



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년07월17일
(11) 등록번호 10-0847168
(24) 등록일자 2008년07월11일

(51) Int. Cl.

H04L 12/56 (2006.01) H04L 12/28 (2006.01)

(21) 출원번호 10-2006-0120334
(22) 출원일자 2006년12월01일
심사청구일자 2006년12월01일
(65) 공개번호 10-2008-0049876
(43) 공개일자 2008년06월05일

(56) 선행기술조사문헌
JP11215182 A
(뒷면에 계속)

(73) 특허권자

엘지노텔 주식회사

서울 강남구 역삼동 679 지에스강남타워 7층,8층

(72) 발명자

박성순

경기 안양시 동안구 평촌동 899 향촌롯데아파트
309동 1004호

(74) 대리인

백만기, 주성민

전체 청구항 수 : 총 17 항

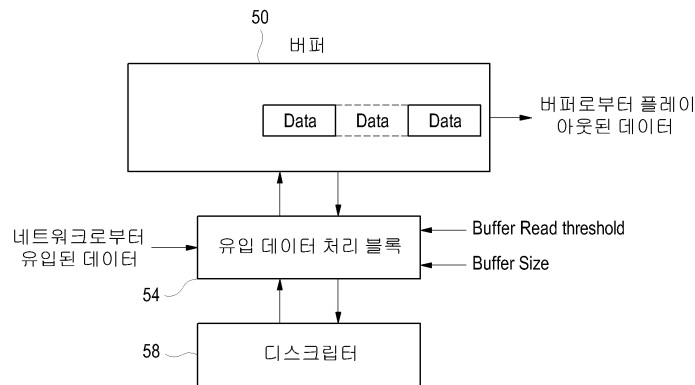
심사관 : 이희봉

(54) UDP/RTP를 사용하는 망에서의 지터 처리를 위한 패킷 순서 재배열 장치 및 방법

(57) 요약

UDP/RTP를 사용하는 망에서의 지터 처리 및 패킷 순서 재배열 장치가 개시된다. 일 실시예에서, 상기 장치는 네트워크로부터 유입되는 데이터 중 제1 기준을 만족하는 데이터를 저장하고, 저장된 데이터 중 제2 기준을 만족하는 데이터를 플레이아웃하는 버퍼, 네트워크로부터 유입되는 데이터가 상기 제1 기준 및 상기 제2 기준을 만족하는지를 판단하여 버퍼에 전달하는 유입 데이터 처리 블록, 및 유입 데이터 처리 블록의 판단의 기준이 되는 파라미터들을 포함하여 유입 데이터 처리 블록에 제공하는 디스크립터 블록을 포함한다. 상기 실시예에 따르면, VoIP와 같은 서비스에 적당하도록 데이터의 전송 품질을 높일 수 있다.

대표도 - 도5



(56) 선행기술조사문헌
JP2003087317 A
KR1020020020955 A
KR1020070053087 A
JP2002305545 A

특허청구의 범위

청구항 1

UDP/RTP(User Datagram Protocol/Real-time Transport Protocol)를 사용하는 네트워크에서의 지터 처리를 위한 패킷 순서 재배열 장치로서,

상기 네트워크로부터 유입되는 데이터 중 제1 기준을 만족하는 데이터를 저장하고, 상기 저장된 데이터 중 제2 기준을 만족하는 데이터를 플레이아웃(play out)하는 버퍼,

상기 유입되는 데이터가 상기 제1 기준 및 상기 제2 기준을 만족하는지를 판단하여 상기 버퍼에 전달하는 유입 데이터 처리 블록, 및

상기 유입 데이터 처리 블록의 상기 판단의 기준이 되는 파라미터들을 포함하여 상기 유입 데이터 처리 블록에 제공하는 디스크립터(descriptor) 블록

을 포함하는 장치.

청구항 2

제1항에 있어서,

상기 네트워크로부터 유입되는 데이터는 RTP 패킷 데이터인 장치.

청구항 3

제1항에 있어서,

상기 파라미터들은

상기 네트워크로부터 유입되는 데이터에 정상적인 연결이 설정되어 있는지를 나타내는 V 파라미터,

상기 버퍼를 플레이아웃할 준비가 되었는지를 나타내는 F 파라미터,

연결당 저장된 데이터의 수를 나타내는 PacketNo 파라미터,

상기 버퍼에 저장된 데이터의 위치를 나타내는 BITMAP 파라미터,

상기 BITMAP 파라미터에서 가장 빨리 읽어야 할 위치를 나타내는 BaseBITMAP 파라미터,

상기 BaseBITMAP 파라미터의 '1'의 위치를 이진수로 나타내는 BaseBITMAP_D 파라미터,

상기 네트워크로부터 유입되는 데이터를 받을 것인지 폐기할 것인지를 결정하는 기준이 되는 순서 번호를 나타내는 BaseSN 파라미터,

첫번째로 플레이아웃된 데이터의 순서 번호를 나타내는 FirstRxSN 파라미터, 및

소정의 시간 단위로 상기 디스크립터 블록을 몇 번 읽었는지를 나타내는 RoundCount 파라미터

를 포함하는 장치.

청구항 4

제1항에 있어서,

상기 파라미터들은 상기 유입 데이터 처리 블록에 의해 업데이트되는 장치.

청구항 5

제1항에 있어서,

상기 버퍼의 사이즈와 상기 버퍼의 플레이아웃 문턱값은 사용자에게 의해 결정되는 장치.

청구항 6

제1항에 있어서,

상기 제1 기준은 상기 네트워크로부터 유입되는 데이터에 정상적인 연결이 설정되어 있는지 여부를 나타내는 조건인 장치.

청구항 7

제1항에 있어서,

상기 제2 기준은 상기 버퍼에 저장된 데이터의 개수가 상기 버퍼의 플레이아웃 문턱값과 같은지 여부를 나타내는 조건인 장치.

청구항 8

제3항에 있어서,

상기 제1 기준은 상기 V 파라미터에 기초하여 결정되는 장치.

청구항 9

제3항에 있어서,

상기 제2 기준은 상기 F 파라미터와 상기 RoundCount 파라미터에 기초하여 결정되는 장치.

청구항 10

UDP/RTP를 사용하는 네트워크에서의 지터 처리를 위한 패킷 순서 재배열 방법으로서,

유입 데이터 처리 블록이 디스크립터를 판독하여 상기 네트워크로부터 유입된 데이터의 연결이 유효한지 확인하는 단계,

상기 유입된 데이터의 연결이 유효하면, 상기 유입 데이터 처리 블록이 버퍼에 저장된 데이터의 개수가 허용 범위 내인지를 확인하는 단계,

상기 버퍼에 저장된 데이터가 허용 범위 내이면, 상기 유입 데이터 처리 블록이 상기 유입된 데이터의 수를 상기 버퍼의 플레이아웃 문턱값과 비교하는 단계,

상기 비교 결과에 따라 상기 유입 데이터 처리 블록이 상기 디스크립터를 업데이트하는 단계, 및

상기 유입된 데이터의 수가 상기 버퍼의 상기 플레이아웃 문턱값과 같으면, 상기 유입된 데이터의 타임스탬프 값에 따라 상기 버퍼로부터 데이터를 플레이아웃하고, 그에 따라 상기 디스크립터를 업데이트하는 단계

를 포함하는 방법.

청구항 11

제10항에 있어서,

상기 버퍼의 사이즈와 상기 버퍼의 상기 플레이아웃 문턱값은 사용자에게 의해 결정되는 방법.

청구항 12

제10항에 있어서,

상기 유입된 데이터의 연결이 유효하지 않으면, 상기 유입 데이터 처리 블록이 상기 유입된 데이터를 폐기하는 단계를 더 포함하는 방법.

청구항 13

제10항에 있어서,

상기 버퍼에 저장된 데이터의 개수가 허용 범위를 벗어나면, 상기 유입 데이터 처리 블록이 상기 유입된 데이터를 폐기하는 단계를 더 포함하는 방법.

청구항 14

제10항에 있어서,

상기 유입된 데이터의 수가 상기 버퍼의 상기 플레이아웃 문턱값과 같아지면, 상기 유입 데이터 처리 블록은 소정의 기준 순서 번호보다 작은 순서 번호를 가지고 유입되는 데이터를 폐기하는 단계를 더 포함하는 방법.

청구항 15

제10항에 있어서,

상기 버퍼에 저장된 데이터의 개수가 상기 버퍼의 사이즈와 같으면, 상기 버퍼에 저장된 데이터를 2개 플레이아웃하고, 그에 따라 상기 디스크립터를 업데이트하는 단계를 더 포함하는 방법.

청구항 16

제10항에 있어서,

상기 버퍼에 저장된 데이터의 개수가 1이면, 해당 데이터는 플레이아웃하고, 해당 디스크립터에서 연결의 유효 여부를 나타내는 파라미터를 제외한 모든 파라미터들을 초기화하는 단계를 더 포함하는 방법.

청구항 17

제10항 내지 제16항 중의 어느 하나의 방법을 실행하는 컴퓨터 수행가능 명령어들을 저장하는 컴퓨터 판독 가능 저장매체.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <11> 본 발명은 UDP/RTP(User Datagram Protocol/Real-time Transport Protocol)를 사용하여 음성, 영상 및 데이터를 전달하는 망에서 발생하는 지터(jitter)의 처리 및 패킷 순서 재배열에 관한 것이다.
- <12> UDP(User Datagram Protocol)는 인터넷 상에서 서로 정보를 주고받을 때 정보를 보낸다는 신호나 받는다는 신호의 송수신 절차를 거치지 않고 보내는 쪽에서 일방적으로 데이터를 전달하는 통신 프로토콜이다. TCP/IP(Transmission Control Protocol/Internet Protocol)에서는 제3 계층에 해당하는 망 계층 프로토콜인 IP와 제4 계층에 해당하는 전송 계층 프로토콜인 TCP 또는 UDP의 어느 하나를 조합하여 데이터를 주고받는데, TCP에서는 세션을 설정한 후에 통신을 개시하지만 UDP에서는 세션을 설정하지 않고 데이터를 상대의 주소로 송출하고 데이터의 에러 복구 등의 절차를 수행하지 않기 때문에, UDP는 TCP보다 신뢰성은 떨어지지만 프로토콜 처리 속도는 훨씬 빠르다. UDP는 IP 서비스에 호스트 대 호스트 통신 대신에 프로세스 대 프로세스 통신을 추가로 제공하고 있으며, 도 1에 도시된 바와 같이 데이터 패킷의 데이터 필드에 최소한의 오버헤드만 추가하여 사용하는 매우 간단한 프로토콜이다. 일반적으로, 만약 프로세스가 작은 메시지를 보내고자 하고 신뢰성에 대해 걱정하지 않는다면, 작은 메시지를 보내기 위해서는 TCP보다 UDP를 사용한다.
- <13> 한편, 음성 데이터를 인터넷 프로토콜(IP) 데이터 패킷으로 변화하여 일반 전화망에서의 통화를 가능하게 해주는 통신 서비스 기술인 VoIP(Voice over Internet Protocol)와 같은 실시간 데이터 서비스에는 UDP에 기초한 RTP/RTCP(Real-time Transport Protocol/RTP Control Protocol)가 널리 이용되고 있다. 도 2는 기본적인 VoIP 망에서의 음성 데이터의 흐름도를 도시하고 있으며, 이는 UDP/RTP를 이용하는 영상 및 데이터를 처리하는 실시간 데이터 서비스에도 적용될 수 있다. 도 2를 참조하면, 음성이 음성 입력 장치에서 아날로그 신호로 변환된 후, 코덱(CODEC)에서 디지털화된다. 디지털화된 음성은 선택된 적절한 압축 방식에 의해 압축되고, 압축시 목음 제거 기능을 선택할 경우 음성 내 목음을 제거하여 발생 트래픽을 줄일 수 있다. 압축된 음성 패킷은 RTP 패킷화되어 인터넷을 통해 전달되게 되는데, RTP 패킷은 도 3에 도시된 바와 같이 각 패킷의 발생 시간을 의미하는 Timestamp 필드가 있어 수신단에서 음성의 플레이아웃(play out) 시간의 동기화가 가능하다. RTP 패킷은 인터넷을 거치면서 전달 지연, 지연 변이, 패킷 손실 등이 발생할 수 있다. 전달 지연은 절대적인 인터넷 통과 지연을 의미하며, 지연 변이, 즉 지터(jitter)는 최대 전달 지연과 최소 전달 지연의 차이로 정의된다. 매 패킷마다 전달 지연이 다르므로 지연 변이가 발생하게 되는 것이다. 이러한 지연 변이는 음성 품질에 악영향을

미치므로, 수신단에 버퍼(buffer)를 사용하여 RTP 패킷들 사이의 지연 변이를 흡수한다. 즉, 인터넷을 통해 전달된 RTP 패킷들을 버퍼에 저장한 후 RTP 패킷 내의 타임스탬프(timestamp) 값에 맞추어 시간에 맞게 플레이아웃하면, 지터가 없는 데이터가 출력되게 된다.

<14> 도 4는 도 2의 흐름도의 수신단에 위치하는 버퍼의 구조를 도시하고 있다. 도 4에 도시된 바와 같이, 인터넷으로부터 유입된 패킷은 먼저 버퍼(40)에 저장되고, 이 저장된 버퍼의 어드레스(address) 또는 위치 정보(location)와 유입된 패킷의 RTP 순서 번호(Sequence Number: SN)가 해당 연결에 할당된 버퍼 제어 메모리(44)에 저장된다. RTP 패킷의 타임스탬프 값에 따라 버퍼에 저장된 패킷을 판독할 시간이 되면, 버퍼 제어 메모리에 저장된 모든 위치 정보 중에서 가장 작은 순서 번호를 가진 어드레스 값이 선택되어, 해당하는 버퍼에 저장된 패킷이 플레이아웃된다. 상기 버퍼 판독(read) 과정이 완료되면, 그 위치 정보는 클리어(clear)되고, 해당 버퍼는 다음에 유입되는 패킷을 위한 준비 상태로 전환된다.

<15> 그러나, 버퍼에 저장된 데이터를 플레이아웃하는 시점을 RTP의 타임스탬프 값에만 의존하게 되면, 버퍼 판독 클럭(clock)의 정밀도(accuracy)가 떨어지는 경우에 타임스탬프 값에 따른 정확한 버퍼 플레이아웃을 수행할 수 없다. 즉, 만약 계속해서 조금씩 지연이 발생할 경우, 언젠가 하나의 패킷이 지연되는 경우가 발생할 수 있다. 또한, 이미 버퍼에서 플레이아웃된 순서 번호보다 낮은 순서 번호의 패킷이 유입될 경우, 불필요한 처리를 한 번 더 하게 되어 시스템 성능에 영향을 미칠 수 있으며, 버퍼와 버퍼 제어 메모리의 판독과 기록이 동시에(synchronous) 일어나기 때문에 패킷 처리 성능에 문제가 발생할 수 있다.

발명이 이루고자 하는 기술적 과제

<16> 본 발명은 상기 종래 기술의 문제점을 해결하기 위한 것으로, UDP/RTP를 사용하는 망에서 지연 변이를 줄이도록 동적으로 수신단의 버퍼를 동작시키고, 네트워크를 통해 전송된 RTP 패킷의 순서를 재배열함으로써, VoIP와 같은 실시간 데이터 전송 서비스에 적당하도록 데이터의 전송 품질을 높이는 것을 목적으로 한다.

<17> 일 실시예에 따르면, UDP/RTP(User Datagram Protocol/Real-time Transport Protocol)를 사용하는 네트워크에서의 지터 처리 및 패킷 순서 재배열 장치가 제공된다. 상기 장치는 상기 네트워크로부터 유입되는 데이터 중 제1 기준을 만족하는 데이터를 저장하고, 상기 저장된 데이터 중 제2 기준을 만족하는 데이터를 플레이아웃하는 버퍼, 상기 유입되는 데이터가 상기 제1 기준 및 상기 제2 기준을 만족하는지를 판단하여 상기 버퍼에 전달하는 유입 데이터 처리 블록, 및 상기 유입 데이터 처리 블록의 상기 판단의 기준이 되는 파라미터들을 포함하여 상기 유입 데이터 처리 블록에 제공하는 디스크립터(descriptor) 블록을 포함한다.

<18> 다른 실시예에 따르면, UDP/RTP를 사용하는 네트워크에서의 지터 처리 및 패킷 순서 재배열 방법이 제공된다. 상기 방법에 따르면, 유입 데이터 처리 블록이 디스크립터를 판독하여 상기 네트워크로부터 유입된 데이터의 연결이 유효한지 확인하고, 상기 유입된 데이터의 연결이 유효하면, 버퍼에 저장된 데이터의 개수가 허용 범위 내 인지를 확인하며, 상기 버퍼에 저장된 데이터가 허용 범위 내이면, 상기 유입된 데이터의 수를 상기 버퍼의 플레이아웃 문턱값과 비교한다. 상기 유입 데이터 처리 블록은 상기 비교 결과에 따라 상기 디스크립터를 업데이트하고, 상기 유입된 데이터의 수가 상기 버퍼의 상기 플레이아웃 문턱값과 같으면, 상기 유입된 데이터의 타임스탬프 값에 따라 상기 버퍼로부터 데이터를 플레이아웃하고, 그에 따라 상기 디스크립터를 업데이트한다.

발명의 구성 및 작용

<19> 본 발명은 UDP/RTP를 사용하는 망에서의 지터 처리 및 패킷 순서 재배열 장치 및 방법을 제공한다. 이하, 첨부 도면을 참조하여, 본 발명의 실시예에 대해 상세히 설명한다.

<20> 도 5는 수신단의 버퍼의 구조의 일 실시예를 도시하고 있다. 도 5에 도시된 바와 같이, 인터넷과 같은 네트워크로부터 유입되는 데이터는 유입 데이터 처리 블록(54)을 통해 사용자 기준에 적합한 데이터만, IP 헤더가 제거된 데이터 형태로 데이터 저장용 버퍼(50)에 저장된다. 유입 데이터 처리 블록(54)을 통하여 유입되는 데이터를 받을 것인지 버릴 것인지 등의 결정은 디스크립터 블록(descriptor)(58)의 내용을 판독함으로써 이루어진다. 도 6에 도시된 바와 같이, 디스크립터는 V, F, PacketNo, BaseBITMAP_D, BaseBITMAP, BITMAP, BaseSN, FirstRxSN, RoundCount 필드를 포함한다. V는 유효 비트(Valid bit)로서 네트워크로부터 유입된 데이터에 정상적인 연결이 설정되어 있는지를 알려주는 비트이며, V 필드가 '0'으로 설정되어 있으면 유입되는 데이터는 폐기된다. F 필드는 플래그(Flag) 비트로서 버퍼에 저장된 데이터의 개수가 사용자에 의해 결정된 버퍼 플레이아웃 문턱값(threshold)과 일치하면 '1'로 설정되며, 이는 버퍼를 읽을 준비가 되었음을 의미한다. PacketNo 필드는 연결당 저장된 패킷의 수를 나타낸다. BITMAP 필드는 버퍼에 저장된 데이터의 위치를 나타내는 파라미터로서,

데이터가 버퍼에 저장되면 해당 위치의 비트가 '1'로 설정되고, BITMAP 필드가 '0'이면 저장된 패킷이 없음을 의미한다. BaseBITMAP 필드는 BITMAP 필드에서 가장 빨리 읽어야 할 위치를 알려주는 파라미터로서, 이 파라미터에 있는 '1'의 위치와 같거나 왼쪽으로 가장 가까이 있는 '1'의 위치가 다음에 읽혀질 버퍼로 결정된다. BaseBITMAP_D 필드는 BaseBITMAP 필드에 있는 '1'의 위치를 이진수로 나타낸 값으로서, 0번 비트가 1이면 '0', 1번 비트가 1이면 '1', 2번 비트가 1이면 '10'이 된다. BaseSN 필드는 유입되는 데이터를 받을 것인지 버릴 것인지를 결정하는 기준이 되는 순서 번호(sequence number)를 나타내는 파라미터로서, BaseSN 필드 보다 작은 값이 유입되면 폐기된다. FirstRxSN 필드는 첫번째로 플레이아웃된 데이터의 순서 번호를 나타내는 파라미터이다. RoundCount 필드는 소정의 시간 단위, 예컨대 5ms 단위로 디스크립터를 몇 번 읽었는지를 나타내는 파라미터로서, 상기 소정의 시간 단위는 사용자에게 의해 결정될 수 있다. 버퍼에 저장된 데이터를 플레이아웃하는 시점은 RoundCount 필드값과 F 비트에 따라 결정된다. 예컨대 AMR 코덱(Adaptive Multi Rate CODEC)의 경우는 타임스탬프가 20ms이므로, 5ms 단위로 디스크립터를 읽을 경우, F 비트가 '1', RoundCount 필드값이 '3'이 되면 버퍼에 저장된 데이터가 플레이아웃된다.

<21> 다시 도 5를 참조하면, 유입 데이터 처리 블록(54)이 정상 동작을 하기 전에 버퍼의 사이즈와 버퍼 플레이아웃 문턱값을 결정하여야 한다. 이 두 값들은 최적의 성능을 얻기 위해 사용자에게 의해 결정될 수 있다. 유입 데이터 처리 블록(54)은 디스크립터 블록(58)과 정확한 동기를 가지며 동작을 한다. 인터넷과 같은 네트워크로부터 데이터가 유입되면, 유입 데이터 처리 블록(54)은 해당 연결의 디스크립터의 V 비트 값을 확인하여, 그 값이 '0'이면 데이터를 폐기하고, '1'이면 데이터를 버퍼(50)에 저장하고 디스크립터 블록(58)의 내용을 업데이트한다. 또한, 저장된 패킷의 개수가 데이터를 플레이아웃하기 위한 문턱값과 같아지면, 디스크립터의 F 비트를 '1'로 설정하고 버퍼(50)를 관독할 시간이 되었음을 알린다. 버퍼(50)를 관독하게 되면, 유입 데이터 처리 블록(54)은 그에 따라 디스크립터의 내용을 업데이트한다.

<22> 도 7은 네트워크로부터 유입되는 데이터의 처리 과정의 일 실시예를 도시하고 있다. 이하에서는, 디스크립터의 동작의 이해를 돕기 위해, 네트워크로부터 유입되는 데이터는 타임스탬프가 20ms인 음성 데이터이고, 버퍼는 최대 120ms(±60ms)의 지연 변이를 수용할 수 있으며, 버퍼에 저장된 데이터를 플레이아웃하기 위한 버퍼 플레이아웃 문턱값은 3이라고 가정한다. 이러한 가정 하에서는 24비트의 BITMAP 필드와 BaseBITMAP 필드는 하위 6비트(5:0)만 사용되고, BaseBITMAP_D 필드는 0에서 5까지만 카운터된다.

<23> 도 7을 참조하면, 네트워크로부터 데이터가 유입되면, 유입 데이터 처리 블록은 먼저 디스크립터를 읽어 유입된 데이터의 연결이 유효한지 확인하여(s71), 유효하지 않으면 폐기하고, 유효하면 버퍼에 저장된 데이터의 개수가 허용 범위 내인지를 확인한다(s72). 버퍼에 저장된 데이터의 개수가 허용 범위를 벗어나면, 유입된 데이터를 폐기하며, 허용 범위 내이면 다음 단계로 전달한다.

<24> 이후, 유입 데이터 처리 블록은 사용자에게 의해 설정된 플레이아웃 문턱값과 디스크립터의 PacketNo 필드값을 비교한다(s73). 먼저, PacketNo 필드값이 문턱값보다 작고 PacketNo 필드값이 '0'인 경우, 즉 유입된 데이터가 첫번째 데이터인 경우, PacketNo 필드값이 문턱값보다 작으므로 Flag는 '0'이 되며, 유입된 데이터는 버퍼의 제일 가운데에 저장되고, 그 위치를 알려주는 BITMAP 필드와 BaseBITMAP 필드는 "000100"이 되며, '1'이 2번 비트에 위치하므로 BaseBITMAP_D 필드는 "00010"으로 업데이트된다. 또한, 다음에 유입될 데이터의 순서 번호(SN)와 비교될 BaseSN/FirstRxSN 필드는 유입된 데이터의 순서 번호로 업데이트되며, 버퍼의 관독과 기록 사이클이 완전히 분리되어 있으므로 RoundCount 필드는 디스크립터에서 읽은 값 그대로 사용된다. 업데이트 결과는 다음과 같다 (예 1 참조).

<25> <예 1>

- <26> Flag <= '0';
- <27> PacketNo <= '1';
- <28> BaseBITMAP_D <= "00010";
- <29> BaseBITMAP <= "000100";
- <30> BITMAP <= "000100";
- <31> BaseSN <= RTP Sequence Number of IncomingData;
- <32> FirstRxSN <= RTP Sequence Number of IncomingData;

- <33> RoundCount <= RoundCount of Descriptor;
- <34> PacketNo 필드값이 문턱값보다 작고 PacketNo 필드값이 '1'이며 순서 번호(SN)가 BaseSN 필드보다 1이 큰 데이터가 유입되는 경우에는, PacketNo 필드값이 문턱값보다 작으므로 Flag는 '0'이 되고, PacketNo 필드값은 1 증가되어 '2'가 된다. 새롭게 유입된 데이터의 SN이 BaseSN 필드보다 크므로 BaseBITMAP 필드와 BaseBITMAP_D 필드는 기존의 값을 유지하고, 유입된 데이터의 SN이 BaseSN 필드보다 1이 크므로 새로운 BITMAP 필드는 기존의 BITMAP 필드와 BaseBITMAP 필드를 왼쪽으로 1만큼 시프트(shift)시킨 값과의 OR 연산된 값으로 업데이트된다. 업데이트 결과는 다음과 같다 (예 2 참조).
- <35> <예 2>
- <36> Flag <= '0';
- <37> PacketNo <= '2';
- <38> BaseBITMAP_D <= "00010";
- <39> BaseBITMAP <= "000100";
- <40> BITMAP <= "001100";
- <41> BaseSN <= FirstRxSN of Descriptor;
- <42> FirstRxSN <= FirstRxSN of Descriptor;
- <43> RoundCount <= RoundCount of Descriptor;
- <44> PacketNo 필드값이 문턱값보다 작고 PacketNo 필드값이 '1'이며 순서 번호(SN)가 BaseSN 필드보다 1이 작은 데이터가 유입되는 경우에는, Flag는 '0'이 되고, PacketNo 필드값은 '2'가 된다. 새롭게 유입된 데이터의 SN이 BaseSN 필드보다 작으므로, BaseBITMAP 필드는 새롭게 유입된 데이터의 위치 "000010"으로 변경이 되고, '1'의 위치가 1번 비트이므로 BaseBITMAP_D 필드는 "00001"로 업데이트된다. 유입된 데이터의 SN이 BaseSN 필드보다 1이 작으므로 새로운 BITMAP 필드는 기존의 BITMAP 필드와 BaseBITMAP 필드를 오른쪽으로 1만큼 시프트시킨 값과의 OR 연산된 값으로 업데이트된다. 업데이트 결과는 다음과 같다 (예 3 참조).
- <45> <예 3>
- <46> Flag <= '0';
- <47> PacketNo <= '2';
- <48> BaseBITMAP_D <= "00001";
- <49> BaseBITMAP <= "000010";
- <50> BITMAP <= "000110";
- <51> BaseSN <= RTP Sequence Number of IncomingData;
- <52> FirstRxSN <= RTP Sequence Number of IncomingData;
- <53> RoundCount <= RoundCount of Descriptor;
- <54> PacketNo 필드값이 '2'인 상태에서 새로운 데이터가 유입되면, 앞서 유입된 2개의 데이터 SN과 비교하여 제일 작은 SN을 가지는 데이터가 BaseSN/FirstRxSN 필드가 되고, 버퍼 관독이 가능함을 알려주는 Flag 비트는 '1'로 설정된다. 만약 순차적인 SN을 가지는 데이터가 유입되면, 업데이트 결과는 다음과 같다 (예 4 참조).
- <55> <예 4>
- <56> Flag <= '1';
- <57> PacketNo <= '3';
- <58> BaseBITMAP_D <= "00010";
- <59> BaseBITMAP <= "000100";

- <60> BITMAP <= "011100";
- <61> BaseSN <= FirstRxSN of Descriptor;
- <62> FirstRxSN <= FirstRxSN of Descriptor;
- <63> RoundCount <= RoundCount of Descriptor;
- <64> 일단 Flag 비트가 설정이 되면, BaseSN 필드보다 작은 SN을 가지고 유입되는 데이터는 폐기된다. 데이터가 저장된 위치를 알려주는 BITMAP 필드는 2번 비트를 기준으로 하여 왼쪽으로 비트가 설정되고, 마지막 5번 비트가 설정되면 다음에 유입되는 데이터는 0번 비트의 위치로 가는 방식으로 동작된다.
- <65> 버퍼의 관독 플레이아웃 시점도 알려주는 디스크립터는 연결별로 5ms마다 액세스되고, 한 번 액세스되면 RoundCount 필드가 1씩 증가된다. RoundCount 필드값이 3이 되고, Flag 비트가 '1'이 되면(s74), BaseBITMAP 필드의 위치와 같거나 왼쪽으로 가장 가까이 있는 데이터의 위치부터 관독되고, 그 차이만큼 BaseSN, BaseBITMAP, BaseBITMAP_D 필드값이 업데이트되며, RoundCount 필드는 다시 초기화된다. 만약 RoundCount 필드값이 3이고 디스크립터에서 읽은 파라미터가 위의 <예 4>의 업데이트 내용과 같을 경우, BaseBITMAP 필드의 2번 비트와 BITMAP 필드의 2번 비트가 같은 자리에 위치하므로, 2번 비트가 지시하는 버퍼의 데이터를 플레이아웃하고, 업데이트 결과는 다음과 같다 (예 5 참조).
- <66> <예 5>
- <67> Flag <= '1';
- <68> PacketNo <= '2';
- <69> BaseBITMAP_D <= "00011";
- <70> BaseBITMAP <= "001000";
- <71> BITMAP <= "011000";
- <72> BaseSN <= FirstRxSN of Descriptor + 1;
- <73> FirstRxSN <= FirstRxSN of Descriptor;
- <74> RoundCount <= 0;
- <75> RoundCount 필드값이 3이고, 디스크립터에서 읽은 파라미터가 아래의 <예 6>과 같은 경우,
- <76> <예 6>
- <77> Flag <= '1';
- <78> PacketNo <= '3';
- <79> BaseBITMAP_D <= "00010";
- <80> BaseBITMAP <= "000100";
- <81> BITMAP <= "111000";
- <82> BaseSN <= FirstRxSN of Descriptor;
- <83> FirstRxSN <= FirstRxSN of Descriptor;
- <84> RoundCount <= 3;
- <85> BaseBITMAP 필드의 2번 비트와 왼쪽으로 가장 가까이 있는 비트는 BITMAP 필드의 3번 비트이므로, 업데이트 결과는 다음과 같다 (예 7 참조).
- <86> <예 7>
- <87> Flag <= '1';
- <88> PacketNo <= '2';

- <89> BaseBITMAP_D <= "00100";
- <90> BaseBITMAP <= "010000";
- <91> BITMAP <= "110000";
- <92> BaseSN <= FirstRxSN of Descriptor + 2;
- <93> FirstRxSN <= FirstRxSN of Descriptor;
- <94> RoundCount <= 0;

<95> 버퍼를 동적으로 제어하기 위하여, 버퍼가 가득 찬 경우(s75)에는 버퍼에 저장된 데이터를 1개가 아닌 2개를 읽어 다음에 유입될 데이터를 위한 공간을 마련할 수 있다. 버퍼에 저장된 데이터의 개수가 2 이상인 경우(s76)에는 단계(s74)와 동일한 과정으로 처리되고, 버퍼에 저장된 데이터의 개수가 1인 경우(s77)에는 해당되는 데이터는 판독하고 해당 디스크립터에 저장된 파라미터 중 V 비트를 제외한 모든 파라미터를 초기화하여 새로 데이터가 유입될 경우에는 단계(s73)부터 다시 시작한다.

<96> 본 발명 및 그 다양한 기능적 구성요소들은 특정 실시예들로 설명되었으나, 본 발명은 하드웨어, 소프트웨어, 펌웨어, 미들웨어 또는 이들의 조합으로 구현될 수 있으며, 시스템, 서브시스템, 구성요소들 또는 이들의 서브 구성요소들로 활용될 수 있음을 이해해야 한다. 소프트웨어로 구현되는 경우, 본 발명의 요소들은 필요한 작업들을 수행하기 위한 명령어들/코드 세그먼트들이다. 프로그램 또는 코드 세그먼트들은 프로세서 판독 가능 매체와 같은 머신 판독 가능 매체, 컴퓨터 프로그램 제품 내에 저장될 수 있으며, 또는 캐리어 웨이브로 구체화되는 컴퓨터 데이터 신호 또는 캐리어에 의해 변조된 신호에 의해 전송 매체 또는 통신 링크를 통해 전송될 수 있다. 머신 판독 가능 매체 또는 프로세서 판독 가능 매체는 머신(예컨대, 프로세서, 컴퓨터 등)에 의해 판독되고 실행 가능한 형태로 정보를 저장 또는 전송할 수 있는 임의의 매체를 포함할 수 있다.

<97> 이상 실시예들은 이해를 돕기 위하여 도면에 도시된 실시예를 참고로 설명되었으나, 이는 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 기술적 보호 범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

발명의 효과

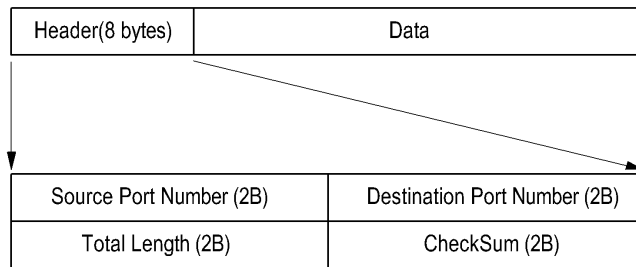
<98> 이상의 실시예들에 따르면, UDP/RTP를 사용하는 망에서 발생하는 지터를 처리하는데 있어서 동적으로 수신단의 버퍼를 동작시키고, 인터넷과 같은 네트워크를 통해 전송된 RTP 패킷의 순서를 재배열함으로써, VoIP와 같은 서비스에 적당하도록 데이터의 전송 품질을 높일 수 있다. 또한, 버퍼의 사이즈와 버퍼의 플레이아웃 문턱값을 사용자 요구에 따라 설정할 수 있도록 하여, 다양한 망 특성에 적응할 수 있다.

도면의 간단한 설명

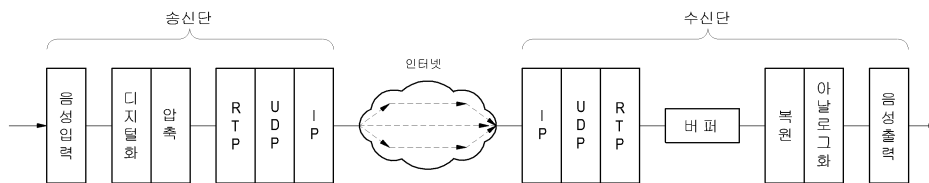
- <1> 도 1은 UDP 패킷 포맷을 나타낸 도면.
- <2> 도 2는 VoIP 망에서의 음성 흐름도.
- <3> 도 3은 RTP 헤더 포맷을 나타낸 도면.
- <4> 도 4는 종래의 버퍼의 구조를 개략적으로 도시한 도면.
- <5> 도 5는 버퍼의 구조의 일 실시예를 개략적으로 도시한 도면.
- <6> 도 6은 디스크립터(descriptor)의 포맷의 일 실시예를 나타낸 도면.
- <7> 도 7은 일 실시예에 따른 데이터 처리 흐름도.
- <8> <도면의 주요 부분에 대한 부호의 설명>
- <9> 40, 50: 버퍼 44: 버퍼 제어 메모리
- <10> 54: 유입 데이터 처리 블록 58: 디스크립터 블록

도면

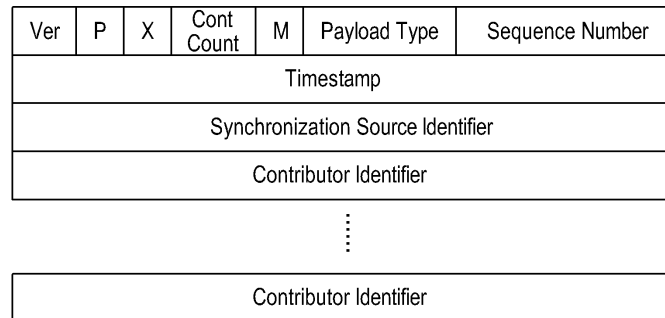
도면1



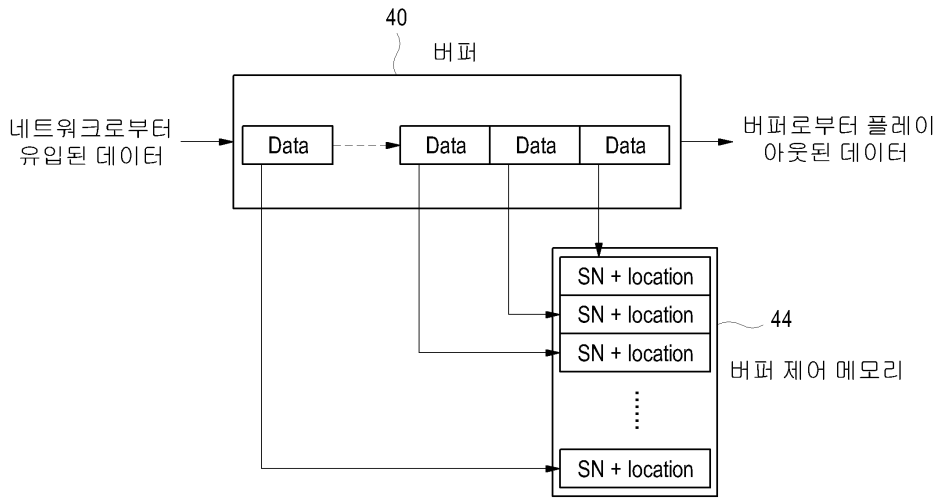
도면2



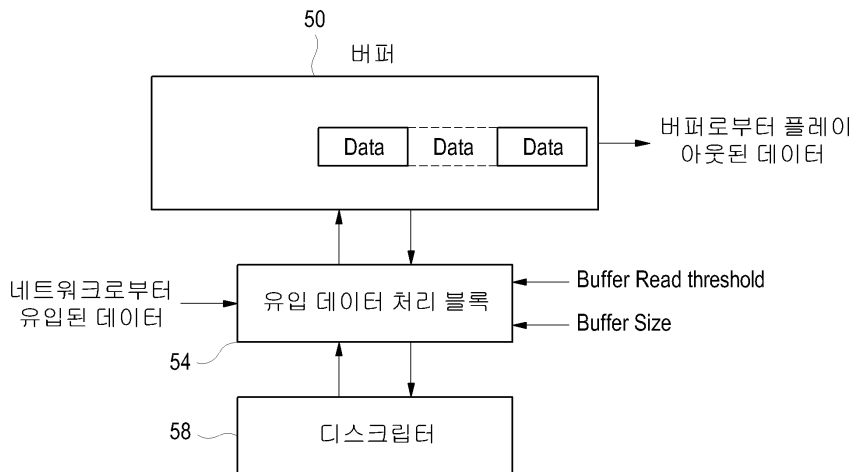
도면3



도면4



도면5



도면6

95	94	93	89	88	84	83	60	59	36	35	20	19	4	3	0
V(63)	F(62)	PacketNo(4:0)	BaseBITMAP_D(4:0)	Base Bitmap(23:0)	BITMAP(23:0)	BaseSN(15:0)	FirstRxSN(15:0)	RoundCount(3:0)							

도면7

