkt. Sammensetningen kan endres over tid, men er den samme for alle deltakerne.

25 Tilveiebringelsen av bare en enkelt sammensetning er et signifikant problem, fordi hver deltaker derfor må motta en konferansestrøm skreddersydd slik at den er akseptabel for det endepunktet i konferansen som har dårligst ytelse. I denne situasjonen utnyttes derfor mange endepunkter ikke med sin fulle kapasitet, og de kan oppleve degraderte bilder og lyd som resultat.

30 I det siste har moderne MCU'er blitt utformet for å tillate at en unik betraktning (eng.: view) dannes for hver deltaker. Dette muliggjør at den fullstendige kapabilitet for hvert endepunkt kan utnyttes, og det tillater også ulike sammensetninger for ulike deltakere, slik at f.eks. fremhevelsen av en bestemt deltaker i konferansen kan være forskjellig for en annen bruker. Imidlertid er prosessering av videodata i

35 sanntid en i høy grad prosessorintensiv oppgave. Den involverer også flytting av store mengder data. Dette er særlig tilfelle så snart dataene har blitt dekomprimert for å utføre høykvalitetsprosessering. Således er prosesseringskraft og båndbreddebegrensninger en signifikant flaskehals i dannelsen av høykvalitets videokonferanse MCU'er som muliggjør produksjon av multiple betraktninger av

40 konferansen.

Fig. 1 viser en typisk tidligere kjent MCU-arkitektur. Eksempelarkitekturen har et flertall av digitale signalprosessorer 2 slik som Texas Instruments TMS-serien, som er forbundet via en tidsdelt multiplekset (TDM) buss 4. En kontroller og nettverkgrensesnitt 6 er også forbundet til TDM-bussen. Hver DSP 2 er allokert én eller flere tidsluker på TDM-bussen. Det vil forstås at TDM-bussen er en signifikant flaskehals. Selv om økt prosesseringskraft for MCU'en kan oppnås ved å tilføye kraftigere DSP'er eller ytterligere DSP'er, må all data som flyter mellom DSP'ene og mellom nettverket 8 og DSP'ene passe i et endelig antall tidsluker på TDM-bussen 4. Derfor vil denne formen av arkitektur generelt skalere dårlig, og den kan ikke tilfredsstillere prosesseringskravene for pr. deltaker sammensetninger.

Fig. 2 viser en alternativ tidligere kjent konfigurasjon. I dette eksemplet blir et flertall av DSP'er 2-1 hver forbundet til en Peripheral Component Interconnect (PCI) buss 10-1. Tilsvarende er et flertall av DSP'er 2-2, 2-3 og 2-4 forbundet til respektive PCI-busser 10-2, 10-3 og 10-4. PCI-bussene 10-2, 10-3 og 10-4 er i sin tur forbundet via buffere 12 til en ytterligere PCI-buss 14. En signifikant fordel ved denne arkitekturen, sammenlignet med den som er vist i fig. 1, er at DSP'ene i gruppe 2-1 kan kommunisere sammen med hverandre, der den eneste flaskehalsen er PCI-bussen 10-1. Dette er også tilfelle for gruppene 2-2, 2-3 og 2-4. Imidlertid, dersom en DSP i gruppe 2-1 skulle ønske å kommunisere med en DSP f.eks. i gruppe 2-3, må PCI-bussen 14 benyttes. Derfor, selv om denne arkitekturen er en signifikant forbedring sammenlignet med den som er vist i fig. 1, hva angår skalerbarhet og evne til effektivt å bruke et flertall av DSP'er, må PCI-bussen 14 fortsatt benyttes for bestemte kombinasjoner av intra DSP-kommunikasjon, og således bli en ytelsesbegrensende faktor for MCU-arkitekturen.

Det har blitt gjort forsøk på å avlaste prosessering fra DSP'ene. F.eks. produserer IDT en "Pre-processing switch (PPS)", under delenr. IDT 70K2000, for bruk med DSP'er. DSP'en utfører forhåndsbestemte funksjoner før levering til en prosessor slik som en DSP eller FPGA. Prosessering bestemmes basert på adresseområdet for svitsjen som pakkene sendes til. Brikken er utformet f.eks. for bruk i 3G mobiltelefoni, og er utformet f.eks. for å avlaste grunnleggende oppgaver fra DSP'er som normalt ville utføres ineffektivt av DSP'en.

En tredje MCU-arkitektur som tilveiebringer en høy skalerbarhet og svært effektiv prosesseringsplattform er beskrevet i US 20080158338 og US 20090213126. Fig. 3 viser et hovedkort 20, hvilket hovedkort bærer en field programmable gate array (FPGA) 24 og et flertall datterkort 22. FPGA 24 ruter data mellom en kontroller (ikke vist), nettverkgrensesnitt (ikke vist) og flertallet av datterkort 22. Linkene som forbinder hovedkortet 20 med det første laget av datterkort kan ha en båndbredde f.eks. på 3 Gb/sek. eller høyere. Hvert datterkort har et flertall av prosessorer, dvs. digitale signalprosessorer (DSP'er) forbundet via en datterkortsvidsj. Hver datterkortsvidsj er konfigurert til å svitsje data mellom flertallet av DSP'er og mellom hovedkortet, datterkort og andre datterkort. I ett

eksempel, og med henvisning også til fig. 4, har hvert datterkort 20 fire DSP'er 28, hvert med assosiert minne 30. Hvert datterkort har også en FPGA 32 som innbefatter en svitsj 34. FPGA 32 innbefatter også prosessorer 36, og to linker med høy båndbredde 38. Selv om denne arkitekturen er en stor forbedring sammenlignet med den alternative tidligere kjente teknikk, siden kort til kort kommunikasjon i stor grad reduseres, er kort til kort kommunikasjonen fortsatt avhengig av en prosessor som filtrerer pakker og redistribuerer mediepakker til DSP'en ved bruk av full nettverksstakk. Dette danner en unødvendig byrde på prosessoren, som reduserer hastigheten for systemet. Selv om det særlig er nevnt i relasjon til arkitekturen i fig. 3 og 4, er dette et enda større problem for PSI-bussbaserte MCU-arkitekturer vist i fig. 2.

### Sammenfatning av oppfinnelsen

Det er et behov i teknikken for en fremgangsmåte og innretning for å filtrere mediepakker fra videokonferansemediestrømmer uten å måtte ty til en full nettverksstakk. Fremgangsmåten og innretningen ifølge den foreliggende oppfinnelsen tillater filtrering av mediepakker fra videokonferansemediestrømmer implementert på en liten soft-core-prosessor for å verifisere, klassifisere og redigere mediepakker ved gigabit eternett linjehastighet.

Oppfinnelsen er angitt ved de vedføyde krav.

I samsvar med et første aspekt omfatter den foreliggende oppfinnelsen en fremgangsmåte for å filtrere mediepakker fra en videokonferansedatastrøm (eng.: a video conferencing data stream) i en videokonferanseinnretning, idet videokonferanseinnretningen omfatter et medieprosesseringsnettverk, et generelt nettverk, en soft-core prosessor, et Look-up Table minne (LUT) og en full nettverksstakk. Soft-core-prosessoren mottar en pakke av en videokonferansemediestrøm, og bestemmer først hvorvidt lengden av pakken er tilstrekkelig lang til å inneholde media, deretter bestemmer soft-core-prosessoren hvorvidt den støtter protokollen for pakken. Etter de initielle verifikasjonstrinnene finner soft-core-prosessoren en mediestrøm-ID for pakken, og sender en forespørsel til LUT ved bruk av strøm-ID'en som en inngangsverdi, mens den i parallell bestemmer hvorvidt pakken er en gyldig mediepakke. Når soft-core-prosessoren mottar en destinasjonsadresse i medieprosesseringsnettverket fra LUT, og pakken er bestemt å være en gyldig mediepakke, modifiserer soft-core-prosessoren headeren i pakken med destinasjonsadresse mottatt fra LUT, og ruter pakken til den modifiserte destinasjonsadressen.

I samsvar med et andre aspekt ved den foreliggende oppfinnelsen, når det bestemmes at en pakke er en gyldig mediepakke uten at det mottas en destinasjonsadresse i medieprosesseringsnettverket fra LUT, rutes pakken til den fulle nettverksstakken, der en destinasjonsadresse for pakken i medienettverket gjenvinnes, og pakken rutes til destinasjonsadressen i medienettverket. Deretter

oppdateres LUT med destinasjonsadressen i medienettverket ved bruk av den korresponderende mediestrøm-ID'en for pakken som indeksverdi.

5 I samsvar med et tredje aspekt ved den foreliggende oppfinnelsen blir en pakke som bestemmes ikke å være en gyldig mediepakke, en sannsynlig mediepakke og/eller er en protokoll som ikke støttes av soft-core-prosessoren, rutes direkte til den fulle nettverksstakken.

Andre aspekter og trekk ved den foreliggende oppfinnelsen vil fremstå for fagfolk ved gjennomgang av den følgende beskrivelsen av spesifikke utførelsesformer av oppfinnelsen sammenholdt med de vedføyde tegninger.

#### 10 **Kort beskrivelse av tegningene**

Fig. 1 er et skjematisk blokkdiagram av en tidligere kjent MCU-arkitektur;

Fig. 2 er et skjematisk blokkdiagram av en alternativ tidligere kjent MCU-arkitektur;

15 Fig. 3 er et skjematisk blokkdiagram som viser en annen alternativt tidligere kjent MCU-arkitektur;

Fig. 4 er et skjematisk blokkdiagram av et tidligere kjent datterkort i samsvar med den tidligere kjente MCU-arkitekturen i fig. 3;

Fig. 5 er et skjematisk blokkdiagram av en mediepakkefiltreringsenhet i samsvar med en eksempelutførelsesform av den foreliggende oppfinnelsen;

20 Fig. 6 er et flytskjema for en fremgangsmåte for mediepakkefiltrering i samsvar med den foreliggende oppfinnelsen; og

Fig. 7 er et annet eksempelflytskjema for en fremgangsmåte for mediepakkefiltrering i samsvar med den foreliggende oppfinnelsen.

#### **Detaljert beskrivelse**

25 Den følgende beskrivelsen presenteres for å sette en ordinær fagperson i stand til å lage og benytte de ulike aspekter og eksemplene ved oppfinnelsen. Beskrivelser av spesifikke innretninger, teknikker og anvendelser er tilveiebrakt bare som eksempler. Ulike modifikasjoner til eksemplene beskrevet her vil tydelig fremstå for ordinære fagfolk, og de generelle prinsippene definert her kan anvendes med andre  
30 eksempler og anvendelser, uten å forlate oppfinnelsens ånd og rekkevidde. Således er den foreliggende oppfinnelsen ikke tiltenkt å være begrenset til eksemplene som er beskrevet og vist her, men den skal gis en rekkevidde konsistent med kravene.

Med henvisning til fig. 5 omfatter mediepakkefilterenheten 50 en tri-FIFO 51, en soft-core-prosessor 52 og en Lookup Table, LUT, 59. tri-FIFO'en 51 er enten en  
35 synkron eller asynkron FIFO (First-in First-out) som har tre porter 53, 54 og 55. Den første porten er en datainngangsport 53, typisk forbundet til et datastreamingsnettverk, idet porten er konfigurert til å motta data og korresponderende kontrollsignaler i samsvar med en nettverksprotokoll for

datastreamingsnettverket. Den tredje porten er en datautgangsport 55, forbundet til et medieprosesseringsnettverk. Medieprosesseringsnettverket omfatter typisk et flertall av DSP'er, svitsjer, osv., som beskrevet ovenfor med henvisning til fig. 1, 2 og 3. Medieprosesseringsnettverket er også forbundet til et generelt nettverk i verts MCU'en, idet det generelle nettverket kontrolleres av en full nettverksstakk, eller vertsnettverksstakk 58. Vertsnettverksstakken er også i operasjonell forbindelse med LUT 59. Datautgangsporten 55 er konfigurert til å strøme (eng.: stream) mediedata inn i medieprosesseringsnettverket i samsvar med en nettverksprotokoll for medieprosesseringsnettverket. Alternativt er datautgangsporten 55 konfigurert som en random access-leseport. Den andre porten er en soft-core-prosessorgrensesnittport 54 forbundet til soft-core-prosessoren 52. Grensesnittporten 54 er konfigurert til å sende og motta data til og fra soft-core-prosessoren 52. I en eksempelutførelsesform av den foreliggende oppfinnelsen er grensesnittporten 54 konfigurert som en lese-/skrive random access-port. Videre omfatter tri-FIFO'en også i det minste tre, fortrinnsvis minst fire, datapakkebufferne 56 som sendes mellom de tre portene, i sløyfe fra port 53 til port 54, port 54-55 og til slutt returnerende til port 53 fra port 55. Datapakkebufferne er konvensjonelle datapakkebufferne som er kjent for fagfolk.

Soft-core-prosessoren 52 som er kjent for fagfolk, er en mikroprosessorkjerne i sin helhet implementert ved bruk av logikksyntese. Den kan implementeres i ulike halvlederinnretninger som inneholder programmerbar logikk (f.eks. FPGA, CPLD). Soft-core-prosessoren i den foreliggende oppfinnelsen kan være en 8 bits, 16 bits eller 32 bits prosessor. Soft-core-prosessoren 52 kommuniserer videre med LUT 59 (Look-Up Table memory).

LUT 59 benytter et raskt minne for å holde informasjon beskrevet i nærmere detalj ved beskrivelse av funksjonaliteten for LUT nedenfor. I én eksempelutførelsesform er LUT implementert på samme brikke som soft-core-prosessoren 52, det raske minnet er foretrukket implementert som on-chip SRAM (internt FPGA-minne, f.eks. "blokk RAM"). I en annen eksempelutførelsesform er LUT ekstern i forhold til soft-core-prosessoren 52, i hvilket tilfelle det raske minnet er et eksternt minne, slik som DDR2 SRAM.

Selv om de er beskrevet ovenfor som diskrete innretninger, er i en foretrukket utførelsesform av den foreliggende oppfinnelsen tri-FIFO'en 51, soft-core-prosessoren 52 og LUT 59 implementert i en enkelt FPGA.

Igjen med henvisning til fig. 5, mottar tri-FIFO'en 51 data fra et datastreamingsnettverk på en datainngangsport 53. Avhengig av streamingnettverksprotokollen, mottar datainngangsporten 53 også kontrollsignaler, slik som wr (write) og eop (end of packet), slik det enkelt vil forstås av fagfolk. I dette eksemplet skriver datainngangsporten 53 først til buffere 56a. Når buffere 56a er fullt, en bestemt tidsgrense har forløpt, eller på grunn av andre kjente triggere,

overføres buffere til soft-core-prosessorgrensesnittporten 54. Buffer 56c blir samtidig overført til inngangsporten 53 fra utgangsporten 55, og buffere 56 overføres fra grensesnittporten 54 til utgangsporten 55.

5 Ved grensesnittporten 54 sendes data fra buffere (i dette eksemplet 56a) til soft-core-prosessoren 52. Soft-core-prosessoren 52 mottar så pakkene og utfører en initiell verifisering av de mottatte pakker, fulgt av en spekulativ forespørsel til LUT 59 parallelt med fortsettelse av verifisering av de mottatte pakker ved anvendelse av den nye og oppfinneriske fremgangsmåte som er beskrevet nedenfor med henvisning til fig. 6.

10 Fig. 6 viser en fremgangsmåte ifølge den foreliggende oppfinnelsen, hvor fremgangsmåten starter 60 når soft-core-prosessoren mottar en pakke. I et initielt verifiseringstrinn 61 sjekkes lengden av pakken for å finne hvorvidt pakken er tilstrekkelig lang til å inneholde media, f.eks. RTP (Real-time Protocol) pakker som inneholder media. Deretter, dersom pakken finnes å være en sannsynlig kandidat til  
15 å inneholde media, leses headeren for pakken for å beslutte hvorvidt protokollen for pakken er støttet av soft-core-prosessoren. Alternativt, i en eternettutførelsesform, vil en eternettdestinasjonsverifikasjon og eternett CRC (cyclic redundancy check) verifisering også være del av det initielle verifikasjonstrinnet. I tilfelle en hvilken som helst av disse initielle verifikasjoner svikter, returneres pakken til  
20 grensesnittporten 54, og prosessering av neste pakke begynner. Pakken som er returnert til grensesnittporten blir til slutt rutet til vertsfullnettverkstakken 65.

Etter det initielle verifikasjonstrinnet er det neste trinn 62 å finne en strøm-ID (eng.: stream ID) for pakken. Strøm-ID'en finnes ved lesing av pakkeheaderlengden, og i avhengighet av protokollen for pakken, og lese et felt i pakken ved en offset fra  
25 pakkeheaderen, f.eks. i tilfelle av UDP, destinasjons-UDP-portnummer. Selv om det er beskrevet her med henvisning til UDP, idet strøm-ID'en i en pakke alltid er posisjonert innenfor pakken ved en konstant offset til headeren i samsvar med protokollen av pakken, er denne fremgangsmåten med å finne en strøm-ID anvendelig for flere andre protokoller slik det enkelt forstås av fagfolk med  
30 kjennskap til den foreliggende beskrivelsen. Mediestrøm-ID'en som er nyttig for den foreliggende oppfinnelsen innbefatter, men er ikke begrenset til, destinasjonsporten for et UDP-datagram, RDP SCRID, RTP SSRC, H460/18 multipleks-ID, osv.

Deretter, som en forespørsel til en LUT, eller LUT look-up, for at den foreliggende  
35 oppfinnelsen skal virke ved gigabit eternettlinjehastighet, sendes en spekulativ forespørsel 62 som inneholder strøm-ID til LUT 59, mens ytterligere verifikasjon 63 av pakken fortsetter i parallell. Spekulativ i den foreliggende beskrivelsens forstand antar at pakken som korresponderer med strøm-ID sendt til LUT, er en gyldig ukorumpert mediepakke, som er del av en antatt mediestrøm identifisert av strøm-ID'en.  
40

I trinn 62 sendes en forespørsel til LUT 59 ved bruk av den unike mediestrøm-ID som inngangsverdi eller indeksverdi. Dersom mediestrøm-ID'en allerede eksisterer i LUT, mottar soft-core-prosessoren 52 informasjonen, eller metadata, som hører til denne strøm-ID'en fra LUT. Informasjon mottatt fra LUT er strømstatistikk og/eller en korrekt destinasjonsadresse i medieprosesseringsnettverket. Adressen kan være IPv4 destinasjonsadresser, grensesnittindeks (MAC-adresse), kilde-/destinasjons (src/dst)-port eller en hvilken som helst annen nettverksadresse i samsvar med nettverksprotokoll. I det tilfellet at mediastrøm-ID'en ikke er listet i LUT, mottar soft-core-prosessoren melding om at ingen informasjon finnes for denne mediastrøm-ID, eller alternativt mottar den en tom informasjonsmelding.

Mens LUT look-up foregår, fortsetter en ytterligere verifikasjon av pakken korresponderende med mediastrøm-ID'en i trinn 63. Den ytterligere pakkeverifikasjonen, som typisk omfatter ytterligere pakkeformatverifikasjoner, headerformatverifikasjoner, pakkenytteinnholdstype, osv., bestemmer hvorvidt pakken er en gyldig mediapakke eller ikke. I tilfelle at hvilke som helst av disse ytterligere verifikasjonene svikter, returneres pakken til grensesnittporten 54, hvilke som helst data resulterende fra LUT forkastes, og prosessering av den neste pakke begynner. Pakken returnert til grensesnittporten blir til slutt rutet til vertsfullnettverksstakken 65.

I tilfelle av en korumpert pakke, vil look-up i LUT bli basert på falske data tolket som mediastrøm-ID'en. Dersom det finnes en inngang for denne mediastrøm-ID'en, vil LUT returnere gyldige, men ikke relevante data, som deretter forkastes når pakken gjenkjennes som ugyldig av den ytterligere pakkeverifikasjon 63.

I det tilfellet at både den ytterligere pakkeverifikasjonen 63 bestemmer at en pakke er en gyldig mediepakke, og LUT returnerer metadata for en mediestrøm-ID som samsvarer med pakken, vil i trinn 66, soft-core-prosessoren, basert på informasjon mottatt fra LUT 59, på nytt skrive pakkeheaderne for de mottatte mediepakker med korrekt destinasjonsadresse i medieprosesseringsnettverket, og returnere de modifiserte pakker til buffere ved grensesnittporten 54.

Pakker som hører til en mediestrøm med en mediestrøm-ID som ikke er gjenkjent av LUT 59 og/eller pakker besluttet ikke å være gyldige mediepakker, returneres umodifisert til buffere ved grensesnittporten 54. Når buffere ved grensesnittporten 54 (i dette eksemplet 56a) er fullt, en viss tidsgrense har forlatt, eller som følge av andre kjente triggere, overføres buffere til datautgangsporten 55. Alternativt modifiseres headerne ved adressen for vertsnettverkstakken 58.

Datautgangsporten 55 leser ut data fra buffere, (i dette eksemplet 56a), og mediepakker med modifiserte pakkeheadere sendes til medieprosesseringsnettverket direkte til de respektive korrekte destinasjonsadressene i nettverket. Alle andre pakker rutes til vertsnettverksstakken.



Vertsnettverksstakken oppdaterer så LUT 59 med mediestrøm-ID'ene for nye mediepakker og deres respektive destinasjonsadresser i medieprosesseringsnettverket. I avhengighet av medieprosesseringsnettverksprotokollen mottar datautgangsporten 55 også kontrollsignaler, slik som rd (read) og done fra medieprosesseringsnettverket, slik det enkelt forstås av fagfolk.

Fig. 7 viser en eksempelutførelsesform i samsvar med den foreliggende oppfinnelsen, der mediepakker er RTP-pakker transportert over IP/UDP. I trinn 71 blir pakkelengden sjekket for å beslutte hvorvidt pakkelengden er tilstrekkelig til å inneholde RTP-mediepakker, i hvilket tilfelle fremgangsmåten fortsetter til trinn 72. Dersom pakken finnes å være sannsynlig til å inneholde RTP-media, rutes pakken til vertsnettverksstakken, trinn 81, og prosessering av den neste pakken begynner.

I det neste trinnet, 72, leses headeren for pakken for å beslutte hvorvidt pakken er en IP/UDP-pakke. Igjen, dersom pakken ikke er en IP/UDP-pakke, rutes pakken til vertsnettverksstakken, trinn 81, og prosessering av den neste pakken begynner.

I trinn 73 finnes en strøm-ID for pakken ved å lese IP-headerlengden, og IP-headerlengden til offset til destinasjons-UDP-portnummeret. En forespørsel, 74, sendes deretter til LUT 59 ved bruk av destinasjons UDP-portnummeret som inngangsverdi eller indeksverdi. I tilfelle destinasjons-UDP-portnummeret ikke er listet i LUT, sender LUT en melding om at ingen informasjon finnes for dette UDP-portnummeret, eller alternativt sender LUT en tom informasjonsmelding, og pakken rutes til vertsnettverksstakken, trinn 81, og prosessering av den neste pakken begynner.

Dersom UDP-portnummeret allerede eksisterer i LUT, returnerer LUT korrekt destinasjonsadresse i medieprosesseringsnettverket, i dette tilfellet en IPv4-destinasjonsadresse.

Slik det er beskrevet ovenfor med henvisning til fig. 6, mens LUT look-up tar sin tid, blir pakken videre verifisert i trinnene 75-79, og igjen, dersom et hvilket som helst av disse trinnene svikter, rutes pakken til vertsnettverksstakken, trinn 81, og prosessering av den neste pakken begynner.

Testen i trinnene 75-79 er alminnelige pakkeverifikasjonstester kjent av fagfolk, og vil således ikke bli beskrevet i nærmere detalj. Trinn 75 verifiserer IP-pakkesjekksum. Trinn 76 verifiserer IP-headerformat, f.eks. sjekke at IP-versjonnummeret er 4. Trinn 77 verifiserer UDP-pakkeformat. Trinn 78 verifiserer RTP-pakkeformat, og endelig sjekker trinn 79 hvorvidt RTP-flagg og nytteinnholdstype angir at pakken inneholder media.

I trinn 80 blir headerne for pakker som er identifisert som RTP-mediepakker og sendt over IP/UDP, som har de riktige versjoner av protokollene, modifisert med

metadata mottatt fra LUT, og de rutes til de korrekte destinasjonsadresser i medieprosesseringsnettverket.

5 Fremgangsmåten for å filtrere mediepakker fra en videokonferansedatastrøm, og den korresponderende videokonferanseinnretningen, som beskrevet her, kan implementeres i en multipoint control unit (MCU).

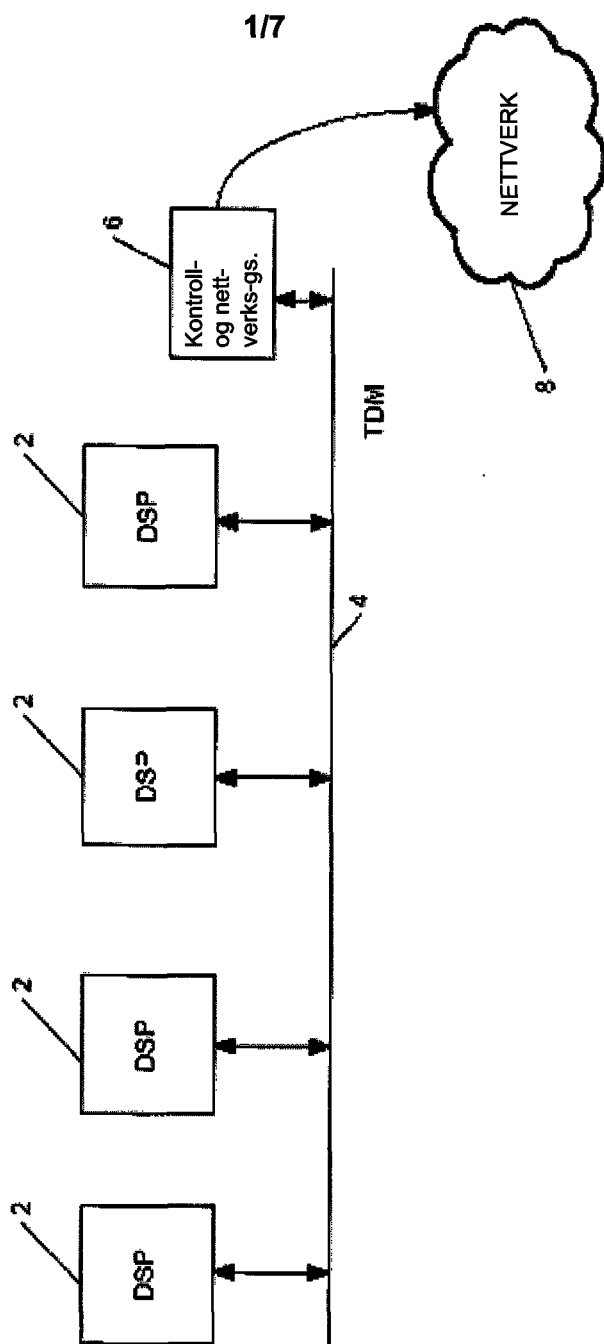
10 Selvsagt vil andre trekk og fordeler fremstå for fagfolk. Den ovenstående systemoversikten representerer noen eksempelutførelsesformer, men andre implementasjoner vil fremstå for fagfolk, og alle slike alternativer er ansett ekvivalente med, og innenfor rekkevidden for den foreliggende oppfinnelsen, som bare er begrenset ved sine krav.

## PATENTKRAV

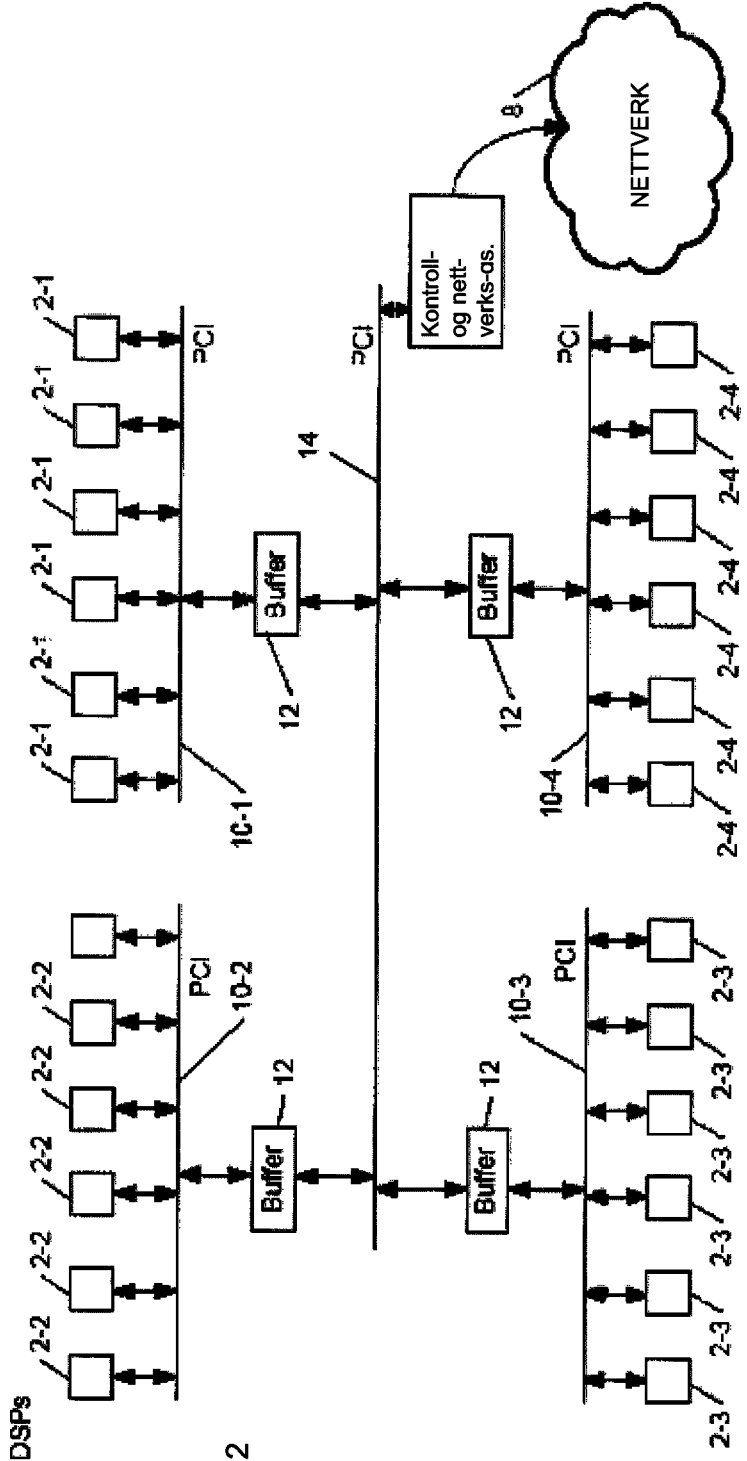
1. Fremgangsmåte for å filtrere mediepakker fra en videokonferansedatastrøm i en videokonferanseinnretning, idet videokonferanseinnretningen omfatter et medieprosesseringsnettverk, et generelt nettverk, en soft-core-prosessor, en Look-up Table memory (LUT) og en full nettverksstakk, idet fremgangsmåten omfatter trinnene:
  - 5 å motta (60), ved soft-core-prosessoren, en pakke i en videokonferansemediestrøm,
  - 10 å bestemme (61) hvorvidt lengden av pakken er tilstrekkelig lang til å inneholde media,
  - å bestemme (61) hvorvidt protokollen av pakken er støttet av soft-core-prosessoren,
  - å finne (62) en mediestrøm-ID for pakken,
  - 15 å sende en forespørsel til LUT ved bruk av mediestrøm-ID'en som en inngangsverdi mens i parallell å bestemme (63) hvorvidt pakken er en gyldig mediepakke, og
  - når en destinasjonsadresse mottas i medieprosesseringsnettverket fra LUT og det besluttes at pakken er en gyldig mediepakke,
  - 20 å modifisere (66) headeren for pakken med destinasjonsadressen mottatt fra LUT, og
  - å rute (65) pakken til den modifiserte destinasjonsadressen.
2. Fremgangsmåte i samsvar med krav 1, hvor fremgangsmåten videre omfatter trinnene:
  - 25 når det bestemmes at pakken er en gyldig mediepakke, og det ikke mottas en destinasjonsadresse i medieprosesseringsnettverket fra LUT,
  - å rute pakken til den fulle nettverksstakken,
  - å gjenvinne en destinasjonsadresse for pakken i medienettverket,
  - å rute pakken til destinasjonsadressen i medienettverket, og
  - 30 å oppdatere LUT med destinasjonsadressen i medienettverket ved bruk av den korresponderende mediestrøm-ID for pakken som indeksverdi.
3. Fremgangsmåte i samsvar med krav 1, hvor mediestrøm-ID'en for en pakke finnes ved å lese et felt i pakken ved en offset fra pakkeheaderen definert ved protokollen for pakken.
4. Fremgangsmåte i samsvar med krav 1, hvor i det tilfelle at pakken ikke er tilstrekkelig lang til å inneholde mediadata, pakken rutes til fullnettverksstakken.
- 35 5. Fremgangsmåte i samsvar med krav 1, hvor i det tilfellet at protokollen av pakken ikke er støttet av soft-core-prosessoren, pakken rutes til fullnettverksstakken.

6. Fremgangsmåte i samsvar med krav 1, hvor i tilfelle at pakken bestemmes ikke å være en gyldig mediepakke, pakken rutes til fullnettverksstakken.
7. Videokonferanseinnretning tilpasset å utføre fremgangsmåten i samsvar med ett av de ovenstående krav, idet videokonferanseinnretningen omfatter  
5 en tri-FIFO (51),  
en soft-core-prosessor (52), og  
en Look-Up Table memory (LUT) (59).
8. Videokonferanseinnretning i samsvar med krav 7, hvor LUT (59) og soft-core-prosessen (52) er implementert i én halvlederinnretning.
- 10 9. Videokonferanseinnretning i samsvar med krav 7, hvor tri-FIFO'en (51), LUT'en (59) og soft-core-prosessen (52) er implementert i én halvlederinnretning.
10. Videokonferanseinnretning i samsvar med krav 8 eller 9, hvor halvlederinnretningen er en Field Programmable Gate Array (FPGA).
11. Videokonferanseinnretning i samsvar med krav 7, hvor tri-FIFO'en (51)  
15 omfatter i det minste tre datapakkebufferne.
12. Videokonferanseinnretning i samsvar med krav 11, hvor tri-FIFO'en (51) omfatter fire datapakkebufferne.
13. Videokonferanseinnretning i samsvar med et av kravene 7-12, implementert i en multipoint control unit (MCU).

TIDLIGERE KJENT  
TEKNIKK

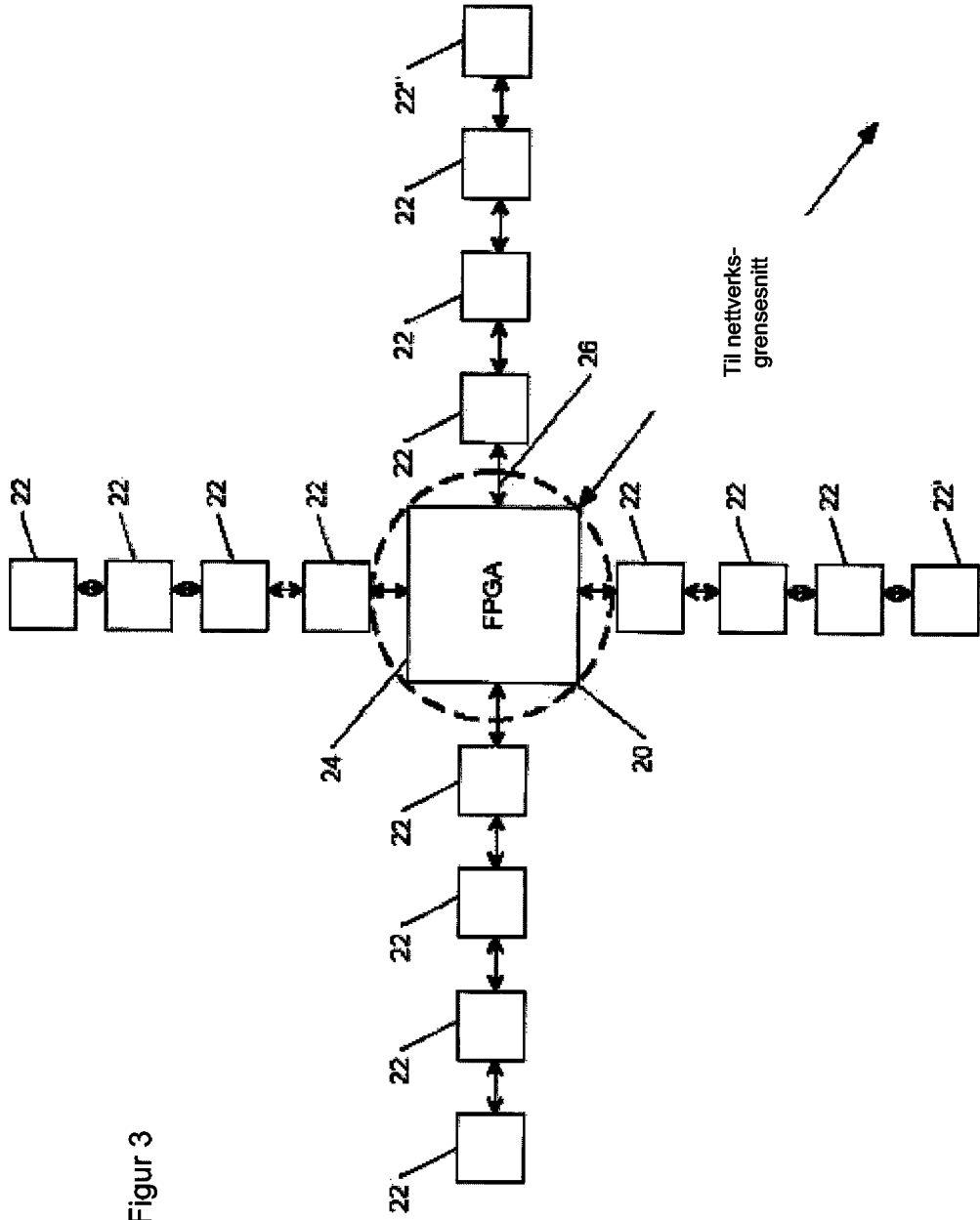


Figur 1



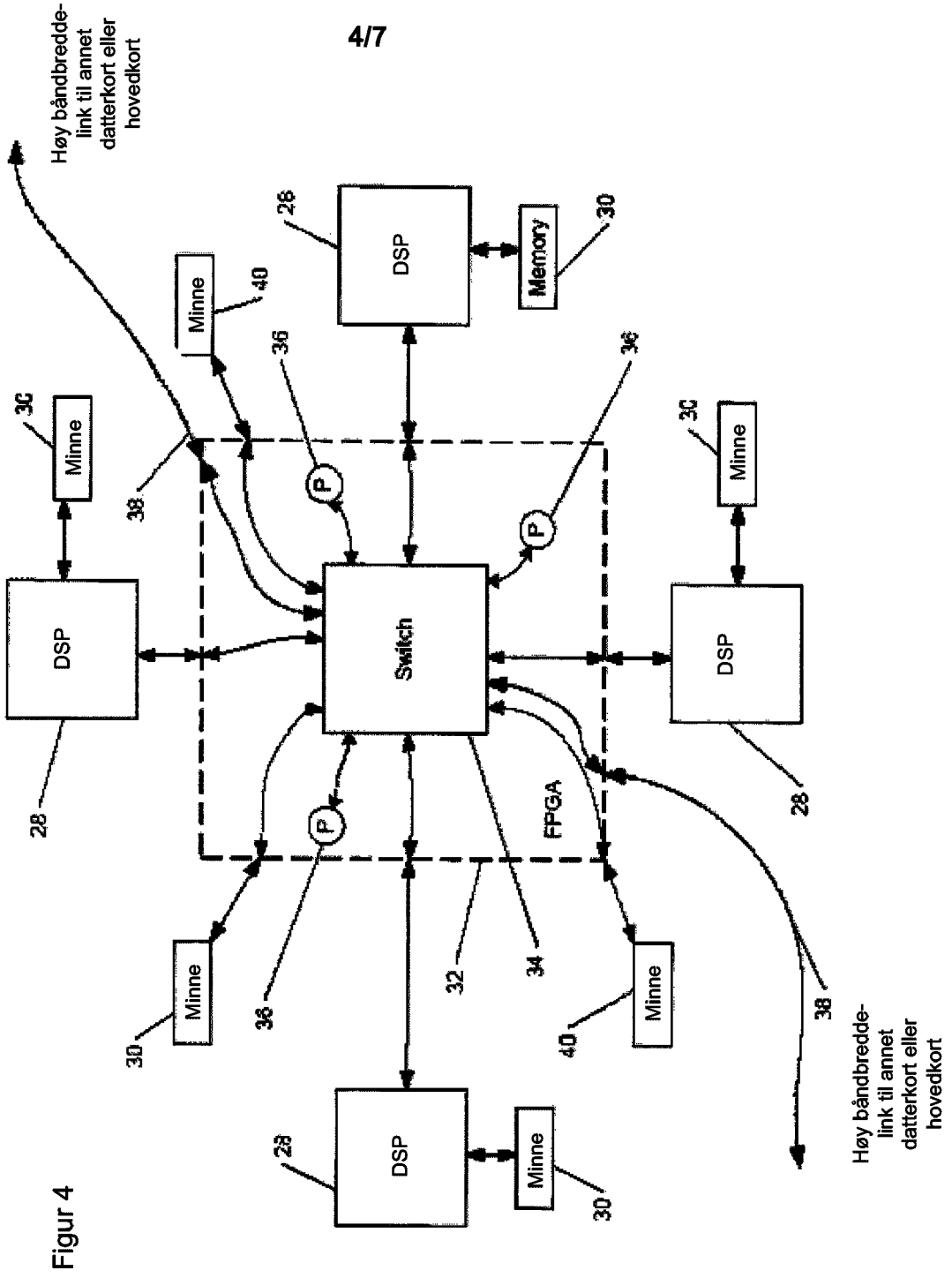
Figur 2

TIDLIGERE KJENT  
TEKNIKK



Figur 3

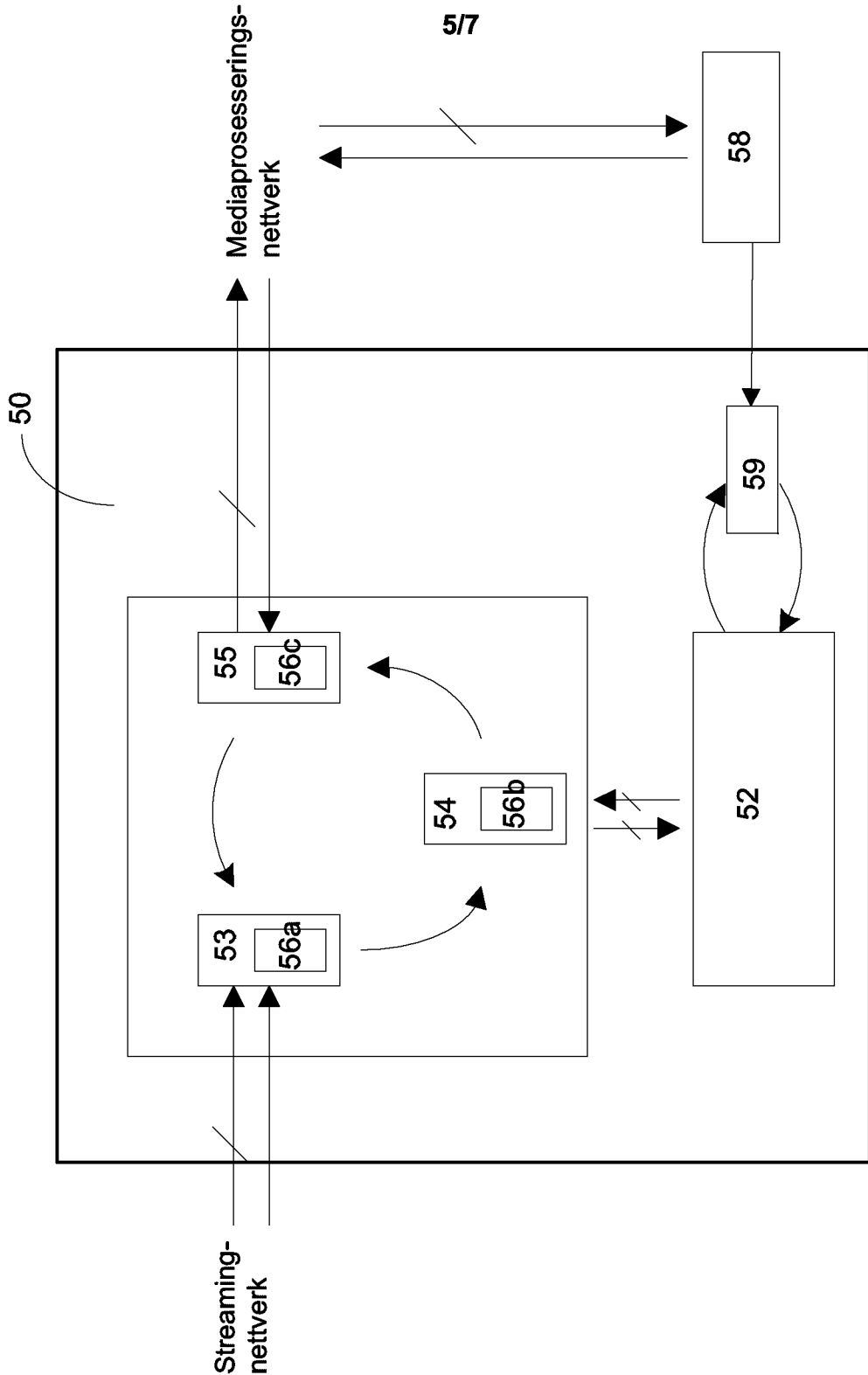
4/7



Figur 4

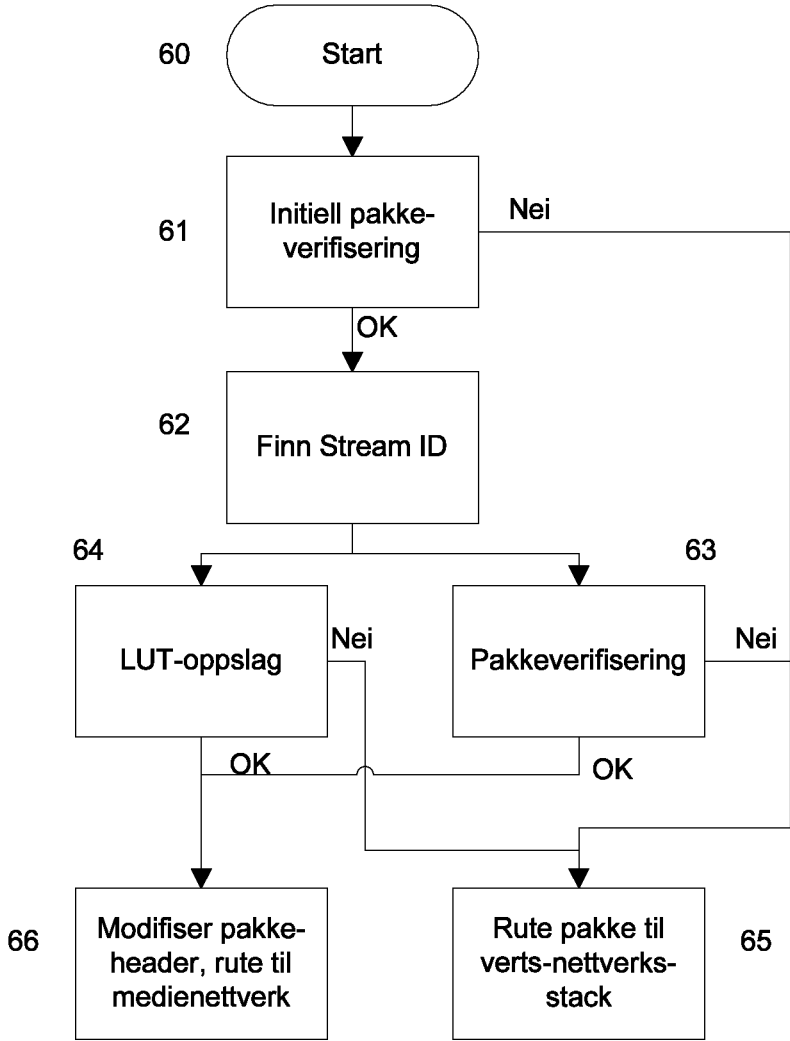
Høy båndbredde-link til annet datterkort eller hovedkort



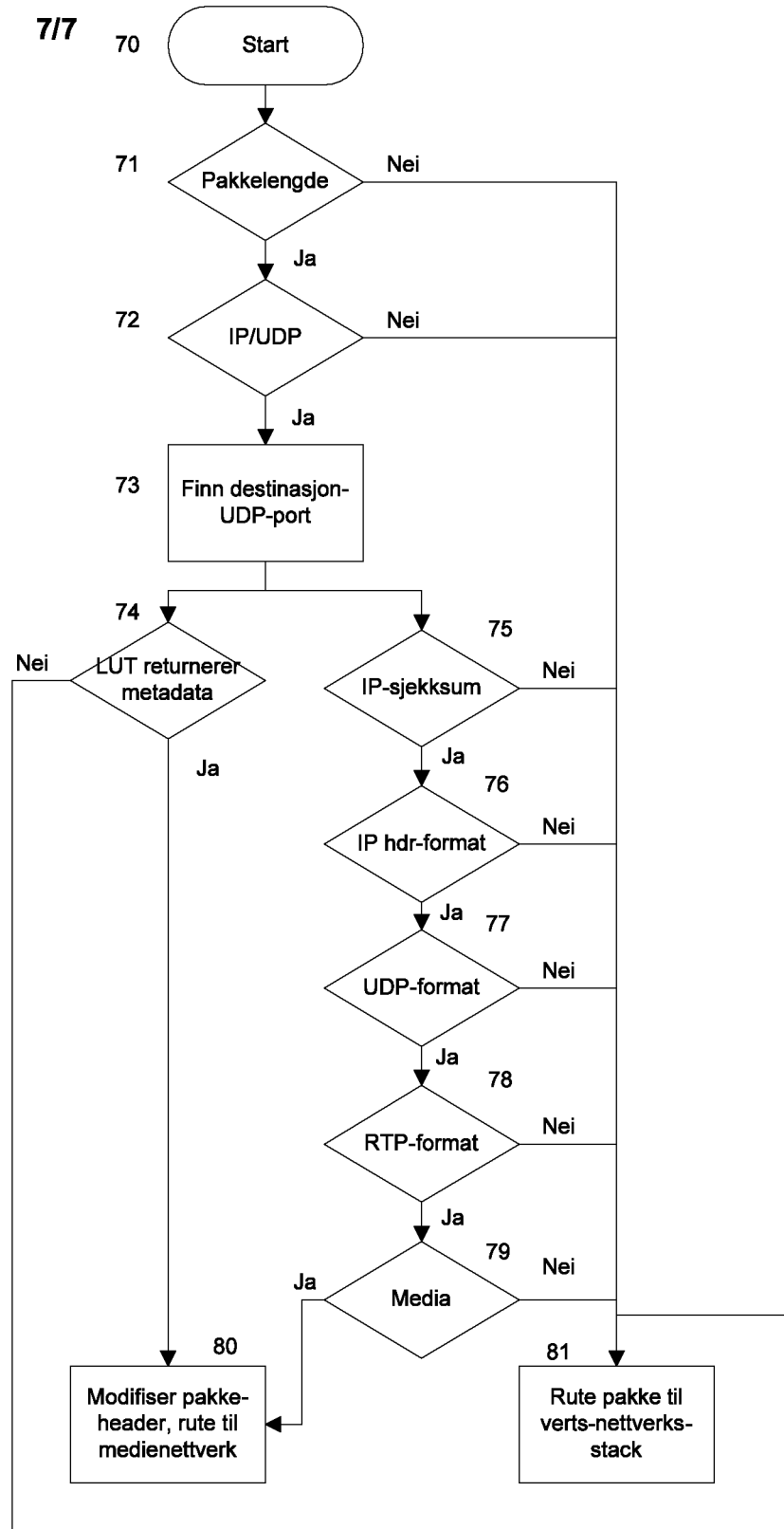


Figur 5

6/7



Figur 6



Figur 7