

(72) 발명자

르마르찬드레미에이.

미국98052워싱턴주레드몬드171번애비뉴엔이10416

보닐라니콜에이.

미국98052워싱턴주레드몬드에이피티.303에이149번
플레이스엔이8127

노비트스키로버트알.

미국98053워싱턴주레드몬드엔이101번스트리트22433

그래이로날드이.

미국98052워싱턴주레드몬드186번씨티엔이10310

하트웰아론

미국98019워싱턴주더밸엔이155번플레이스27309

파워브랜단

미국98105워싱턴주시애틀39번애비뉴엔이5215

커티스브랜트

미국98109워싱턴주시애틀플톤스트리트23

특허청구의 범위

청구항 1

컴퓨터 실행 가능한 명령들(instructions)을 갖는 컴퓨터 판독 가능한 기록 매체에 있어서,

상기 명령들은

메시지에 대한 요청을 이메일 서버 컴포넌트에서 수신하는 단계 - 상기 요청은 메시지의 여러 개의 메시지 본문들이 이용가능하다면, 상기 여러 개의 이용가능한 메시지 본문들 중에서 선택하라는 명령들을 포함함 -,

상기 이메일 서버 컴포넌트와 관련된 데이터 스토어(data store)에 액세스하고, 상기 여러 개의 이용가능한 메시지 본문들 중에서 선택하라는 명령들에 기초하여 이용가능한 메시지 본문들의 포맷을 변환하는 것과 무관하게 상기 메시지의 메시지 본문을 선택하는 단계, 및

선택된 메시지 본문의 포맷을 변환하지 않고 선택된 메시지 본문을 검색 및 리턴하는 단계

를 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 2

제1항에 있어서,

상기 요청은 상기 메시지가 위치한 폴더의 동기화 요청을 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 3

제1항에 있어서,

상기 요청은 이메일 메시지들의 복사 요청을 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 4

제1항에 있어서,

상기 요청은 상기 이용가능한 메시지 본문들 중 어느 메시지 본문이 선택될 것인가를 표시하기 위한 플래그를 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 5

제1항에 있어서,

상기 선택하는 단계는 순위 시스템(ranking system)에 따라 이용가능한 메시지 본문들을 평가하는 단계를 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 6

제1항에 있어서,

상기 선택하는 단계는 메시지를 정의하는 속성에 따라 메시지 본문들을 평가하는 단계를 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 7

제6항에 있어서,

상기 속성은 메시지의 헤더(a header for the message)를 포함하는 컴퓨터 판독 가능한 기록 매체.

청구항 8

컴퓨터에 의해 구현된 방법에 있어서,

이메일 클라이언트 컴포넌트에서, 메시지에 대한 요청을 생성하는 단계 - 상기 요청은 메시지의 여러 개의 메시지 본문들이 이용가능하다면, 상기 여러 개의 이용가능한 메시지 본문들 중에서 선택하라는 명령들을 포함함 -,

및

이메일 서버 컴포넌트에서:

상기 요청에 응답해서,

상기 이메일 서버 컴포넌트와 관련된 데이터 스토어에 액세스하고, 상기 여러 개의 이용가능한 메시지 본문들 중에서 선택하라는 명령들에 기초하여 이용가능한 메시지 본문들의 포맷을 변환하는 것과 무관하게 상기 메시지의 메시지 본문을 선택하는 단계, 및

선택된 메시지 본문의 포맷을 변환하지 않고 선택된 메시지 본문을 검색하고 상기 이메일 클라이언트 컴포넌트에게 리턴하는 단계

를 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 9

제8항에 있어서,

상기 요청은 상기 메시지가 위치한 폴더의 동기화 요청을 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 10

제8항에 있어서,

상기 요청은 이메일 메시지들의 복사 요청을 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 11

제8항에 있어서,

상기 요청은 이용가능한 메시지 본문들 중 어느 메시지 본문이 선택될 것인가를 표시하기 위한 플래그를 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 12

제8항에 있어서,

상기 선택하는 단계는 순위 시스템에 따라 이용가능한 메시지 본문들을 평가하는 단계를 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 13

제8항에 있어서,

상기 선택하는 단계는 메시지를 정의하는 속성에 따라 메시지 본문들을 평가하는 단계를 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 14

제13항에 있어서,

상기 속성은 메시지의 헤더를 포함하는, 컴퓨터에 의해 구현된 방법.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0041]

본 출원은 본 명세서에 참조용으로 인용된 "향상된 클라이언트 서버 통신을 위한 시스템 및 방법(SYSTEM AND METHOD FOR IMPROVED CLIENT SERVER COMMUNICATION)"이라는 제목으로 2003년 1월 3일에 출원된 미국 출원 번호 제60/437,869호, 대리인 도켓번호 제220635호의 이익을 청구한다.

[0042]

본 발명은 일반적으로 컴퓨터 네트워크에 관한 것으로, 특히, 이메일 애플리케이션과 같이 클라이언트와 서버 애플리케이션들 간의 통신을 위한 방법에 관한 것이다.

[0043]

이메일은 통신에 있어서 중요한 방법이 되어 왔다. 이메일 시스템들은 통상 서버 컴포넌트(예를 들면, 마이크로소프트 익스체인지 서버) 및 클라이언트 컴포넌트(예를 들면, 마이크로소프트 아웃룩 또는 마이크로소프트 아웃룩 익스프레스)를 포함한다. 이 컴포넌트들은 통상 컴퓨팅 디바이스들(예를 들면, 서버, PC, 랩톱 및 PDA) 상에서 실행하기 위해 구성되는 소프트웨어 애플리케이션들이다.

[0044]

종종, 통신을 용이하게 하기 위해 이메일 시스템의 클라이언트 컴포넌트 및 서버 컴포넌트와 같은 클라이언트 및 서버는 통신 프로토콜에 대해 합의한다. 프로토콜은 예를 들어 요청 및 응답의 예상 시퀀스와 같이 통신 중에 각각의 가입자의 예상 동작을 정의하는 규칙들을 제시한다. 복잡한 프로토콜은 예상되지 않은 동작을 처리하는 규칙을 갖는다.

[0045]

클라이언트 컴포넌트 및 서버 컴포넌트가 향상됨에 따라, 향상된 버전들이 최종 사용자들에게 분산된다. 새로

운 컴포넌트 특징들 및 네트워크 특징들을 이용하기 위해, 새로운 통신 프로토콜이 발명되는 경우가 종종 있다. 서버 컴포넌트들이 설치된 베이스가 중요한 경우, 클라이언트 컴포넌트는 선택된 이전 버전들의 서버 컴포넌트들과 프로토콜 집합을 통해 통신하는 기능을 갖는다.

[0046] 때때로 전체적으로 대체하기보다는 나중 프로토콜들이 이전의 프로토콜들을 개조하는 경우가 있다. 이러한 경우에, 나중 프로토콜은 이전 프로토콜들을 시뮬레이트하기 위해 인에이블되거나 디스에이블되 수 있는 프로토콜 요소들로 구축될 수 있다. 마찬가지로, 클라이언트 컴포넌트들이 설치된 베이스가 중요한 경우, 서버 컴포넌트는 선택된 이전 버전들의 클라이언트 컴포넌트들과 프로토콜을 통해 통신하는 기능을 갖는다.

[0047] 본 발명은 이러한 시스템 및 방법을 제공한다. 본 발명의 다른 특징들 및 본 발명의 여타 장점들은 본 명세서에 제공된 본 발명의 상세한 설명으로부터 명백해질 것이다.

발명이 이루고자 하는 기술적 과제

[0048] 본 발명은 향상된 클라이언트 및 서버 통신을 위한 시스템 및 방법을 제공한다. 특히, 본 발명은 클라이언트와 서버간 통신에 사용될 수도 있는 향상된 프로토콜에 관한 것이다. 본 발명은 특히 이메일 서버 환경과 관련된 것이지만, 본 명세서에 기술된 특징들은 다른 클라이언트 및 서버 네트워크에서도 사용될 수 있다.

[0049] 본 발명의 한 양상에 따라, 이메일 클라이언트 컴포넌트는 이메일 메시지에 유용한 최상의 메시지 본문(body)을 수신하는 것에 관계하는 이메일 서버 컴포넌트에게 지시할 수 있다. 이메일 서버 컴포넌트는 메시지 요청을 수신할 수도 있는데 상기 요청은 메일의 최상 메시지 본문이 요구된 것임을 나타내는 표시를 갖는다. 이메일 서버 컴포넌트는 이메일 서버 컴포넌트와 관련된 데이터 스토어에 액세스할 수 있고 유효 메시지 본문의 포맷을 변환하는 것과 무관하게 유용한 메시지의 최상 메시지 본문을 결정할 수 있으며, 최상 메시지 본문의 포맷을 변환하지 않고 최상 메시지 본문을 검색 및 리턴할 수도 있다. 이처럼, 이메일 서버 컴포넌트에서 이메일 본문의 변환이 일어나지 않기 때문에, 이메일 서버 컴포넌트의 처리 시간이 감소된다.

[0050] 본 발명의 다른 양상에 따라, 특정 속성 또는 속성 집합(예를 들면, 헤더들)의 전송 요청시, 속성 또는 속성들이 데이터 객체 내에 정의되어 있지 않은 경우, 이메일 서버 컴포넌트는 전체 데이터 객체를 전송할 수도 있다. 이메일 클라이언트 컴포넌트는 폴더의 데이터 객체에 대한 요청을 생성하는데, 상기 요청은 데이터 객체의 적어도 하나의 속성이 요구된 것임을 나타내는 표시를 포함한다. 이메일 서버 컴포넌트는 요청을 수신하고, 폴더 및 폴더의 데이터 객체들에 액세스한다. 폴더의 각각의 데이터 객체에 있어서, 적어도 하나의 속성이 데이터 객체에서 정의되어 있으면, 이메일 서버 컴포넌트는 데이터 객체의 적어도 하나의 속성을 검색하고 이메일 클라이언트 컴포넌트에게 리턴한다. 데이터 객체에 대해 적어도 하나의 속성도 정의되지 않은 경우, 이메일 서버 컴포넌트는 데이터 객체를 검색하여 이메일 클라이언트 컴포넌트에게 리턴한다.

[0051] 본 발명의 또 다른 양상에 따라, 이메일 클라이언트 컴포넌트는 강제로 이메일 서버 컴포넌트가 이메일 메시지를 유니코드로 제공하게 한다. 이메일 클라이언트 컴포넌트는 적어도 하나의 이메일 메시지에 대한 요청 및 이메일 클라이언트 컴포넌트가 이메일 메시지들이 유니코드 포맷으로 되기를 요구한다는 표시를 송신한다. 이메일 서버 컴포넌트는 상기 요청 및 표시를 수신한 응답으로 적어도 하나의 메시지를 검색하고, 각각의 이메일 메시지에 대해, 이메일 메시지가 유니코드 포맷으로 이용가능하면, 유니코드 포맷을 이메일 클라이언트 컴포넌트에게 제공한다. 이메일 메시지가 유니코드 포맷으로 이용가능하지 않으면, 이메일 서버 컴포넌트는 이메일 메시지를 유니코드 포맷으로 변환하여 유니코드 포맷을 이메일 클라이언트 컴포넌트에게 제공한다.

[0052] 본 발명의 또 다른 양상에 따라, 이메일 클라이언트 컴포넌트에 의해 송신된 요청은 요청에 대한 응답 크기에는 제한이 없음을 나타낼 수도 있으며, 필요한 경우 이메일 서버 컴포넌트가 베퍼를 채울 수 있게 한다. 이메일 클라이언트 컴포넌트는 요청 내의 다수의 서브요청들(subrequests)을 송신하는데, 서브요청들의 각각은 이메일 서버 컴포넌트의 동작을 요청하며 크기 정보를 포함한다. 각각의 서브요청에 응답해서, 크기 정보가 이메일 서버 컴포넌트가 예상하는 범위 내의 크기 한계를 포함하면, 이메일 서버 컴포넌트는 크기 한계에 대한 응답을 제한한다. 크기 정보가 이메일 서버 컴포넌트가 예상한 범위 밖의 크기 한계를 포함하면, 이메일 서버 컴포넌트는 크기 정보에서 새로운 크기 한계를 찾는다. 새로운 크기 한계는 "유효 베퍼를 채우는" 것과 같이 임의적일 수도 있다.

발명의 구성 및 작용

[0053] 본 발명의 다양한 실시예들이 기술되기에 앞서, 본 발명의 다양한 실시예들이 실행될 수도 있는 컴퓨터 및 네트워크 환경이 이제부터 기술될 것이다. 필요하지 않더라도, 본 발명은 컴퓨터에 의해 실행되는 프로그램들로 구

현될 수도 있다. 일반적으로, 프로그램들은 특정 태스크들을 실행하거나 특정 추상적 데이터 타입들을 구현하는 루틴들, 객체들, 컴포넌트들, 데이터 구조들 등을 포함한다. 본 명세서에서 사용되는 "프로그램"이라는 용어는 단일 프로그램 모듈 또는 일제히 동작하는 다중 프로그램 모듈들을 의미할 수도 있다. 본 명세서에서 사용되는 "컴퓨터"라는 용어는 퍼스널 컴퓨터(PC), 핸드헬드 디바이스, 멀티프로세서 시스템, 마이크로프로세서-베이스 프로그램 가능한 소비 전자기기, 네트워크 PC, 미니컴퓨터, 태블릿 PC, 메인프레임 컴퓨터, 마이크로프로세서 또는 마이크로컨트롤러를 갖는 소비 제품, 라우터, 게이트웨이, 허브 등과 같이 하나 이상의 프로그램들을 전자적으로 실행하는 임의의 디바이스를 포함한다. 본 발명은 또한 통신 네트워크를 통해 연결된 원격 처리 장치에 의해 테스크가 실행되는 분산 컴퓨팅 환경에서도 사용될 수 있다. 분산 컴퓨팅 환경에서, 프로그램들은 로컬 및 원격 메모리 스토리지 디바이스들 모두에 위치할 수 있다.

[0054] 본 발명이 사용될 수 있는 네트워크 환경의 일례가 이제부터 도 1을 참조하여 기술될 것이다. 일례의 네트워크는 구름 모양으로 도시된 네트워크(11)를 통해 서로 통신하는 수개의 컴퓨터들(10)을 포함한다. 네트워크(11)는 라우터, 게이트웨이, 허브 등과 같은 다수의 공지된 컴포넌트들을 포함할 수도 있으며, 유선 및/또는 무선 매체들을 통해 컴퓨터들(10)이 통신할 수 있게 해준다. 네트워크(11)를 통해 상호 작용할 때, 컴퓨터들 중 하나 또는 그 이상의 컴퓨터들은 다른 컴퓨터들에 대해 클라이언트, 서버 또는 피어(peer)로 작용할 수 있다. 따라서, 본 명세서에 포함된 특정 일례들이 이러한 타입들의 컴퓨터들 모두에 대해 언급하지 않더라도, 본 발명의 다양한 실시예들은 클라이언트, 서버, 피어 또는 그 결합체에서 실행될 수도 있다.

[0055] 도 2를 참조해 보면, 본 명세서에 기술된 본 발명의 전부 또는 일부가 구현될 수 있는 컴퓨터의 기본 구성의 예가 도시되어 있다. 가장 기본적인 구성으로, 컴퓨터(10)는 통상 적어도 하나의 프로세싱 유닛(14) 및 메모리(16)를 포함한다. 프로세싱 유닛(14)은 본 발명의 다양한 실시예들에 따라 태스크들을 수행하기 위해 명령들을 실행한다. 태스크들을 실행할 때, 프로세싱 유닛(14)은 소정의 결과를 야기하기 위해 컴퓨터(10)의 다른 파트들 및 컴퓨터(10) 밖의 디바이스들에게 전자 신호들을 송신할 수 있다. 컴퓨터(10)의 정확한 구성 및 타입에 따라, 메모리(16)는 휘발성(예를 들면 RAM)일 수도 있고, 불휘발성(예를 들면 ROM 또는 플래시 메모리)일 수도 있으며, 아니면 2개의 결합물일 수도 있다. 이 가장 기본적인 구성은 도 2에 점선(18)으로 도시되어 있다. 또한, 컴퓨터는 추가의 특징들/기능들을 가질 수도 있다. 예를 들어, 컴퓨터(10)는 또한 자기 또는 광 디스크 또는 테이프를 포함하지만 이에 제한되지 않는 추가의 스토리지(제거 가능 스토리지(201) 및/또는 제거 불가능 스토리지(202))를 포함할 수도 있다. 컴퓨터 기억 매체는 컴퓨터 실행 가능한 명령들, 데이터 구조, 프로그램 모듈 또는 다른 데이터를 포함하는 정보 기억을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 불휘발성, 제거 가능 및 제거 불가능 매체들을 포함한다. 컴퓨터 기억 매체는 RAM, ROM, EEPROM, 플래시 메모리, CD-ROM, DVD(digital versatile disk) 또는 다른 광 스토리지, 자기 카세트, 자기 테이프, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 또는 희망 정보를 기억하는데 사용될 수 있고 컴퓨터(10)에 의해 액세스될 수 있는 임의의 다른 매체를 포함하며, 이에 제한되는 것은 아니다. 임의의 컴퓨터 기억 매체는 컴퓨터(10)의 일부일 수 있다.

[0056] 컴퓨터(10)는 양호하게 디바이스가 다른 디바이스들과 통신할 수 있게 해주는 통신 커넥션(205)을 포함한다. 통신 커넥션은 통신 매체의 일례이다. 통신 매체는 통상 컴퓨터 판독 가능한 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터를 반송파와 같은 변조된 데이터 신호로 또는 다른 전송 메카니즘으로 구현하며 임의의 정보 전달 매체들을 포함한다. 예를 들어, "통신 매체"라는 용어는 유선망 또는 다이렉트-와이어드 커넥션과 같은 유선 매체들 및 음향, RF, 적외선 및 다른 무선 매체들과 같은 무선 매체들을 포함하며, 이에 제한되는 것은 아니다. 본 명세서에서 사용된 "컴퓨터 판독 가능한 매체"라는 용어는 컴퓨터 기억 매체 및 통신 매체 모두를 포함한다.

[0057] 컴퓨터(10)는 또한 키보드, 마우스, 펜, 음성 입력 디바이스, 터치 입력 디바이스 등과 같은 입력 디바이스들(204)을 포함할 수도 있다. 디스플레이(20), 스피커, 프린터 등과 같은 출력 디바이스들(203)이 포함될 수도 있다. 상기 모든 디바이스들은 종래 기술에서 널리 공지된 것으로 본 명세서에는 장황하게 기술될 필요가 없다.

[0058] 본 발명은 향상된 클라이언트 및 서버 통신을 위한 시스템 및 방법에 관한 것으로, 특히, 클라이언트와 서버 간의 통신을 위해 사용될 수 있는 향상된 프로토콜에 관한 것이다. 본 발명은 특히 이메일 서버 환경에 관한 것이지만, 본 명세서에 기술된 특징들이 다른 클라이언트 및 서버 네트워크에서 사용될 수도 있다. 그러나, 편의상, 본 발명은 클라이언트/서버 이메일 환경과 관련해서 기술된다.

[0059] 본 발명은 2개 이상의 벤더들의 클라이언트 애플리케이션들 또는 컴포넌트들 및/또는 2개 이상의 벤더들의 서버

애플리케이션들 또는 컴포넌트들을 갖는 클라이언트/서버 환경에서 구현될 수 있다. 이를 위하여, 도 3은 네트워크 이메일 환경의 다수의 버전들의 클라이언트 및 서버 컴포넌트들을 도시한 블록도를 보여준다. 일반적으로, 클라이언트 및 서버 컴포넌트들은 역방향으로 호환 가능하도록 구성된다. 즉, 클라이언트 컴포넌트는 서버 컴포넌트들의 최신 버전 및 레거시 버전들과 통신할 수 있으며, 그 반대도 마찬가지다. 프로토콜 집합은 다수의 버전들 간의 통신을 위해 설정된다. 프로토콜 집합은 수개의 상이한 프로토콜들을 구성할 수 있으며, 각각은 독립적이다. 대안으로, 프로토콜 컴포넌트 집합이 사용될 수 있으며, 특정 컴포넌트들은 프로토콜 집합 내의 특정 프로토콜들을 구성하는데 사용된다.

[0060] 임의의 경우에, 도 3에 도시된 네트워크 이메일 환경에서, 가장 최신 버전의 이메일 클라이언트 컴포넌트(303)는 프로토콜(307)을 사용해서 가장 최신 버전의 이메일 서버 컴포넌트(306)와 통신한다. 그러나, 가장 최신 버전의 이메일 서버 컴포넌트(306)는 프로토콜 집합의 다른 프로토콜들(예를 들어, 도 3의 프로토콜들(308 및 309))을 사용해서 예를 들어 이메일 클라이언트 컴포넌트(302) 및 이메일 클라이언트 컴포넌트(301)과 같이 선택된 이전 버전의 이메일 클라이언트 컴포넌트들과도 통신할 수 있다. 이메일 클라이언트 컴포넌트(303)도 또한 프로토콜들(310 및 311)과 같이, 프로토콜들을 사용해서 예를 들어 이메일 서버 컴포넌트(305) 및 이메일 서버 컴포넌트(304)와 같은, 선택된 이전 버전의 이메일 서버 컴포넌트들과 통신할 수 있다.

[0061] 일반적으로, 본 명세서에서 사용된 바와 같이, 본 발명의 프로토콜을 기술하기 위해, "가장 최신의" 이메일 (서버 또는 클라이언트) 컴포넌트 또는 가장 최신 버전의 이메일 (서버 또는 클라이언트) 컴포넌트는 기술된 새로운 특징 또는 특징들을 알고 있는 서버 또는 클라이언트 컴포넌트이며, 다른 특징들을 사용하고, 구현하며 실행할 수 있다. 본 발명의 프로토콜의 다양한 양상들을 알고 있는 클라이언트 및 서버 컴포넌트들을 기술하기 위해 본 명세서를 통해 상기 용어들이 사용되더라도, 상기 용어들은 또한 기술되고 있는 특정 양상만을 또는 기술되고 있는 양상 외의 것들을 알고 있는 컴포넌트들을 포함한다. 마찬가지로, "이전" 이메일 컴포넌트 또는 이전 버전의 이메일 컴포넌트는 본 발명의 프로토콜의 양상들을 알고 있지 않으며 실행할 수 없는 컴포넌트이다.

[0062] 클라이언트와 서버(예를 들어, 가장 최신 버전의 이메일 서버 컴포넌트(306) 및 가장 최신 버전의 이메일 클라이언트 컴포넌트(303)) 간의 프로토콜을 설정하기 위해 프로토콜 협상 프로시저(protocol negotiation procedure)가 종종 사용된다. 이러한 프로토콜 협상은 공지된 것이지만, 독자의 편의를 위해 이메일 클라이언트 컴포넌트(401; 도 4)와 이메일 서버 컴포넌트(402; 도 4) 간의 프로토콜 협상 프로시저가 간략히 기술된다. 이메일 클라이언트 컴포넌트(401)와 이메일 서버 컴포넌트(402) 간의 통신 세션 전에, 이메일 클라이언트 컴포넌트(401)는 클라이언트 버전 정보를 포함하는 메시지(403)를 예를 들어 클라이언트 컴포넌트 버전 스템프의 형태로 이메일 서버 컴포넌트(402)에 송신한다. 이메일 서버 컴포넌트(402)는 예를 들어 서버 컴포넌트 버전 스템프의 형태로 수개의 버전 정보를 포함하는 메시지(404)로 상기 메시지(403)에 응답한다.

[0063] 클라이언트 및 서버 버전 정보는 이메일 클라이언트 컴포넌트(401)와 이메일 서버 컴포넌트(402) 간의 통신 설정을 시도하기 위해 다양한 방법들로 사용될 수도 있다. 예를 들어, 버전 정보는 연속 통신에 적합한 프로토콜을 선택하거나, 또는 다른 통신이 가능한지를 결정하는데 사용될 수도 있다. 프로토콜 설정시, 예를 들어, 버전 정보는 특정 유효 프로토콜 양상들 또는 컴포넌트들을 인에이블 및/또는 디스에이블하는데 사용될 수도 있다.

[0064] 이메일 서버 컴포넌트는 다수의 이메일 클라이언트 컴포넌트들로부터의 요청들을 병렬로 수신 및 처리할 수 있다. 달리 명백하게 기술되지 않는 한, 단일 클라이언트가 도시된 것은 단지 도면들 및 첨부하는 설명을 간단히 하기 위한 것이다.

[0065] 본 발명의 이메일 네트워크는 네트워크에서 클라이언트와 서버 컴포넌트들 간의 질의 및 데이터 전달을 위해 요청 및 응답 교환들을 사용한다. 실제로, 프로토콜의 성능은 이메일 네트워크에서 클라이언트와 서버 간의 통신을 구현하는데 사용되는 기초 통신 네트워크 전송 메카니즘에 의해 영향을 받을 수도 있다. 예를 들어, 기초 통신 네트워크 전송 메카니즘으로서 원격 프로시저 콜(RPC; remote procedure call)을 사용하는 이메일 네트워크에서, 보다 작은 크기(예를 들어, 2KB)의 수개의 원격 프로시저 콜들을 행하는 것 보다는 보다 큰 크기(예를 들어, 32KB)의 단일 원격 프로시저 콜을 행하는 것이 훨씬 효율적일 수도 있다. 이러한 이메일 네트워크에서 성능을 향상시키기 위해 공지된 한가지 방법은 단일 원격 프로시저 콜로 송신을 위한 다수의 요청들 및/또는 응답들을 버퍼링하는 것이다.

[0066] 일례로서, 도 5는 이메일 클라이언트 컴포넌트(501)와 이메일 서버 컴포넌트(502) 간의 요청 및 응답 교환을 도시한 것이다. 이메일 클라이언트 컴포넌트(501) 및 이메일 서버 컴포넌트(502)는 각각 고정 크기 통신 버퍼들

(503, 504, 505 및 506)을 갖는다. 베피들(503, 504, 505 및 506)은 일시적으로 데이터를 유지하기 위한 메모리의 예약된 영역이다. 이메일 클라이언트 컴포넌트(501)는 베피(503)의 콘텐츠를 베피(504)에 송신하기 전에 하나 이상의 서브요청들 또는 원격 오퍼레이션들(ROP; remote operation)로 베피(503)를 채움으로써 요청-응답 사이클을 개시한다.

[0067] 베피(504)에 수신된 후에, 각각의 ROP는 이메일 서버 컴포넌트(502)에 의해 순서대로 처리되고 대응 결과는 베피(505)에 기록된다. 각각의 ROP는 소정의 결과를 생성한다. 결과는 예를 들어 특정 집합의 이메일 메시지들과 같이 이메일 클라이언트 컴포넌트(501)에 의해 요청된 데이터를 포함한다. 이메일 서버 컴포넌트(502)는 베피(505)를 모니터하고, 거의 다 찼을 때(예를 들어, 나머지가 8KB 미만일 때), 이메일 서버 컴포넌트(502)는 임의의 처리되지 않은 ROP들을 베피(505)의 끝에 기록하고 베피(505)를 베피(506)에 송신한다. 그 후 이메일 클라이언트 컴포넌트(501)는 베피(503)가 다시 찼을 때 이메일 서버 컴포넌트(502)에게 다시 제시하기 위해 베피(503)에 처리되지 않은 ROP들을 기록함으로써 새로운 요청-응답 사이클을 개시한다.

[0068] 응답의 크기는 통상적으로 요청의 크기 보다 평균적으로 더 크다. 예를 들어, 응답 베피들(505 및 506)의 크기는 통상적으로 요청 베피들(503 및 504)의 크기보다 더 크도록 구성된다. 본 발명의 일 실시예에서, 응답 베피들(505 및 506)의 최적 크기는 요청 베피들(503 및 504)의 크기, 32KB의 3배인 96KB로 결정되었다. 일 실시예에서, 이메일 클라이언트 컴포넌트는 베피들(503, 504, 505 및 506) 중 임의의 베피의 크기를 구성할 수 있다.

[0069] 베피들을 사용하는 몇몇 이메일 네트워크들, 예를 들어, 도 5에 도시된 이메일 네트워크는 이메일 클라이언트 컴포넌트와 이메일 서버 컴포넌트 간의 고속 전송 모드를 이용할 수 있다. 고속 전송 모드는 클라이언트에 의해 적어도 2개의 종류들로 분할된 ROP들과 같은 요청들, 서버에서 고속 전송 데이터 소스의 초기화를 야기하는 요청들, 및 고속 전송 데이터 소스로부터 클라이언트로의 데이터의 효율적인 전송을 야기하는 요청들을 포함한다. 예를 들어, 고속 전송 데이터 소스는 데이터베이스 테이블일 수도 있다. 고속 전송 데이터 소스는 후에 데이터에 대한 요청들이, 다른 경우에 가능한 것 보다 덜 지연되어 서비스될 수 있게 해주는 준비된 일시적인 스토어로서 작용한다. 때때로 제2 종류의 고속 전송 모드 요청은 응답의 크기를 명백하게 지정함으로써 효율적인 데이터 전송을 달성하고자 하는데, 예를 들어, 응답의 크기는 전체 클라이언트 수신 베피에서 응답 오버헤드를 뺀 크기로 설정될 수도 있다.

[0070] 도 6a는 적어도 2개의 요청-응답 사이클들을 갖는 고속 전송 오퍼레이션들을 도시한 것이다. 제1 요청(601)시, ROP(예를 들어, FXPrepare)는 서버(502)에서 고속 전송 데이터 소스를 초기화한다. 서버에서, FXPrepare만이 처리되고(즉, 고속 전송 데이터 소스가 초기화되고) 그 결과는 제1 응답(602)시 리턴된다. 제2 요청(603)시, ROP(예를 들어, FXGetBuffer)는 고속 데이터 소스로부터 베피(505)를 채우도록 서버에게 요청한다. 서버는 고속 데이터 소스를 베피로 보내어 비우고, 그 결과를 제2 응답(604)시 리턴한다. 고속 데이터 소스가 비워지기 전에 이메일 서버 컴포넌트를 위한 출력 베피(505)가 채워지면, 추가의 FXGetBuffer ROP들이 요구될 수도 있다.

[0071] 도 6b는 오직 단일 요청-응답 사이클만을 갖는 고속 전송 오퍼레이션을 도시한 것이다. 제1 요청(605)시, FXPrepare 및 FXGetBuffer가 모두 이메일 서버 컴포넌트(502)에 의해 처리되고 양 오퍼레이션들의 결과들이 제1 응답시(606) 리턴된다. 각 베피(503, 504, 505 및 506)의 일부분이 공유 데이터 표로서 명료하게 정의되기 때문에, FXPrepare의 결과는 이메일 서버 컴포넌트(502)에서 FXGetBuffer에 유효하다. 요청-응답 사이클들의 수를 감소시키는 것은 데이터의 보다 효율적인 전송을 야기하기 때문에 바람직하다. 단일 요청-응답 사이클 보다 많은 요청-응답 사이클들을 갖는 고속 전송 오퍼레이션은, 베피(505)가 차서 FXGetBuffer ROP의 결과를 유지하는 못하는 경우를 야기할 수도 있다.

[0072] 달리 명백하게 기술되지 않는 한 일련의 ROP들에 의해 실제로 구현될 수도 있다는 점에서 도 6a, 도 6b 및 본 출원서 도처의 유사한 도면들의 ROP들은 개략적인 것임을 알게 될 것이다.

[0073] 통상, ROP 결과의 크기는 ROP 요청의 크기와 다르다. ROP 결과 크기를 항상 예측할 수 있는 것은 아니다. 데이터 압축 기술들이 ROP 결과의 크기를 감소시키기 위해 사용될 때, ROP 결과의 크기를 예측하는 것은 더 어렵게 된다. ROP 결과의 크기를 예측할 수 없다는 점은 특정 클라이언트 오퍼레이션들을 완료하는데 필요한 응답-요청 사이클들의 수를 최소화하기 위해, 예를 들어, 모든 새로운 메시지들이 단일 요청-응답 사이클로 클라이언트에게 다운로드됨을 확증하기 위해, 프로토콜의 수동 튜닝을 방지할 수 있게 해준다. 프로토콜의 수동 튜닝은 프로토콜 요청들, 응답들 및/또는 ROP들의 시퀀스 및/또는 크기의 수동 구성을 포함한다.

[0074] 본 발명의 한 양상에 따라, 중요한 ROP들(예를 들어, FXGetBuffer)이 그 결과의 크기를 예측하기 위한 요구사항의 제약을 받지 않는다고 명시함으로써 요청-응답 사이클들의 수는 자동으로 최소화된다. 대신에, 베피

(505)(버퍼(506)와 동일함)의 한계에 도달할 때까지 ROP들은 이메일 서버 컴포넌트(502)에 의해 처리된다.

[0075] 일례로서, 다수의 버전들의 이메일 서버 컴포넌트들을 포함하는 환경에서, 개별 ROP들이 이전 버전의 서버 컴포넌트들 및 최신 버전의 서버 컴포넌트들에 대해 정의될 수도 있다. 최신 버전들은 결과의 크기를 예측하기 위한 요구 사항의 제약을 받지 않는다. 이들 ROP들의 특징들은 이하의 표에 기술된다:

	이전 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP	가장 최신 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP
ROP ID	FXGetBuffer	FXGetBuffer
다수의 모드들에서 사용되는 파라미터들	요구된 크기: 서버가 출력 버퍼에서 확보해야만 하는 크기	요구 크기: 이전 버전에 의해 예상된 최대값을 넘어서는 값, 예를 들어, 32KB 보다 큰 값으로 설정된다. 이는 서버에게 새로운 크기 한계 파라미터를 찾도록 신호해주는 것이다.
새로운 파라미터들	n/a	크기 한계: 서버가 출력 버퍼를 채울 수 있는 한계를 서버에게 알려준다.

[0077] 이전 버전의 서버 컴포넌트들을 위한 ROP들은 구성이 현준하는 종래 기술의 ROP들과 유사하다. 즉, ROP들은 응답을 유지하기 위해 확보되어야만 하는 출력 버퍼(예를 들어, 송신 버퍼(505))에서의 크기를 예측하고 규정한다. 대조적으로, 가장 최신 버전의 서버 컴포넌트를 위해 규정된 출력 버퍼에 대한 크기는 예측되지 않는데, 그 대신 이전 버전의 서버 컴포넌트에 의해 예측된 최대값을 넘어서는 값, 예를 들어, 32KB 보다 큰 값으로 설정된다. 출력 버퍼 크기가 서버 컴포넌트에 의해 예상되는 값을 넘어서도록 설정된다는 사실은 서버 컴포넌트에게 예를 들어 서버 컴포넌트를 위한 출력 버퍼의 필링(filling)일 수도 있는 새로운 크기 한계 파라미터를 찾아 보라고 신호해 주는 것이다. 이 특징들은 요청-응답 사이클들의 수를 자동으로 최소화해주는데, ROP들을 처리하는 이메일 서버 컴포넌트의 복잡성은 단지 약간만 증가한다.

[0078] 명백히 기술되지 않는 한, 상기 표나 본 출원서 도처의 유사한 표들에 도시된 파라미터들의 순서는 이메일 클라이언트 컴포넌트 또는 이메일 서버 컴포넌트에 의해 파라미터들이 네트워크를 통해 송신되거나 메모리에 저장되는 순서와 반드시 상관있는 것이 아님을 유의한다. 또한, 편의상 변경되지 않은 파라미터들은 생략될 수도 있다.

[0079] 이메일 네트워크에서, 프로토콜의 전형적인 임무들 중 하나는 이메일 클라이언트 컴포넌트들과 이메일 서버 컴포넌트들 간의 데이터 객체들, 예를 들어, 이메일 메시지들의 전송을 달성하는 것이다. 데이터 객체들의 다른 예로서 예를 들어 이메일 메시지들 및 다른 데이터 객체들을 포함할 수도 있는 이메일 폴더들, 및 이메일 메시지들을 처리하기 위한 규칙들을 포함하거나 폴더에 포함된 데이터 객체들이 어떻게 표시될 것인지를 정의할 수도 있는 폴더 연관 정보(FAI; folder associated information) 데이터 객체들을 포함한다. 데이터 객체들은 이메일 클라이언트 컴포넌트에게 불투명할 수도 있는데, 즉, 이메일 클라이언트 컴포넌트는 데이터 객체의 콘텐츠를 해석하는 수단을 갖지 않을 수도 있다. 대안적으로, 데이터 객체들은 명칭 속성들로 구성될 수도 있다. 예를 들어, 이메일 메시지는 "수신원(to)", "송신원(from)", "주제", "중요도", "본문 1", "본문 2", "본문 3", "첨부 1", "첨부 2" 등으로 명명된 속성들을 포함할 수도 있다.

[0080] 데이터 객체들이, 데이터 객체들이 불투명한 이메일 네트워크들을 통해 명명된 속성들로 구성될 수도 있는 이메일 네트워크들의 한가지 장점은, 데이터 객체 부분만을 전송할 수 있는 프로토콜의 능력 때문에 프로토콜 성능을 향상시킬 수도 있다는 점이다. 명명된 속성들은 전체 데이터 객체를 전송하지 않고 데이터 객체의 특정 속성들이 전송될 수 있게 해준다.

[0081] 예를 들어, 이메일 메시지는 헤더 속성들의 집합 및 본문 속성들의 집합으로 구성될 수 있다. 이메일 클라이언트 컴포넌트의 요구 사항들은 프로토콜이 헤더 속성들을 먼저 전송한 후에 본문 속성들을 전송하거나 아니면 아예 전송하지 않을 수도 있다는 점이다. 이러한 특징은 사용자가 모든 메시지들이 전부 다운로드되기 전에 수개의 메시지들에 대한 헤더 정보를 볼 수 있게 해준다. 이러한 특징을 사용해서, 대역폭 활용을 통한 보다 미세한 제어가 클라이언트 컴포넌트에 의해 획득될 수도 있는데, 이는 프로토콜 성능에 실제적으로 영향을 줄 수 있다. 또한, 클라이언트 컴포넌트는 이러한 특징을 사용해서 대역폭 활용을 절감할 수 있는데(예를 들어, 오직 선택된 헤더들에 대해서만 본문들이 다운로드될 수도 있음), 이는 특히 저 대역폭 환경에 바람직하다.

서버 컴포넌트가 2개의 개별 요청-응답 사이클들(즉, 헤더 및 본문에 각각 하나의 사이클)로 본문 및 헤더 속성들을 송신하도록 구성되면, 프로토콜 성능이 반드시 증가하는 것은 아니다. 예를 들어, 이메일 클라이언트 컴포넌트의 요구사항이 헤더 및 본문 속성들을 동시에 필요로 하는 것이라면, 단일 요청-응답 사이클이 헤더 및 본문을 모두 검색할 수 있는 상황에 비해 프로토콜 성능이 감소될 수 있다. 따라서, 데이터 객체들이 명명된 속성들로 구성될 수 있게 해주는 간단한 동작은 그 자체로 프로토콜 성능을 자동적으로 향상시키기에는 충분치 않다. 프로토콜 성능을 향상시키는 것은 데이터 객체를 구성하는 속성들을 선택하는 것과 속성들이 프로토콜에 의해 사용되는 방법에 의해 좌우된다. 상기 선택은 가장 최신 및 이전 버전의 이메일 클라이언트 컴포넌트들의 요구 사항들 및 가장 최신 및 이전 버전의 이메일 서버 컴포넌트들의 요구 사항들을 포함하는 다수의 요인들에 따라 좌우될 수 있다. 이메일 클라이언트 컴포넌트 요구 사항의 일례들은 상이한 정보의 디스플레이를 위한 상이한 긴급성 레벨을 충족시키는 것과 이메일 클라이언트 컴포넌트 사용자에 의해 설정된 프리퍼런스 (preference)에 부응하는 것을 포함한다. 이메일 서버 컴포넌트 요구 사항들의 예들은 데이터의 효율적인 기억 및 검색과 프로토콜 요청들의 효율적인 처리를 포함한다.

종래 기술의 이메일 환경은 명명된 속성들로 구성될 수 있는 데이터 객체들을 사용하는데, 예를 들어, 명명된 속성들의 헤더 집합 및 본문 집합을 포함하여 2개의 집합들이 개별적으로 요청 및/또는 처리될 수 있게 해 주는 이메일 메시지를 사용한다. 다른 종래 기술의 예는 명명된 속성들의 본문 집합이 평문(plain text), HTML(hypertext mark-up language), RTF(rich text format) 등과 같은 다수의 이메일 메시지 포맷들로 된 이메일 메시지 본문의 다수의 버전들을 포함하는 이메일 메시지이다. 이러한 상황에서, 종래 기술의 이메일 서버 컴포넌트들은 이메일 메시지의 본문에 대한 프로토콜 요청에 다수의 방법들로 응답할 수 있다. 최저 복잡성 응답은 이메일 메시지 본문의 모든 버전들을 송신하는 것일 수 있지만, 이러한 응답은 대역폭 활용을 증가시킬 수 있다.

도 7a는 이러한 상황에서 이전 (종래 기술) 버전의 이메일 서버 컴포넌트가 응답하기 위해 사용하는 프로시저를 도시한 것이다. 단계(701)에서, 이메일 서버 컴포넌트는 각각의 이메일 메시지 본문의 포맷을 검사한다. 포맷들 중 하나가 소정의 표준 포맷(예를 들어, RTF)이면, 프로시저는 단계(703)로 진행하고 표준 포맷 이메일 메시지 본문이 요청 이메일 클라이언트 컴포넌트에게 송신된다. 포맷들 중 어느 것도 소정 표준 포맷이 아니면, 단계(701)는 단계(702)로 진행해서 이메일 메시지 본문 버전들 중 하나가 표준 포맷으로 변환된다. 도 7a에 도시된 서브프로시저는 이메일 메시지 본문의 단일 버전만이 존재하는 경우에 사용될 수도 있지만, 이메일 메시지 본문은 프로토콜에 의해 요구된 표준 포맷이 아닐 수 있다.

도 7b는 본 발명에 따라 가장 최신 버전의 이메일 메시지 컴포넌트가 사용하는 프로시저의 부분을 도시한 것이다. 단계(704)에서, 이메일 서버 컴포넌트에 의해 사용중인 서브프로시저를 야기하는 프로토콜 요청은 BEST_BODY 플래그를 검사한다. 이메일 클라이언트 컴포넌트가 가장 최신 버전이고 플래그와 관련된 기능을 구현하기를 희망한다는 본 예의 플래그 및 본 명세서에서 사용된 다른 플래그들은 이메일 서버 컴포넌트에게 사용된다. 다른 표시들이 사용될 수도 있다. 예를 들어, 상기 기능은 가장 최신의 이메일 클라이언트 컴포넌트가 검출되면 디폴트로 구현될 수도 있다.

임의의 경우에, BEST_BODY 플래그가 발견되지 않으면, 단계(704)는 단계(701)로 진행하고, 도 7a를 참조해서 설명된 바와 같이 계속된다.

플래그가 발견되면, 프로시저는 단계(705)로 진행해서, 최상 이메일 메시지 본문이 요청 이메일 클라이언트 컴포넌트에게 송신되기 위해 선택된다. 요청 이메일 메시지와 관련된 이메일 메시지 본문이 하나만 존재하면, 바로 그것이 최상 이메일 메시지 본문이다. 예를 들어, 상이한 포맷들을 갖는 유효한 이메일 메시지 본문들이 수개이면, 이메일 서버 컴포넌트는 예를 들어 이메일 메시지 본문 포맷들(예를 들어, RTF, HTML, 평문)의 소정 순위에 따라 그들 중에서 최상 이메일 메시지 본문을 선택한다. 그 다음에 프로세스는 단계(703)로 진행해서, 선택된 이메일 메시지 본문이 이메일 클라이언트 컴포넌트에게 송신된다. 본 실시예에서, 이메일 클라이언트 컴포넌트는 다수의 이메일 메시지 본문 포맷들을 표시할 수 있고, 따라서 이메일 서버 컴포넌트는 이메일 메시지 본문들을 표준 포맷으로 변환하라는 요구로부터 자유로워진다. 또한, 이메일 클라이언트 컴포넌트는, 필요하다면, 최상 이메일 메시지 본문을 상이한 포맷으로 변환할 수도 있다.

이메일 서버 컴포넌트가 이메일 메시지 본문들을 변환하는 임무로부터 해방되기 때문에, 본 발명은 향상된 성능을 제공한다. 또한, 가장 최신 버전의 이메일 서버 컴포넌트는 복잡성이 적당하게 증가한 채로 이전 버전의 이메일 클라이언트 컴포넌트로부터의 프로토콜 요청들에 응답할 수도 있다.

ROP들을 이용해 서버 커먼드와 이미일 클라이언트 커먼드 간의 이미일 풀더의 복제를 달성하는데 사용될 수

있다. 폴더 동기화 요청이 예를 들어 SynchFolder ROP에 의해 달성될 수 있다. 이메일 클라이언트 컴포넌트가 표준이 아닌 이메일 메시지 본문 포맷들을 표시할 수 있는 경우, 그것은 서버에게 이메일 메시지 본문을 표준 포맷으로 리턴하도록 요구하기 보다는 이메일 서버 컴포넌트가 유효한 이메일 메시지 본문들 중에서 최상 포맷을 선택할 수 있음을 나타내기 위해 SynchFolder ROP에서 BEST_BODY 플래그를 설정할 수 있다. 이메일 서버 컴포넌트는 복잡성이 적당하게 증가한 채로 BEST_BODY 플래그를 가지고 아니면 BEST_BODY 플래그 없이 ROP들을 적절하게 처리할 수 있다. 이전 버전 및 가장 최신 버전의 서버들과의 통신을 위한 ROP들은 예를 들어 이하의 표에 기술된 특징들을 포함할 수도 있다:

[0090]	이전 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP	가장 최신 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP
ROP ID	SynchFolder	SynchFolder
새로운 파라미터들	n/a	<u>BEST_BODY</u> 플래그: 설정되면, 이메일 서버 컴포넌트는 최상 이메일 메시지 본문을 선택해서 이메일 클라이언트 컴포넌트에게 송신한다. 이메일 메시지 본문을 소정 표준 포맷으로 변환하는 것은 불필요하다.

[0091] 도 8a 내지 도 8c는 이메일 서버 컴포넌트와 이메일 클라이언트 컴포넌트 간의 이메일 메시지 집합을 전송하는 수개의 상이한 혼존 모드들을 도시한 것이다. 각각의 모드에 있어서, 각각의 이메일 메시지는 헤더 집합 및 본문 집합을 포함하는 명명된 속성들을 가지며, 수개의 이메일 메시지들이 폴더에 포함되어 있다. 도 8a는 전체 아이템 전송 모드를 도시한 것이다. 본 도면은 제1 이메일 메시지 헤더(801)가 전송되고 그 후 제1 이메일 메시지 본문(802)이 제2 이메일 메시지 헤더(803) 전에 전송되고 그 후 제2 이메일 메시지 본문(803)이 전송되는 방식으로 이메일 메시지 집합이 다 전송될 때까지 계속되는 것을 도시하고 있다. 도 8b는 헤더들이 먼저 전송되는 모드를 도시하고 있다. 본 모드에서는, 제1 이메일 메시지 헤더(805)가 전송되고 그 후 제2 이메일 메시지 헤더(806)가 전송되는 등 모든 이메일 메시지 헤더들이 다 전송된 후에 제1 이메일 메시지 본문(807)이 전송되고 그 후 제2 이메일 메시지 본문(808)이 전송되는 등 이러한 방식으로 이메일 메시지 집합이 모두 전송된다. 도 8c는 헤더들 만이 전송되는 모드를 도시한 것이다. 모드 명칭이 암시하는 바와 같이, 이메일 메시지 헤더들(809)만이 이메일 메시지 집합 전송 요청에 응답해서 전송된다. 이메일 메시지 본문들(810)은 추가의 명백한 요청에 응답해서만 전송될 것이다. 상기 모드들 중 임의의 모드에서, 전송 시퀀스는 예를 들어 특정 이메일 메시지 본문에 대해 보다 높은 우선 순위의 이메일 클라이언트 컴포넌트 요청에 의해 일시적으로 인터럽트될 수도 있다.

[0092] 이메일 폴더는 이메일 메시지 집합의 전송 요청에 대한 타깃의 일례이다. 그러나, 이메일 폴더는 이메일 메시지들 외에 데이터 객체들을 포함할 수도 있다. 상술된 바와 같이, 전송 모드들은 헤더들이 먼저 전송되는 모드 및 헤더들만이 전송되는 모드처럼 종종 이메일 메시지 헤더들 및 이메일 메시지 본문들과 관련해서 정의된다. 이러한 전송 모드들에서, 명명된 속성들의 헤더 집합 및/또는 명명된 속성들의 본문 집합이 잘 정의되지 않은 데이터 객체들을 전송하려는 시도는 프로토콜 실패를 야기할 수 있다. 본 발명의 한 양상은 명명된 속성들의 헤더 집합 및/또는 명명된 속성들의 본문 집합이 잘 정의되지 않았지만, 부분이 아닌 전체가 항상 전송될 수 있는 데이터 객체들을 제공함으로써 이러한 상황을 피한다. 본 실시예는 예로서 도 8d로 도시될 수 있다. 본 일례에서, 이메일 서버 컴포넌트와 이메일 클라이언트 컴포넌트 간의 전송이 헤더만이 전송되는 모드에서 발생될 수 있다. 따라서, 제1 이메일 메시지 헤더(811)가 전송된 다음에 데이터 객체(812)가 다음으로 전송될 후보가 된다. 명명된 속성들의 헤더 집합이 FAI와 같이 데이터 객체(812)에 대해 잘 정의되어 있지 않아서, 전체 데이터 객체가 전송된다. 전송될 다음 후보는 잘 정의된 명명된 속성들의 헤더 집합을 가지며(즉, 후보 데이터 객체는 명명된 속성들의 헤더 집합에 속하며 이메일 클라이언트 컴포넌트에 의해 명백하게 정의된 모든 명명된 속성들을 소유함) 따라서 이메일 메시지 헤더(813)만이 전송된다.

[0093] 본 발명의 양상을 구현하는 한가지 방법의 일례는 상술된 SynchFolder ROP와 같이 동기화 ROP에 포함될 수도 있는 IGNORE_MODE_ON_FAI와 같은 플래그를 사용하는 것이다. 이메일 서버 컴포넌트는 복잡성이 적당하게 증가한 채로 IGNORE_MODE_ON_FAI 플래그를 가지거나 가지지 않은 양 상황에서 ROP들을 적절하게 처리할 수 있다. ROP들은 이메일 서버 컴포넌트와 이메일 클라이언트 컴포넌트 간의 이메일 폴더의 복제를 달성하기 위해 이하의 표에 기술된 특징들을 포함할 수 있다:

[0094]

	이전 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP	가장 최신 버전의 서버들과 통신하기 위해 프로토콜에 의해 사용될 수 있는 ROP
ROP ID	SynchFolder	SynchFolder
새로운 파라미터들	n/a	<u>IGNORE_MODE_ON_FAI</u> 플래그: 설정되면, FAI와 같이 명명 속성들의 헤더 및/또는 본문의 정의된 집합을 갖지 않는 데이터 객체를 위해, 이메일 서버 컴포넌트는 유력한 전송 모드와 무관하게 전체 데이터 객체로 전송 요청에 응답할 수 있다.

[0095]

이메일 메시지들은 하나 이상의 이메일 네트워크 사용자들에게 통상 어드레스된다. 이메일 메시지는 이메일 서버 컴포넌트에 의해 저장이 허용된 것인지가 고려되어 전달될 수 있다. 이메일 네트워크는 수개의 이메일 서버 컴포넌트들을 가질 수도 있다. 통상, 이메일 네트워크 프로토콜은 이메일 네트워크 사용자가 새로운 메시지들을 검사해야만 하는 이메일 서버 컴포넌트들의 수를 제한하기 위한 소정의 기법을 갖는다. 일반적인 예는 특정 이메일 네트워크 사용자에게 어드레스된 이메일 메시지들을 사용자의 홈 서버라고 하는 특정 이메일 서버 컴포넌트가 수용할 것임을 규정하는 홈 서버 기법이다. 이러한 경우에, 이메일 클라이언트 컴포넌트는 예를 들어 새로운 이메일 메시지들을 정기적으로 검사하거나 새로운 이메일 메시지 통지를 등록할 때 홈 서버만을 고려하도록 구성될 수 있다.

[0096]

도 9는 간단한 홈 서버 기법의 일례도 복잡성을 가질 수 있음을 도시한 것이다. 도 9에 도시된 예에서, 특정 이메일 서버 컴포넌트(901)는 먼저 특정 이메일 네트워크 사용자를 위한 홈 서버로 지정된다. 시간이 지남에 따라, 사용자를 위해 지정된 홈 서버는 통상적으로 관리라는 이유로 다른 이메일 서버 컴포넌트들(903 및 905)로 변경된다. 이메일 서버 컴포넌트들(901, 903 및 905)은 예를 들어 물리적으로 다르거나 또는 논리적으로 다를 수 있거나 또는 다른 버전들일 수 있다. 이메일 클라이언트 컴포넌트(902)는 시간 T_0 로부터 시간 T_1 일 때까지 이메일 서버 컴포넌트(901)하고만 통신할 수 있으며, 그 후 이메일 클라이언트 컴포넌트(904)는 시간 T_2 일 때까지 이메일 서버 컴포넌트(903)하고만 통신할 수 있으며, 그 후 이메일 클라이언트 컴포넌트(906)는 이메일 서버 컴포넌트(905)하고만 통신할 수 있다. 이메일 클라이언트 컴포넌트들(902, 904 및 906)은 동일할 수도 있고 상이할 수도 있다. 이메일 서버 컴포넌트들(901 및 903)은 시간 T_2 이후에 존재할 수도 있고 존재하지 않을 수도 있다. 이러한 복잡성들은 후술되는 이메일 메시지 저장 복제와 특히 관련된다.

[0097]

이메일 메시지들은 예를 들어 공지된 데이터베이스 기술들을 사용해서 구현될 수 있는 명시 이메일 메시지 스토어의 이메일 서버 컴포넌트에 의해 저장될 수 있다. 이메일 서버 컴포넌트는 하나 이상의 메시지 스토어들을 가질 수 있다. 이메일 네트워크 사용자는 홈 메시지 스토어를 가질 수 있다. 홈 메시지 스토어들을 변경하는 것은 홈 서버들을 변경하는 것에 관해 기술된 것과 동일한 효과들을 가질 수 있다.

[0098]

몇몇 이메일 네트워크 프로토콜들은 이메일 클라이언트 컴포넌트 구역에 로컬인 기억 서비스에 이메일 메시지 스토어의 부분들을 복제하는 기능을 포함한다. 원격 이메일 메시지 스토어의 부분들을 로컬 이메일 기억 서비스에 복제함으로써 프로토콜 성능을 향상시킬 수 있으며, 예를 들어, 모든 새로운 이메일 메시지들을 뷰(view)하라는 명시 이메일 네트워크 사용자 요청에 앞서 상기 모든 새로운 이메일 메시지들을 로컬 이메일 기억 서비스에 복제함으로써 인식된 프로토콜 성능을 향상시킬 수도 있다. 이러한 복제는 또한 추가의 이메일 클라이언트 컴포넌트 기능들을 제공하며, 예를 들어, 이메일 네트워크 사용자가 네트워크 연결 인터럽션 중에 이메일 메시지들을 뷰할 수 있게 해준다.

[0099]

이메일 네트워크 환경에서, 단순한 복제는 신속하게 비효율적으로 될 수 있다. 예를 들어, 이메일 서버 컴포넌트가 특정 이메일 네트워크 사용자와 관련된 하나의 이메일 메시지를 가지며 그 메시지가 네트워크 사용자를 위해 이미 클라이언트 컴포넌트에 복제되어 있으며, 새로운 이메일 메시지가 그 이메일 네트워크 사용자를 위해 도착하는 경우, 단순한 복제 요청에 응답해서 2개의 이메일 메시지들이 송신되어야만 할 필요가 있다. 2개의 이메일 메시지들의 복제 후에 다른 새로운 이메일 메시지가 도착하면, 단순한 복제 요청에 응답해서 3개의 이메일 메시지들이 송신되어야만 하는 것 등이 요구된다. 몇몇 이메일 네트워크 프로토콜들은 이메일 메시지 스토어들의 중분식 복제에 대해 이러한 문제점을 완화하기 위해 제공되어 왔다. 중분식 복제시, 이전에 성공적인 중분식 복제 후에 발생한 이메일 메시지 스토어에 대한 변경만이 복제 요청에 응답해서 송신되어야만 하며, 예

를 들어, 최종의 성공적인 증분식 복제 이후로, 변경된 사항이 새로운 이메일 메시지가 도착한 것 뿐이라면, 증분식 복제 요청에 응답해서 새로운 이메일 메시지만이 송신될 필요가 있다.

[0100] 도 10은 증분식 복제를 위해 제공된 프로토콜의 보다 상세한 일례를 도시한 것이다. 이메일 메시지 스토어는 이메일 폴더들로 분할될 수 있다. 각각의 이메일 폴더는 서로 무관하게 복제될 수 있어서, 복제 프로세스에 대한 보다 미세한 제어가 가능하다. 본 일례에서, 증분식 복제 프로세스는 이메일 서버 컴포넌트(502)로부터 이메일 클라이언트 컴포넌트(501)로의 변경 뿐만 아니라 이메일 클라이언트 컴포넌트(501)로부터 이메일 서버 컴포넌트(502)로의 변경 지연을 포함하기 때문에, 이를 동기화라고 한다. 동기화 요청(1001)에 이어, SynchFolder ROP가 이메일 서버 컴포넌트(502)에 의해 처리된다. ROP는 folderID 파라미터(도시되지 않음) 및 stateblob₀ 파라미터를 포함한다. folderID 파라미터는 동기화 요청(1001)의 타깃인 이메일 폴더를 식별한다. stateblob₀ 파라미터는 이메일 서버 컴포넌트(502)가 최종 동기화 아래에 이메일 폴더에 대해 변경된 사항이 있다면 이를 판정할 수 있게 해주는 정보를 포함한다. 요청(1001)이 이메일 클라이언트 컴포넌트(501)에 의한 타깃 폴더에 대한 제1 동기화 요청을 나타내면, 이메일 서버 컴포넌트(502)는 이메일 메시지 스토어의 타깃 이메일 폴더가 빈 폴더에 비해 변경되었는지를 판정한다. 요청(1001)에 대한 응답(1002)시, 이메일 서버 컴포넌트(502)는 타깃 폴더에 추가된 임의의 이메일 메시지를 및/또는 다른 데이터 객체들 및 타깃 폴더로부터 삭제된 임의의 이메일 메시지를 및/또는 다른 데이터 객체들의 리스트를 포함하는 임의의 변경 사항들을 이메일 클라이언트 컴포넌트(501)에게 송신한다. 이메일 서버 컴포넌트(502)는 또한 동기화에 바로 이어 이메일 클라이언트 컴포넌트(501)에서 나타날 타깃 폴더의 상태를 나타내는 새로운 stateblob₁을 생성하고 상기 stateblob₁을 응답(1002)시 송신한다. 이메일 클라이언트 컴포넌트(501)가 요청(1001)에서와 같이 동일한 폴더에 대한 다음 동기화 요청(1003)을 송신할 때, 요청(1003)은 응답(1002)에 따라 리턴된 동일한 stateblob₁을 파라미터로 포함할 것이다. 상술된 바와 같이, 이메일 서버 컴포넌트(502)는 타깃 폴더에서 발생한 변경 사항들이 있다면 이를 판정해서 응답(1004)시 상기 변경 사항들을 새롭게 생성된 stateblob₂와 함께 다시 이메일 클라이언트 컴포넌트(501)에 송신하기 위해 stateblob₁에 포함된 정보를 사용할 것이다.

[0101] stateblob 데이터 객체의 크기가 크면, 이는 예를 들어, 모든 이메일 폴더 동기화 요청에 따라 이메일 서버 컴포넌트로부터 또는 이메일 서버 컴포넌트로 송신되기 때문에, 프로토콜 성능에 악영향을 줄 수 있다. 이메일 폴더 동기화에 대한 몇몇 이메일 네트워크 프로토콜에서, stateblob는 대부분이 이메일 클라이언트 컴포넌트가 본 이메일 메시지들에 대한 변경 사항들을 식별하는 메시지 changeID 데이터 객체 집합으로 구성될 수도 있다. 이메일 메시지 변경 사항은 변경된 이메일 메시지가 컴포넌트에 전송될 때 이메일 클라이언트 컴포넌트 및/또는 이메일 서버 컴포넌트가 이미 본 것이라고 할 수 있다.

[0102] 메시지 changeID 데이터 객체의 한 목적은 전체 이메일 네트워크의 문맥에서 이메일 메시지에 대한 변경 사항을 유일하게 식별하는 것일 수도 있다. 홈 서버 기법을 사용하는 이메일 네트워크에서, 사용자의 홈 서버는 메시지 changeID 데이터 객체를 이전에 보지 못한 이메일 메시지 변경 사항과 연관시킬 책임이 있을 수 있다. 예를 들어, 홈 서버는 serverID 데이터 객체 및 일련 번호를 포함하는 메시지 changeID 데이터 객체들을 사용할 수 있다. serverID 데이터 객체는 전역 고유 식별자들과 같이 공지된 기술들을 사용해서 전체 이메일 네트워크 문맥에서 이메일 서버 컴포넌트를 유일하게 식별할 수 있다. 이러한 식별자들 자체의 크기가 큰 경우, serverID 데이터 객체가 대신 이메일 서버 컴포넌트에 의해 유지되는 식별자 루프 테이블로의 색인일 수도 있다. 이메일 서버 컴포넌트가 이전에 보지 못한 이메일 메시지의 저장을 허용할 때마다 증가하는, 예를 들어 폭이 6 바이트인, 이메일 서버 컴포넌트 구역 내의 카운터로 일련 번호가 제공될 수 있다.

[0103] 편의상, 메시지 changeID 데이터 객체는 예를 들어 "S₁:1"로 표시될 수도 있는데, 'S₁'은 제1 이메일 서버 컴포넌트에 대한 serverID 데이터 객체를 나타내고 '1'은 일련 번호를 나타낸다. 메시지 changeID 데이터 객체 집합은 예를 들어 "S₁:1, S₁:2, S₁:3"으로 표시될 수 있으며, "S₁:1", "S₁:2" 및 "S₁:3"은 serverID S₁을 갖는 이메일 서버 컴포넌트에 의해 사용되는 연속 메시지 changeID 데이터 객체들이다.

[0104] stateblob가 대부분이 이메일 클라이언트 컴포넌트가 본 이메일 메시지 변경 사항들을 나타내는 메시지 changeID 데이터 객체 집합("Message Changes Seen" 집합)으로 구성되는 경우, 그 크기를 감소시키기 위해 집합을 인코드하는 몇몇 기술들이 개발되었다. 예를 들어, 집합 "S₁:1, S₁:2, S₁:3, S₁:4"는 "S₁:1-4"로 인코드될 수 있다. 또한, 이메일 서버 컴포넌트는 사용하는 일련 번호들이 항상 증가하고 있음을 보증할 수 있다. 이러한 경우에, 즉, 기능의 손실 없이 최소 및 최대 일련 번호들을 포함하는 범위로서, Message Changes Seen 집합, 예

를 들어, "S₁:1, S₁:3, S₁:5, S₁:7"이 "S₁:1-7"로 인코드될 수도 있다.

[0105] 도 9에 의해 도시된 시나리오에서, Message Changes Seen 집합은 현 홈 서버(예를 들어, S₃) 외에 이메일 서버 컴포넌트들(예를 들어, S₁, S₂)에 의해 생성된 메시지 changeID 데이터 객체들을 포함할 수도 있다. 현 홈 서버에 의해 생성된 메시지 changeID 데이터 객체는 원시 메시지 changeID라고 할 수 있고, 다른 이메일 서버 컴포넌트들에 의해 생성된 메시지 changeID 데이터 객체는 외부 메시지 changeID라고 할 수 있다. 이전 버전의 이메일 서버 컴포넌트들과 통신하기 위한 이메일 네트워크 프로토콜들은 이메일 서버 컴포넌트를 기초로 최소 및 최대 일련 번호들을 포함하는 범위로서 인접하지 않은 외부 메시지 changeID 시퀀스들의 최적화에 대비해서는 제공되지 못했다. 이하의 표는 본 발명의 실시예에서 이러한 최적화를 포함하는 이점을 설명한 것이다:

[0106]		이전 버전의 서버가 사용하는 최적화(현 홈 서버 S ₃)	가장 최신 버전의 서버가 사용하는 최적화(현 홈 서버 S ₃)
최적화전 Message Changes Seen 집합	S ₁ :1, S ₁ :3, S ₁ :5, S ₁ :7 S ₂ :1, S ₂ :3, S ₂ :5, S ₂ :7 S ₃ :1, S ₃ :3, S ₃ :5, S ₃ :7		
최적화후 Message Changes Seen 집합	S ₁ :1, S ₁ :3, S ₁ :5, S ₁ :7 S ₂ :1, S ₂ :3, S ₂ :5, S ₂ :7 S ₃ :1-7	S ₁ :1-7 S ₂ :1-7 S ₃ :1-7	

[0107] 본 발명의 실시예는 이메일 서버 컴포넌트와 이메일 클라이언트 컴포넌트 간의 이메일 폴더의 동기화를 달성하기 위해 이하의 표에 기술된 특징들을 포함하는 ROP들을 사용한다. 이메일 서버 컴포넌트는 복잡성을 조금만 증가시키고도 향상된 stateblob 인코딩 기술을 구현할 수 있다.

[0108]		이전 버전의 서버들과 통신할 때 프로토콜에 의해 사용될 수도 있는 ROP	가장 최신 버전의 서버들과 통신할 때 프로토콜에 의해 사용될 수도 있는 ROP
	ROP ID	SynchFolder	SynchFolder
	새로운 모드에서 사용되는 변경되지 않은 파라미터들	stateblob: 외부 메시지 changeID 데이터 객체들의 비인접 집합을 포함하지 않는 최적화	stateblob: 외부 메시지 changeID 데이터 객체들의 비인접 집합을 포함하는 향상된 최적화

[0109] 도 11a 및 도 11b는 SynchFolder ROP에 응답하기 위해 이전 버전의 서버 및 가장 최신 버전의 서버에 의해 각각 사용될 수 있는 서브프로시저들 간의 차이점을 도시한 것이다. 도 11a는 단계들(1101, 1102 및 1103)을 도시하고 있다. 단계(1101)에서, 초기 Message Changes Seen 집합이 생성된다. 단계(1102)에서, 원시 메시지 changeID 데이터 객체들인 Message Changes Seen 집합의 멤버들이 최적화된다. 단계(1103)에서, 최적화된 Message Changes Seen 집합이 동기화를 요청한 이메일 클라이언트 컴포넌트에 대한 응답으로 송신될 수도 있는 stateblob 데이터 객체에 추가된다. 도 11b는 추가 단계(1104)를 포함하는데, 이는 외부 메시지 changeID 데이터 객체들인 Message Changes Seen 집합의 멤버들이 Message Changes Seen 집합이 단계(1103)에서 stateblob 데이터 객체에 추가되기 전에 최적화되어 최적화가 향상됨을 보여 준다.

[0110] 이메일 메시지 스토어를 이메일 폴더들로 분할하는 것은 동기화 프로세스에 대한 보다 미세한 제어를 제공하지만, 자동적으로 프로토콜 성능을 향상시키는 것은 아니며, 프로토콜 성능이 오히려 떨어지게 할 수도 있다. 예를 들어, 몇몇 프로토콜들은 각각의 메시지 스토어 폴더가 개별적으로 동기화될 것을 요구한다. 각각의 동기화 오퍼레이션은 통상 소정의 오버헤드를 갖는데, 상기 오버헤드는 중요한 것일 수 있다. stateblob 데이터 객체들을 사용하는 동기화 오퍼레이션은 유효 오버헤드를 가질 수도 있는 오퍼레이션들의 일례이다. 전체 메시지 스토어를 동기화하는 경우에, 각각의 메시지 스토어 폴더가 개별적으로 동기화될 것을 요구한 프로토콜들은 보다 적은 수의 동기화 오퍼레이션들을 요구하는 프로토콜들에 비해 불리할 수 있다.

[0111] 전체 메시지 스토어를 동기화하고 동기화를 유지하는 것은 이메일 클라이언트 컴포넌트를 위한 바람직한 목표이다. 종래 기술의 이메일 클라이언트 컴포넌트들은 프로토콜 성능에 상당한 악영향을 야기했을 때 조차도 상기 목표를 달성하고자 하였다. 본 발명의 한 양상은 심층 계층표(deep hierarchy table)를 사용해서 상기 목표를 달성하는 동안 프로토콜의 악영향을 최소화할 수 있다는 점이다. 종래 기술의 이메일 서버 컴포넌트는 심층 계

총표를 제공할 수 없었다.

[0112] 이메일 메시지 스토어가 이메일 폴더들로 분할되는 경우, 이메일 폴더들은 계층들로 조직화될 수 있다. 도 12는 이메일 폴더 계층의 일례를 도시한 것이다. 도 12에서, 폴더(1204)는 폴더(1203)의 서브폴더이다. 또한, 폴더(1203)는 폴더(1202)의 서브폴더이다. 폴더(1201)는 루트 폴더이다. 루트 폴더는 임의의 다른 폴더의 서브폴더가 아니다. 모든 다른 폴더들은 루트가 폴더(1201)인 폴더 계층의 멤버들이다. 통상, 폴더 계층의 각각의 폴더는 임의의 다른 폴더를 직접 참조하지 않는다. 폴더는 서브폴더들에 대해서만 직접 참조할 수 있다. 폴더는 또한 서브폴더인 임의의 폴더들만 직접 참조할 수 있다. 여러 경우, 계층의 루트 폴더만이 임의의 모든 폴더들을 직접 참조 할 수 있다.

[0113] 심층 계층표는 폴더 계층의 모든 폴더에 대한 정보를 포함할 수 있다. 각각의 폴더는 심층 계층표의 행을 가질 수 있다. 따라서 심층 계층표의 정보는 특정 시간 동안 이메일 폴더의 콘텐츠가 변경되었는지를 결정하기 위해서 사용될 수도 있다. 특정 시간 동안의 이메일 폴더에 대한 변경 결정은 특정 시간의 시초에 취해진 폴더의 행의 복사본을 시간의 끝에서 취해진 폴더의 행의 복사본과 간단히 비교함으로써 구현될 수 있다. 한 실시예에서, 심층 계층표의 각각의 행은 이하의 속성들을 포함한다:

속성명	속성 타입	주
Folder ID	FID	FID 타입은 전역 고유 식별자 (GUID) 및 6 바이트 일련 번호를 포함한다. 상기 값은 이메일 네트워크 문맥에서 이메일 폴더를 유일하게 식별하는데 사용될 수 있다.
PR_LOCAL_COMMIT_TIME_MAX	Timestamp	timestamp는 폴더의 콘텐츠가 변경되는 때마다 갱신된다.
PR_DELETED_COUNT_TOTAL	QWORD	상기 값은 폴더로부터 삭제된 아이템들의 총 수를 카운트한 값이다.

[0115] 심층 계층표의 이메일 폴더의 행의 속성들은 폴더의 콘텐츠가 변경될 때마다 갱신될 수 있다. 심층 계층표 갱신을 효율적으로 구현하기 위해, 출원인들은 심층 계층표에 대한 신속하고 직접적인 참조가 도움이 된다는 사실을 알았다. 심층 계층표에 액세스하고자 할 때 예측가능한 적은 수의 간접 레벨들이 있어야만 함을 알았다. 예를 들어, 폴더 계층의 임의의 레벨에 심층 계층표를 배치하는 것은 예측 가능한 수의 간접 레벨들에 대비하지 못한다. 본 발명의 한 실시예에서, 이러한 이유로 심층 계층표는 이메일 네트워크 사용자의 이메일 메시지 스토어 폴더 계층의 루트 폴더와 연관될 수 있다.

[0116] 이메일 클라이언트 컴포넌트와 이메일 서버 컴포넌트 간의 통신은 통신 세션들로 분할될 수 있다. 세션들 간에, 예를 들어, 네트워크 연결 방해 중에, 이메일 메시지 스토어 동기화의 손실이 발생할 수 있다. 통신 세션 시초에 이메일 메시지 스토어 동기화를 재확립하기 위해, 이전 버전의 이메일 서버 컴포넌트들과 통신하기 위한 몇몇 프로토콜들은 폴더 계층의 각각의 폴더에 대한 SynchFolder ROP를 사용했다. 통상, 폴더들 중 몇몇 폴더들의 콘텐츠는 세션들 사이에서 변경되지 않을 것이다. 타깃으로서 변경되지 않은 폴더를 갖는 SynchFolder ROP는 "null synch"를 야기한다. "null synch"가 이메일 클라이언트 컴포넌트에게 전송중인 임의의 폴더의 변경을 야기하지 않더라도, 예를 들어, 중요할 수도 있는 stateblob 데이터 객체와 관련된 오버헤드를 여전히 갖는다.

[0117] 도 13은 심층 계층표를 사용해서 "null synch" 결과들을 방지하는 본 발명의 실시예를 도시한 것이다. 제1 요청(1301)에서, 이메일 클라이언트 컴포넌트(501)는 심층 계층표를 요청하는 ROP(예를 들어, GetHierarchyTable)을 이메일 서버 컴포넌트(502)에게 송신한다. 제1 응답(1302)에서, 심층 계층표의 복사본이 이메일 클라이언트 컴포넌트(501)에게 제공된다. 통상, 이메일 클라이언트 컴포넌트(501)는 심층 계층표의 이전 복사본을 가질 것이다. 이메일 클라이언트 컴포넌트(501)는 2개의 복사본들의 행끼리 비교해서 이메일 서버 컴포넌트(502)의 사용자 이메일 메시지 스토어의 어떤 폴더들이 변경되었는지를 신속하게 결정할 수 있다. 그 다음, ROP들(예를 들어, SynchFolder)이 변경된 상기 폴더들만을 동기화하는데 사용된다. 요청(1303) 및 응답(1304)은 변경된 폴더들을 동기화하기 위해 필요할 때 반복될 수도 있다. 성공적인 동기화에 이어, 이메일 클라이언트 컴포넌트의 심층 계층표의 복사본은 응답(1302)시 송신된 최신 복사본과의 일치를 위해 갱신될 수도 있다. 이메일 클라이언트 컴포넌트(501)가 심층 계층표의 이전 복사본을 갖고 있지 않으면, 최신 복사본의 행을 가진 모든 폴더들이

동기화될 수도 있다.

[0118] 사용자의 이메일 메시지 스토어의 동기화가 일단 확립되면, 동기화는 상술된 세션 단계들의 개시를 정기적으로 반복함으로써(즉, 이메일 서버 컴포넌트를 폴링함으로써) 유지될 수 있지만, 이러한 방법은 단점들을 갖는다. 예를 들어, 폴링 기간이 사용자의 이메일 메시지 스토어에 대한 변경들 사이 기간 보다 더 짧을 수도 있다. 이러한 경우에, 비교적 다수의 심층 계층표 비교들이 폴더들이 변경되지 않았음을 나타낼 것이다. 이러한 비교는 사실상 비효율적이며, 따라서 이를 방지할 수 있는 프로토콜이 보다 효율적일 수 있다.

[0119] 몇몇 이메일 네트워크들은 예를 들어 특정 이메일 폴더의 콘텐츠가 변경될 때 이메일 서버 컴포넌트가 통지해주도록 이메일 클라이언트 컴포넌트가 가입하는 서비스를 포함한다. 몇몇 이전 버전의 이메일 클라이언트 컴포넌트들은 사용자의 폴더 계층의 각각의 폴더와 연관된 변경 통지들에 대한 개별적인 가입(subscription)을 생성함으로써 상기 서비스를 사용해서 사용자의 이메일 메시지 스토어의 동기화를 유지한다. 본 발명의 실시예에서, 이메일 클라이언트 컴포넌트는 심층 계층표와 연관된 변경 통지에 대한 단일 가입만을 생성할 수도 있다. 단일 가입을 확립하는데 보다 적은 수의 ROP들이 필요하고 보다 적은 서버측 리소스들이 소비되기 때문에 단일 가입은 보다 효율적이다.

[0120] 도 13을 더 참조해 보면, 본 발명의 한 양상에 따라, 가장 최신 버전의 이메일 클라이언트 컴포넌트(501)가 이메일 서버 컴포넌트(502)와의 통신 세션의 초기에 제1 요청(1301)시 GetHierarchyTable ROP를 사용할 때, 이메일 클라이언트 컴포넌트(501)는 응답(1302)시 리턴되는 심층 계층표와 연관된 변경 통지들에 대해 자동으로 가입된다. 이메일 클라이언트 컴포넌트에서 사용자의 이메일 메시지 스토어의 이메일 폴더에 대한 변경이 발생할 때, 예를 들어, 이메일 메시지가 폴더에 추가될 때, 심층 계층표는 상술된 바와 같이 갱신된다. 심층 계층표에 대한 변경은 이메일 클라이언트 컴포넌트(501)로의 통지 경고(1305)를 트리거한다. 통지 경고가 요청(1301)에 의한 가입에 대한 응답이지만, 명시적인 요청-응답 사이클의 부분은 아니다. 따라서, 본 발명에 의해 제공된 통지 시스템을 사용해서 이메일 네트워크의 오버헤드를 상당히 줄일 수 있다.

[0121] 단일 가입은 다수의 통지들을 야기할 수도 있다. 한 실시예에서, 경고가 비연결성 네트워크 전송 메카니즘, 예를 들어, UDP/IP(User Datagram Protocol/Internet Protocol)를 사용해서 전달되지만, 임의의 적합한 네트워크 전송 메카니즘이 사용될 수도 있다. 경고에 응답해서, 이메일 클라이언트 컴포넌트(501)는 ROP(예를 들어, GetNotification)를 포함하는 요청(1306)을 이메일 서버 컴포넌트(502)에게 송신한다. 응답(1307)시, 심층 계층표의 임의의 변경된 행들(즉, 통지를 트리거한 변경 폴더에 대응하는 행들)이 이메일 클라이언트 컴포넌트(501)에게 송신된다. 이메일 클라이언트 컴포넌트(501)는 변경된 폴더들만을 동기화하기 위해 ROP들(예를 들어, SynchFolder)을 사용한다.

[0122] 다수의 이메일 클라이언트 컴포넌트들은 예를 들어 협력 기능(collaborative functionality)을 제공하기 위해 동일한 데이터 객체와 연관된 변경 통지들에 대해 가입될 수 있다. 도 18에 도시된 바와 같이, 이메일 클라이언트 컴포넌트들(1801, 1802 및 1803)은 이메일 서버 컴포넌트(1804)에 위치한 동일한 데이터 객체(도시되지 않음)와 연관된 변경 통지들에 대해 가입되어 있다. 이메일 클라이언트 컴포넌트(1803)는 데이터 객체에 대한 변경을 야기하는 이메일 서버 컴포넌트(1804)에게 ROP(1805)를 송신한다. 변경의 결과로서, 이메일 서버 컴포넌트(1804)는 변경 통지들(1806, 1807 및 1808)을 이메일 클라이언트 컴포넌트들(1801, 1802 및 1803)에 송신한다. 변경 통지는 단지 변경된 데이터 객체를 식별하는 것외에는 별다른 정보가 없을 수도 있다. 따라서, 예를 들어, 이메일 클라이언트 컴포넌트가 그것이 특정 변경의 원인이라는 것을 판정할 방도가 없을 것이다. 데이터 객체가 예를 들어 이메일 폴더이면, 변경 통지들(1806, 1807 및 1808)은 각각의 이메일 클라이언트 컴포넌트(1801, 1802 및 1803)가 변경된 폴더에 대해 동기화를 개시하게 할 수도 있다. 본 일례에서, 이메일 클라이언트 컴포넌트(1803)가 변경에 대한 책임이 있으므로, 그 결과는 "null synch"가 될 것이다.

[0123] 상술된 이유로, "null synch"를 야기하는 동기화를 제거하는 것이 바람직할 수 있다. 그러나, 기술된 통지 동작이 항상 바람직하지 않은 것이 아닐 수도 있으며 몇몇 이메일 클라이언트 컴포넌트들은 통지 동작에 의존할 수도 있다. 본 발명의 한 양상은 프로토콜 성능을 향상시키기 위해 가장 최신 버전의 이메일 서버 컴포넌트들의 통지 동작을 구성할 수 있는 이메일 클라이언트 컴포넌트의 기능을 제공하면서, 동시에 이전 버전의 이메일 클라이언트 컴포넌트들에게 변경되지 않은 통지 동작을 제공하는 것이다.

[0124] 도 19a는 이전 버전의 이메일 서버 컴포넌트들에 의해 제공될 수 있는 통지 동작을 도시한 것이다. 도 19b는 본 발명의 한 양상에 따라 구성 가능한 통지 동작을 도시한 것이다. 요구되면, 가장 최신의 이메일 클라이언트 컴포넌트는 도 19b에 도시된 일례에서 예를 들어 요청을 가진 플래그 IGNORE_OWN 플래그를 제공함으로써 도 19b의 통지 동작이 가능함을 이메일 서버 컴포넌트에게 나타낼 수도 있다.

[0125] 단계(1901)에서, 가입자 집합에서 통지될 다음 후보가 선택된다. 단계(1904)에서, IGNORE_OWN 플래그에 대한 가입이 검사된다. 플래그가 존재하지 않으면, 단계(1904)는 단계(1902)로 분기하여, 통지가 후보 가입자에게 송신된다. 플래그가 발견되면, 단계(1904)는 단계(1905)로 분기하여, 상기 통지를 가입자가 트리거했는지를 결정하기 위해 가입이 다시 검사된다. 상기 결정은, 예를 들어, 가입을 위해 사용된 세션의 통신 세션 식별자("세션 ID")를 검사함으로써 달성될 수 있다. 예를 들어, 세션 ID는 전역 고유 식별자 및 6 바이트의 일련 번호를 포함한다. 통지는 또한 원인과 관련된 세션 ID에 대해 검사된다. 2개가 일치하면, 통지는 억제된다. 결과는 통지를 야기한 이메일 클라이언트 컴포넌트가 그 통지를 수신하지 않게 되는 것이다. 서브프로시저는 그 후 후술되는 단계(1903)로 진행한다.

[0126] 가입자가 통지를 트리거하지 않았으면, 가입과 관련된 세션 ID는 통지 원인과 연관된 세션 ID와 동일하지 않으며, 단계(1905)는 단계(1902)로 분기하여 통지가 송신된다. 프로세스는 그 후 단계(1903)로 진행하여, 통지될 가입자가 더 있는지를 판정한다. 더 있으면, 서브프로시저는 단계(1901)로 리턴하고, 그렇지 않으면, 서브프로시저는 종료된다.

[0127] 상술된 바와 같이, 이메일 메시지들의 캐시 스토리지를 사용하는 이메일 클라이언트 컴포넌트는, 예를 들어 ROP를 통해, 로컬 클라이언트 데이터 스토어와 이메일 서버 컴포넌트에서 유효한 데이터 스토어 간의 메시지를 또는 다른 데이터 객체들의 동기화를 요청할 수도 있다. 이메일 클라이언트 컴포넌트는 서버 스토어로부터 클라이언트 스토어로 복사될 메시지들을 유사하게 요청할 수도 있다. 어떠한 경우에서든, 요청은 고속 전송 모드를 사용해서 달성될 수 있다.

[0128] 통상, 메시지들 또는 파일과 같은 다른 데이터가 동기화 또는 복사되도록 요청될 때, 요청(예를 들어, ROP)은 동기화가 요구되는 모든 메시지들에 대한 표시를 포함한다. 이 리스트는 예를 들어 상술된 stateblob 특징을 사용함으로써 이메일 서버 컴포넌트에 의해 자동으로 구성될 수도 있다. 이전 버전(종래 기술)의 이메일 서버 컴포넌트들의 경우, ROP 요청의 하나의 메시지 또는 데이터 객체의 에러는 요청시 모든 아이템들의 실패를 야기했다. 이러한 프로세스는 도 14a에 도시되어 있으며, 여기서 ROP(예를 들어, FXPrepare)를 포함하는 요청은 복사 또는 동기화를 위해 지정된 messageID 집합과 함께 전송된다. 고속 전송 메카니즘이 이메일 서버 컴포넌트(502)에서 셋업되며, 고속 전송 ID는 단계(1402)에서 이메일 클라이언트 컴포넌트(501)에게 전송된다. 이메일 클라이언트 컴포넌트(501)는 그 후 예를 들어 FXGetBuffer ROP(단계(1403))를 포함하는 요청을 통해 데이터 객체들의 복사 또는 동기화를 요청한다. 이메일 서버 컴포넌트(502)가 요청된 메시지들의 개방을 시도할 때, 하나 또는 그 이상의 메시지들 또는 데이터 객체들에 대해 에러가 발생한다. 에러의 예로는 파괴되는 메시지 또는 데이터 객체, 서버 고장, 이메일 서버 컴포넌트(502)의 메모리 부족, 또는 데이터 객체에 대한 바이러스 검출을 포함한다.

[0129] 에러 후, 이메일 서버 컴포넌트(502)는 단계(1404)에서 이메일 클라이언트 컴포넌트(501)에 스트리밍된 데이터의 치명적인 ROP 에러를 송신한다. 이에 따라, 동기화가 실패하고, messageID 집합 내의 메시지들이 동기화나 복사되지 못하고, stateblob 또는 유사한 갱신 정보가 이메일 클라이언트 컴포넌트(501)에 의해 수신되지 않는다. 이메일 클라이언트 컴포넌트(501)는 다른 시간에 데이터 객체들의 동기화 또는 복사를 요청해야만 한다. 이메일 서버 컴포넌트(502)에서 에러가 고정되지 않으면, 에러 메시지들이 계속해서 송신될 수도 있고, messageID 집합 내의 메시지들이 결코 동기화되거나 복사되지 못할 수도 있다.

[0130] 본 발명의 한 양상에 따라, 치명적인 ROP 에러 대신, 가장 최신의 이메일 서버 컴포넌트는 특정 데이터 객체(예를 들어, 이메일 메시지)에 대한 에러 정보를 송신해서, 상기 데이터 객체만에 대한 동기화가 실패하게 할 수도 있다. 이러한 특징은 에러를 갖는 메시지 또는 다른 데이터 객체가 응답 내에 포함되더라도 ROP 내의 메시지들 또는 다른 데이터 객체들 또는 다른 요청이 송신되어 동기화 또는 복사될 수 있게 해준다.

[0131] 객체-특정 에러(object-specific error)를 처리하는 방법의 일례로서, 가장 최신의 이메일 서버 컴포넌트는 객체 에러를 갖는 데이터 객체에 대한 데이터 스트림에서 에러 메시지를 송신할 수도 있다. 본 예에서, 편의상, 에러를 FXErrorInfo라고 한다. 요구되면, 후술된 바와 같이, FXErrorInfo는 에러를 갖는 데이터 객체에 대한 messageID와 같은 정보 및 메시지가 실패한 이유에 관한 추가 정보를 가질 수 있다.

[0132] 도 14b는 메시지 M₃에서 에러가 발생하는 동기화를 도시한 것이다. 에러는 메시지 M₁, 메시지 M₂, 그 다음 FXErrorInfo, 및 그 다음에 메시지 M₄를 포함하는 FXGetBuffer 응답(1405)을 넣는다. FXErrorInfo 정보는 이메일 클라이언트 컴포넌트(501)가 어떤 메시지가 에러를 갖는지를 알게 하고 응답 내의 모든 다른 메시지들을 동기화할 수 있게 해 준다. 에러 메시지 FXErrorInfo가 에러 원인에 관한 정보를 포함하면, 정보는 예를 들어 에

러 메시지를 사용자에게 디스플레이함으로써 클라이언트 컴포넌트에 의해서 작용될 수 있다.

[0133] 이하의 표는 FXErrorInfo가 취할 수 있는 포맷의 일례를 도시한 것이다:

[0134]

FXErrorInfo		
속성명	속성 타입	주
버전	WORD	본 구조의 버전
에러 코드	DWORD	
MessageID	MID	MID 타입은 전역 고유 식별자 (GUID) 및 6 바이트 일련 번호를 포함한다. 이는 에러를 야기한 메시지의 messageID이다.
...	...	0개의 또는 보다 많은 수의 속성 들이 추가될 수 있다.
보조 필드 크기	ULONG	배열 크기
보조 필드	BYTE array	에러 세부사항들을 전달하기 위한 구조화되지 않은 배열

[0135] 상기 표에서 알 수 있는 바와 같이, 일례의 포맷은 버전 속성, 에러 코드 및 messageID를 포함한다. 또한, 요구되면, 하나 이상의 속성이 추가될 수도 있다. 또한, 상술된 바와 같이, 보조 필드가 에러 세부 사항을 전달하기 위해 정의될 수도 있다. 이에 따라, 속성은 에러 세부 사항들의 필드 크기(예를 들어, 배열)를 규정하기 위해 정의될 수도 있으며, 필드가 예를 들어 에러 세부 사항을 전달하기 위한 구조화되지 않은 배열로 제공될 수도 있다. 상술된 바와 같이, 에러 세부 사항들은 이메일 클라이언트 컴포넌트(501)가 원하는 대로 처리될 수 있다.

[0136] FXErrorInfo는 제1 응답의 동기화가 완료되어, 예를 들어, stateblob 또는 다른 정보가 이메일 클라이언트 컴포넌트(501)에게 제공될 수 있게 해준다. 이메일 클라이언트 컴포넌트는 이제 메시지 M_4 를 통해 동기화 되기 때문에, 다음 동기화에 대한 요청(1406)은 M_4 후의 메시지들(예를 들어, M_5 및 M_6)을 갖는 응답(1407)을 야기할 수도 있다.

[0137] 이메일 클라이언트 컴포넌트(501)가 가장 최신 버전이고, 따라서 FXErrorInfo 메시지를 처리할 수 있음을 표시하기 위해, 예를 들어, 동기화 또는 복사를 요청하는 ROP와 함께 송신될 수 있는 FXRecoverMode 플래그가 정의될 수 있다. FXErrorInfo 메시지를 처리할 수 있는 이메일 서버 컴포넌트(502)에게 전달되도록 이메일 클라이언트 컴포넌트(501)에 대해 다른 표시들이 사용될 수도 있다.

[0138] 이메일 서버 컴포넌트(502)가 하나 이상의 메시지를 또는 다른 데이터 객체들을 이메일 클라이언트 컴포넌트(501)에 송신할 때, 이메일 클라이언트 컴포넌트로의 데이터 스트림은 속성 태그들(예를 들어, ptags)에 의해 분리되거나 정의될 수 있다. 예를 들어, 메시지 리스트는 각각의 메시지에 대해 개시 메시지 ptag 및 종료 메시지 ptag를 포함할 수도 있다. 개시 ptag 및 종료 ptag 사이에 스트링 속성을 갖는 속성 리스트 ptag 및 주제 ptag가 있을 수도 있다. 주제 ptag가 주제 자체의 다음에 이어질 수 있다. 다른 속성 태그들이 포함될 수도 있다.

[0139] 메시지를 송신할 때 에러가 발생하는 경우에, FXErrorInfo가 ptag로서 제공될 수 있으며, 상기 표에 의해 정의된 바와 같이 2진 속성들을 가질 수도 있다. 성공적인 메시지 및 에러가 발생한 메시지를 모두 갖는 데이터 스트림의 일례가 후술된다. 에러가 발생한 경우에, 종료 메시지 ptag는 이 특정 메시지에 대해 사용되지 않고 ptag FXErrorInfo가 상기 메시지에 대한 최종 ptag이다.

```

ptagMessageListStart
  ptagMessageStart
    ptagPropList
      ptagSubject [PT_STRING]
      "Re: Your email"

  ...

ptagMessageEnd
ptagMessageStart

  ...

ptagFXErrorInfo [PT_BINARY]
  [Contents as described by table]
ptagMessageStart

  ...

ptagMessageEnd
ptagMessageListEnd

```

[0140]

[0141]

도 15a는 이메일 서버 컴포넌트(502)가 이전 버전의 이메일 클라이언트 컴포넌트(501)에게 메시지들을 전송하기 위해 사용할 수 있는 단계들을 도시한 것이다. 단계(1501)에서 시작해서, 예를 들어, 메시지 집합을 최종 전송 데이터 스토어에 배치함으로써, 메시지 집합이 준비된다. 단계(1502)에서, 메시지는 예를 들어 이메일 서버 컴포넌트(502)의 송신 버퍼에 배치된 직후에 스트리밍을 시작한다. 메시지를 스트리밍할 때 에러가 발생하면, 치명적인 ROP 에러가 단계(1504)에서 이메일 클라이언트 컴포넌트(501)에게 스트리밍된다. 서브프로시저는 그 후 종료된다. 메시지를 스트리밍할 때 에러가 발생하지 않으면, 단계(1503)에서 집합에 메시지들이 더 있는지가 결정된다. 더 있으면, 프로세스는 단계(1502)로 돌아가서, 다음 메시지가 스트리밍된다. 그렇지 않으면, 서브프로시저는 종료된다.

[0142]

도 15b는 이메일 서버 컴포넌트(502)의 가장 최신 버전에 의해 메시지 집합을 처리하기 위한 프로시저를 도시한 것이다. 단계들은 이메일 클라이언트 컴포넌트가 가장 최신 버전이거나 이전 버전이거나에 따라 상이하다. 단계들(1501~1504)은 이전 버전의 이메일 클라이언트 컴포넌트에 대한 것이고, 선행절과 동일한 참조 부호들을 갖는 단계들은 동일하다.

[0143]

단계(1502)에서, 메시지를 스트리밍할 때 에러가 발생하면, 단계(1505)에서 요청이 FXRecoverMode와 같은 플래그를 포함하는지가 결정된다. 요청이 플래그를 포함하면, 이메일 클라이언트 컴포넌트(501)는 가장 최신 버전이고, 단계(1505)는 단계(1506)로 분기하여, FXErrorInfo가 이메일 클라이언트 컴포넌트(501)에게 스트리밍된다. 프로세스는 그 후 단계(1503)로 진행한다. 요청이 플래그를 포함하지 않으면, 단계(1505)는 단계(1504)로 분기하여, 치명적인 ROP 에러가 스트리밍된다. 그 후, 서브프로시저는 종료된다.

[0144]

보는 바와 같이, 요청시 플래그의 존재는 고장 및 치명적인 ROP 에러 송신을 대신해서 FXErrorInfo를 스트리밍하게 함으로써 스트리밍 프로세스가 계속되게 해준다. 플래그가 이메일 클라이언트 컴포넌트(501)의 가장 최신 버전에 의해 송신된다. 이메일 클라이언트 컴포넌트들의 이전 버전들은 플래그를 포함하지 않으며, 따라서 상술된 바와 같이 에러는 치명적인 ROP 에러 스트리밍을 야기한다.

[0145]

요구되면, 다른 실시예에서, 에러 메시지(예를 들어, FXErrorInfo)는 전체 메시지 대신 메시지 또는 다른 데이터 객체의 특정 속성들에 대해 송신될 수 있다. 예를 들어, FXErrorInfo는 메시지 본문 또는 메시지에 첨부된 사항에 대해서 송신될 수도 있다. 이메일 클라이언트 컴포넌트(501)는 그 후 에러 없이 성공적으로 송신된 속성들을 동기화 또는 복사할 수 있으며, 에러들을 갖는 속성들만이 동기화되거나 복사되지 않는다.

[0146]

때때로, 메시지 또는 다른 데이터 객체는 다수의 FXGetBuffer 응답들에 걸치는 충분한 크기일 수도 있다. 이러한 메시지들을 처리하기 위해, 이메일 클라이언트 컴포넌트(501)는 롤백(rollback) 로직을 포함함으로써 임의의

부분적으로 수신된 메시지를 처리한 후 여러 메시지 수신 후의 다른 메시지들을 계속해서 적합하게 수신할 수 있다.

[0147] 경우에 따라서는 이메일 메시지들과 같은 데이터 객체들의 복사 또는 동기화 진행에 관한 피드백을 이메일 클라이언트 컴포넌트에게 제공하는 것이 바람직할 수도 있다. 본 발명의 한 양상에 따라, 이메일 클라이언트 컴포넌트(501)의 가장 최신 버전은 예를 들어 데이터 객체들의 동기화 또는 복사를 요청할 때 PROGRESS_MODE와 같은 플래그를 이메일 서버 컴포넌트(502)에게 송신함으로써 진행 모드들을 처리할 수 있음을 나타낼 수도 있다. 응답시, 이메일 서버 컴포넌트(502)의 가장 최신 버전은 모든 메시지들의 총 크기, 메시지들의 총 수, 및 각각의 메시지들의 총 크기, 또는 상기 정보들 중 임의의 한 정보 또는 그 결합과 같은 다양한 정보를 메시지들과 함께 송신할 수도 있다.

[0148] 예를 들어, 도 16a에 도시된 바와 같이, 이전 버전의 이메일 클라이언트 컴포넌트(501)에 있어서, 메시지 집합에 대한 고속 전송 요청(1601 및 1603)에 대한 응답시, 이메일 클라이언트 컴포넌트(501)는 메시지들을 수신한다. 도 16a에서, 2번의 응답들(1604 및 1606)에서 메시지들이 수신된다. 고속 전송 메카니즘을 사용하는 이전 버전의 이메일 클라이언트 컴포넌트들(501)에서, 클라이언트에게 스트리밍되는 메시지들의 진행 표시는 제공되지 않았다.

[0149] 그러나, 도 16b에 도시된 바와 같이, 이메일 클라이언트 컴포넌트에 의한 메시지 집합 요청에 대한 응답(1607) 시, 이메일 서버 컴포넌트(502)는 전송되는 데이터 객체들의 총 수 및 전송되는 모든 데이터 객체들의 총 크기를 제공할 수도 있다. 상기 정보는 도 16b에서 "P_{all}"로 표시된다. 이메일 서버 컴포넌트(502)의 가장 최신 버전은 도 16b에서 "P₁, P₂, P₃, ..."으로 표시된 각각의 메시지의 크기를 제공할 수도 있다. 또한, 요구되면, 각각의 메시지 및 메시지 전체 그룹과 관련된 정보가 각각의 메시지가 FAI인지 또는 실제 이메일 메시지인지에 관한 추가 정보를 포함할 수도 있다. 한 실시예에서, 도 16b에서 "P_{all}"로 표시된 정보는 제로 데이터 객체들이 전송되더라도 데이터 스트림의 프로세싱을 간단하게 하기 위해 항상 고속 전송 요청에 대한 응답시 송신된다.

[0150] 전송되는 모든 데이터 객체들의 크기 및 수에 대한 포맷의 일례는 이하의 표에 기재되어 있다.

IncrSyncProgressMode		
속성명	속성 타입	주
버전	WORD (예를 들어, 16 비트 정수)	본 구조의 버전
cAssocMsgs	DWORD (예를 들어, 32 비트 정수)	전송될 FAI 데이터 객체들의 수
lTotalAssocMsgSize	QWORD (예를 들어, 64 비트 정수)	전송될 FAI 데이터 객체들의 총 크기
cNormalMsgs	DWORD	전송될 이메일 메시지들의 수
lTotalNormalMsgSize	QWORD	전송될 모든 이메일 메시지들의 총 크기

[0152] 보는 바와 같이, FAI 데이터 객체들의 수, 모든 FAI 데이터 객체들의 총 크기, 전송되는 이메일 메시지들의 수, 및 전송되는 모든 이메일 메시지들의 총 크기에 대한 개별적인 속성들이 정의될 수 있다. 다른 결합들 및 추가의 속성들이 요구되면 포맷에 추가될 수도 있다.

[0153] 이하의 표는 각각의 메시지와 함께 제공될 수도 있는 크기 및 다른 정보에 대한 포맷을 도시한 것이다.

IncrSyncProgressModePerMsg		
속성명	속성 타입	주
메시지 크기	LONG	다음 메시지의 크기
FAI 플래그	BOOL	다음 메시지가 FAI인지 나타냄

[0155] 보는 바와 같이, 포맷은 다음 메시지의 크기 및 다음 메시지가 FAI인지의 여부를 포함한다.

[0156] 도 17a 및 도 17b는 각각 이메일 컴포넌트들의 이전 버전 및 이메일 컴포넌트들의 가장 최신 버전에 따라 메시지 집합을 스트리밍하기 위한 단계들을 도시한 것이다. 도 17a의 단계들은 도 15a의 단계들(1501-1503)과 유사하다. 도 17b의 경우, 가장 최신 이메일 클라이언트 컴포넌트(501)에 의해 예를 들어 ROP와 함께

PROGRESS_MODE 플래그가 송신되었다. 메시지 집합이 단계(1701)에서 준비된 후에, 플래그가 존재하는지가 결정된다. 존재하면, 진행 데이터 토탈들이 단계(1702)에서 송신되고, 프로세스는 단계(1502)로 진행해서, 제1 메시지가 스트리밍된다. 플래그가 존재하지 않으면, 단계(1701)는 단계(1502)로 직접 분기한다.

[0157] 제1 메시지가 스트리밍된 후에, 프로세스는 단계(1703)로 진행해서, 플래그가 유효한지가 결정된다. 유효하면, 단계(1703)는 단계(1704)로 분기해서, 메시지 마다의 진행 데이터가 스트리밍된다. 프로세스는 그 후 상술된 단계(1503)로 진행한다. 플래그가 유효하지 않으면, 단계(1703)는 단계(1503)로 직접 분기한다.

[0158] 데이터를 가장 최신 클라이언트 컴포넌트에게 송신하는 가장 최신 서버 컴포넌트를 위한 데이터의 스트림의 일례가 후술된다. 데이터의 스트림은 상술된 데이터의 스트림과 유사하지만, 예를 들어 2진 속성들을 가질 수도 있는 진행 토탈 데이터(ptagIncrSyncProgressMode)에 대한 ptag들을 추가로 포함한다. 또한, 각각의 메시지에 있어서, 예를 들어, ptagIncrSyncProgressModePerMsg와 같은 메시지 마다의 진행 데이터가 제공된다.

```

PtagIncrSyncProgressMode [PT_BINARY]
    [Contents as described by table]
ptagMessageListStart
    PtagIncrSyncProgressModePerMsg [PT_BINARY]
        [Contents as described by table]
    ptagMessageStart
        ptagPropList
            ptagSubject [PT_STRING]
                "Re: Your email"

    ...
    ptagMessageEnd
    PtagIncrSyncProgressModePerMsg [PT_BINARY]
        [Contents as described by table]
    ptagMessageStart
    ...
    ptagMessageEnd
    PtagIncrSyncProgressModePerMsg [PT_BINARY]
        [Contents as described by table]
    ptagMessageStart
    ...
    ptagMessageEnd
ptagMessageListEnd

```

[0159] 도시된 일례에서, 진행 토탈 데이터(ptagIncrSyncProgressMode)를 포함하는 ptag들 및 메시지 진행 데이터(ptagIncrSyncProgressModePerMsg)에 대한 ptag들은 메시지 리스트 앞과 각각의 메시지 앞에, 각각 위치한다. 그러나, 데이터 객체들의 스트리밍의 구조는 진행 데이터가 메시지를 내에 또는 메시지 리스트 내에 포함될 수 있도록 교정될 수도 있다. 또한 메시지를 및/또는 메시지 리스트들을 전체적으로 제거하는 ptag들을 제거하기 위해 데이터 객체들의 스트리밍의 구조를 교정할 수 있다.

[0161] 진행 데이터를 수신하는 이메일 클라이언트 컴포넌트는 이메일 서버 컴포넌트로부터의 데이터 객체들의 동기화 또는 복사 진행을 결정하기 위해 상기 데이터를 사용할 수도 있으며, 각각의 개별 메시지의 진행을 결정하기 위해 메시지 마다의 진행 데이터를 사용할 수도 있다. 상기 정보는 예를 들어 동기화 진행에 대한 실시간 정보를 모니터할 때 도움이 될 수도 있다.

[0162] 이메일 메시지 또는 다른 데이터 객체들을 저장하기 위해 사용될 수도 있는 몇몇 상이한 문자 집합들이 있다.

예를 들어, ASCII가 영어 문자들을 기억하기 위해 가장 흔하게 사용된다. 그러나, ASCII는 8-비트 문자들에 기초하기 때문에 모든 언어들의 문자들을 저장하기에는 충분하지 않다. 따라서 ASCII 코드는 오직 256개의 문자들을 위해서만 사용될 수 있는데, 이는 영어에는 충분하지만 보다 많은 문자들을 갖는 언어들을 위해서는 충분하지가 않다. 한편, 유니코드는 각각의 문자에 대해 16 비트들(2 바이트)을 사용하는 문자 집합이고, 따라서, ASCII 보다 많은 수의 문자들을 포함할 수 있다. 유니코드는 65,536 문자들을 가질 수 있어서, 세계의 거의 모든 언어들을 인코드하는데 사용될 수 있다. 유니코드는 ASCII 문자 집합을 포함한다.

[0163] 일반적으로, 이메일 클라이언트 컴포넌트(501)의 이전 버전들은 지정된 코드 페이지, 또는 문자 집합 및/또는 연관된 언어를 갖는다. 예를 들어, 이메일 클라이언트 컴포넌트(501)의 특정 버전은 독일어 코드 페이지를 가질 수 있으며, 다른 버전은 ANSI 코드 페이지를 가질 수도 있다. 경우에 따라서 이메일 클라이언트 컴포넌트(501)는 지정된 코드 페이지 외의 문자 집합들로 된 이메일들을 수신하는 것이 바람직할 수도 있다. 본 발명의 한 양상에 따라, 가장 최신 클라이언트 컴포넌트는 이메일 서버 컴포넌트가 유니코드로 모든 이메일들을 제공하도록 할 수 있다. 일단 이메일들이 이메일 클라이언트 컴포넌트(501)에 의해 수신되면, 유니코드 이메일들은 클라이언트의 코드 페이지로 변환되거나, 또는 대안적으로 유니코드 포맷으로 유지될 수도 있다.

[0164] 이메일 클라이언트 컴포넌트(501)가 이메일들이 유니코드로 제공될 것을 요청함을 나타내기 위해, 이메일 클라이언트 컴포넌트(501)는 예를 들어 FORCEUNICODE와 같은 플래그를 이메일 서버 컴포넌트(502)에 제공할 수 있다. 이 플래그는 ROP와 같이 요청과 함께 제공될 수도 있다. 이메일 서버 컴포넌트(502)가 가장 최신 버전이면, 이메일 서버 컴포넌트(502)는 유효한 경우 이메일의 유니코드 버전을 제공할 수 있으며, 또는 다른 문자 집합들로 된 이메일 메시지들을 유니코드로 변환할 수 있다.

[0165] 도 20은 본 발명의 한 양상에 따라 메시지를 위해 특정 문자 집합을 제공하는 단계들을 도시한 것이다. 단계(2001)에서, 이메일 서버 컴포넌트(502)는 데이터 스토어로부터 메시지를 검색한다. 단계(2002)에서, FORCEUNICODE 플래그가 존재하는지를 결정한다. 존재하지 않으면, 단계(2002)는 단계(2003)로 분기해서, 이메일 서버 컴포넌트(502)가 필요하면 변환해서 이메일 클라이언트 컴포넌트의 지정된 코드 페이지의 이메일 메시지를 제공한다.

[0166] FORCEUNICODE 플래그가 존재하면, 단계(2002)는 단계(2004)로 분기해서, 메시지가 유니코드로 저장되어 있는지를 결정한다. 유니코드로 되어 있으면, 단계(2004)는 단계(2005)로 분기해서, 메시지는 유니코드 문자 집합으로 이메일 클라이언트 컴포넌트(501)에 제공된다. 메시지가 유니코드로 저장되어 있지 않으면, 단계(2004)는 단계(2006)로 진행해서, 메시지는 유니코드로 변환되고, 그 후 프로세스는 단계(2005)로 진행해서, 메시지가 유니코드로 이메일 클라이언트 컴포넌트에게 제공된다.

[0167] 본 명세서에 인용된, 공보, 특히 출원 및 특허들을 포함하는 모든 참고 문헌들은 각각의 참고 문헌이 참조용으로 인용된 것임을 개별적으로 명확하게 표시되었고 본 명세서에 온전히 그대로 기술된 것처럼 동일한 정도로 참조용으로 인용되어 있다.

[0168] 본 발명을 기술하는 문맥에서 (특히 이하의 청구항들의 문맥에서) 용어들 "하나의"("a" 및 "an") 및 "상기"("the") 및 유사한 지시 대상을 사용하는 것은, 본 명세서에서 달리 지시되거나 또는 문맥상 명백하게 부인되지 않는 한 단수 및 복수를 커버하는 것으로 해석될 것이다. 용어 "포함하는"("comprising", "having", "including" 및 "containing")은 달리 언급되지 않는 한 개방적인 용어로 해석될 것이다(즉, "포함하지만, 제한적인 것은 아님"("including, but not limited to")을 의미한다). 본 명세서에서 값들의 범위들을 기술한 것은 본 명세서에서 달리 지시되지 않는 한 범위 내에 속하는 각각의 개별 값에 개별적으로 연관시키는 단지 속기 방법으로서 작용하려는 의도이며, 각각의 개별 값은 본 명세서에 개별적으로 기술된 것처럼 명세서에 편입된다. 본 명세서에 기술된 모든 방법들은 본 명세서에서 달리 지시되거나 또는 문맥상 명백하게 부인되지 않는 한 임의의 적합한 순서대로 실행될 수 있다. 본 명세서에 제공된 임의의 일례들 및 모든 일례들 또는 일례의 언어(예를 들어, "~와 같은(such as)")를 사용한 것은 달리 청구되지 않는 한 단지 본 발명을 더 잘 설명하기 위한 것으로 본 발명의 범위를 제한하는 것은 아니다. 명세서의 어떤 언어도 임의의 청구되지 않은 요소가 본 발명의 실행에 필수적인 것으로 해석되어서는 안된다.

[0169] 본 발명을 실시하도록 발명자들에게 공지된 최상의 모드를 포함하는 본 발명의 양호한 실시예들이 본 명세서에 기술되어 있다. 상기 양호한 실시예들의 변화는 상술된 설명을 읽을 때 본 기술 분야에 숙련된 자들에게 명백해질 수 있다. 본 발명자들은 숙련된 자들이 이러한 변화를 적합하게 사용하기를 기대하며, 본 발명자들은 본 명세서에 명확하게 기술된 바와 다른 방법으로 본 발명이 실행되기를 의도한다. 따라서, 본 발명은 적용 법률에 의해 허용되는 대로 첨부된 청구항들에 기술된 주제의 모든 변경물 및 동등물을 포함한다. 또한, 본 명세서

에서 달리 지시되거나 또는 문맥상 명백하게 부인되지 않는 한 모든 가능한 변화의 상술된 요소들의 임의의 결합물도 본 발명에 포함된다.

발명의 효과

[0170] 본 발명은 향상된 클라이언트 및 서버 통신을 위한 시스템 및 방법을 제공한다. 특히, 본 발명은 클라이언트와 서버간 통신에 사용될 수도 있는 향상된 프로토콜에 관한 것이다. 본 발명은 특히 이메일 서버 환경과 관련된 것이지만, 본 명세서에 기술된 특징들은 다른 클라이언트 및 서버 네트워크에서도 사용될 수 있다.

[0171] 본 발명의 한 양상에 따라, 이메일 클라이언트 컴포넌트는 이메일 메시지에 유용한 최상의 메시지 본문을 수신하는 것에 관계하는 이메일 서버 컴포넌트에게 지시할 수 있다. 이메일 서버 컴포넌트는 메시지 요청을 수신할 수도 있는데 상기 요청은 메일의 최상 메시지 본문이 요구된 것임을 나타내는 표시를 갖는다. 이메일 서버 컴포넌트는 이메일 서버 컴포넌트와 관련된 데이터 스토어에 액세스할 수 있고 유효 메시지 본문의 포맷을 변환하는 것과 무관하게 유용한 메시지의 최상 메시지 본문을 결정할 수 있으며, 최상 메시지 본문의 포맷을 변환하지 않고 최상 메시지 본문을 검색 및 리턴할 수도 있다. 이처럼, 이메일 서버 컴포넌트에서 이메일 본문의 변환이 일어나지 않기 때문에, 이메일 서버 컴포넌트의 처리 시간이 감소된다.

[0172] 본 발명의 다른 양상에 따라, 특정 속성 또는 속성 집합(예를 들면, 헤더들)의 전송 요청시, 속성 또는 속성들이 데이터 객체 내에 정의되어 있지 않은 경우, 이메일 서버 컴포넌트는 전체 데이터 객체를 전송할 수도 있다. 이메일 클라이언트 컴포넌트는 폴더의 데이터 객체에 대한 요청을 생성하는데, 상기 요청은 데이터 객체의 적어도 하나의 속성이 요구된 것임을 나타내는 표시를 포함한다. 이메일 서버 컴포넌트는 요청을 수신하고, 폴더 및 폴더의 데이터 객체들에 액세스한다. 폴더의 각각의 데이터 객체에 있어서, 적어도 하나의 속성이 데이터 객체에서 정의되어 있으면, 이메일 서버 컴포넌트는 데이터 객체의 적어도 하나의 속성을 검색하고 이메일 클라이언트 컴포넌트에게 리턴한다. 데이터 객체에 대해 적어도 하나의 속성도 정의되지 않은 경우, 이메일 서버 컴포넌트는 데이터 객체를 검색하여 이메일 클라이언트 컴포넌트에게 리턴한다.

[0173] 본 발명의 또 다른 양상에 따라, 이메일 클라이언트 컴포넌트는 강제로 이메일 서버 컴포넌트가 이메일 메시지를 유니코드로 제공하게 한다. 이메일 클라이언트 컴포넌트는 적어도 하나의 이메일 메시지에 대한 요청 및 이메일 클라이언트 컴포넌트가 이메일 메시지들이 유니코드 포맷으로 되기를 요구한다는 표시를 송신한다. 이메일 서버 컴포넌트는 상기 요청 및 표시를 수신한 응답으로 적어도 하나의 메시지를 검색하고, 각각의 이메일 메시지에 대해, 이메일 메시지가 유니코드 포맷으로 이용가능하면, 유니코드 포맷을 이메일 클라이언트 컴포넌트에게 제공한다. 이메일 메시지가 유니코드 포맷으로 이용가능하지 않으면, 이메일 서버 컴포넌트는 이메일 메시지를 유니코드 포맷으로 변환하여 유니코드 포맷을 이메일 클라이언트 컴포넌트에게 제공한다.

[0174] 본 발명의 또 다른 양상에 따라, 이메일 클라이언트 컴포넌트에 의해 송신된 요청은 요청에 대한 응답 크기에는 제한이 없음을 나타낼 수도 있으며, 필요한 경우 이메일 서버 컴포넌트가 버퍼를 채울 수 있게 한다. 이메일 클라이언트 컴포넌트는 요청 내의 다수의 서브요청들을 송신하는데, 서브요청들의 각각은 이메일 서버 컴포넌트의 동작을 요청하며 크기 정보를 포함한다. 각각의 서브요청에 응답해서, 크기 정보가 이메일 서버 컴포넌트가 예상하는 범위 내의 크기 한계를 포함하면, 이메일 서버 컴포넌트는 크기 한계에 대한 응답을 제한한다. 크기 정보가 이메일 서버 컴포넌트가 예상한 범위 밖의 크기 한계를 포함하면, 이메일 서버 컴포넌트는 크기 정보에서 새로운 크기 한계를 찾는다. 새로운 크기 한계는 "유효 버퍼를 채우는" 것과 같이 임의적일 수도 있다.

도면의 간단한 설명

[0001] 도 1은 네트워크에 의해 접속된 컴퓨터들을 도시한 개략도.

[0002] 도 2는 본 발명의 실시예를 구현하는데 사용될 수 있는 일례의 컴퓨터 시스템을 도시한 개략도.

[0003] 도 3은 이메일 클라이언트 컴포넌트 및 이메일 서버 컴포넌트의 다수의 버전들을 갖는 환경을 도시한 개략도.

[0004] 도 4는 이메일 클라이언트 컴포넌트와 이메일 서버 컴포넌트 간의 프로토콜 협상 프로시저의 일례를 도시한 프로토콜 도면.

[0005] 도 5는 이메일 클라이언트 컴포넌트 및 이메일 서버 컴포넌트가 고정 크기 통신 버퍼들을 갖는 일례의 이메일 네트워크를 도시한 개략도.

[0006] 도 6a는 고속 전송 동작을 완료하기 위해 2개의 요청-응답 사이클들을 필요로 하는 일례의 프로토콜을 도시한

프로토콜 도면.

[0007] 도 6b는 고속 전송 동작을 완료하기 위해 단일 요청-응답 사이클을 필요로 하는 일례의 프로토콜을 도시한 프로토콜 도면.

[0008] 도 7a는 이메일 클라이언트 컴포넌트에 이메일 메시지 본문을 송신하기 위한 일례의 프로시저를 도시한 플로우챠트.

[0009] 도 7b는 본 발명의 한 양상에 따라 이메일 클라이언트 컴포넌트에 이메일 메시지 본문을 송신하기 위한 프로시저를 도시한 플로우챠트.

[0010] 도 8a는 전체 아이템 전송 모드를 도시한 시퀀스 도면.

[0011] 도 8b는 헤더가 먼저 전송되는 모드를 도시한 시퀀스 도면.

[0012] 도 8c는 헤더들만이 전송되는 모드를 도시한 시퀀스 도면.

[0013] 도 8d는 헤더가 먼저 전송되는 모드 또는 헤더들만이 전송되는 모드에 대한 예외를 도시한 시퀀스 도면.

[0014] 도 9는 시간에 따라 변경되는 이메일 클라이언트 컴포넌트의 홈 이메일 서버 컴포넌트를 도시한 개략도.

[0015] 도 10은 이메일 클라이언트 컴포넌트와 이메일 서버 컴포넌트 간의 이메일 폴더들을 동기화하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0016] 도 11a는 stateblob 파트를 최적화하기 위한 일례의 프로시저를 도시한 플로우챠트.

[0017] 도 11b는 본 발명에 따라 stateblob 파트를 최적화하기 위한 프로시저를 도시한 플로우챠트.

[0018] 도 12는 이메일 폴더 계층을 도시한 개략도.

[0019] 도 13은 본 발명의 한 양상에 따라 이메일 메시지 스토어를 동기화하고 동기화를 유지하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0020] 도 14a는 ROP 레벨에서 에러 정보를 전달하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0021] 도 14b는 본 발명의 한 양상에 따라 메시지 단위로 에러 정보를 전달하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0022] 도 15a는 ROP 레벨에서 에러 정보를 생성하기 위한 프로시저를 도시한 플로우챠트.

[0023] 도 15b는 본 발명의 한 양상에 따라 메시지 단위로 에러 정보를 생성하기 위한 프로시저를 도시한 플로우챠트.

[0024] 도 16a는 고속 전송 동작을 실행하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0025] 도 16b는 본 발명의 한 양상에 따라 고속 전송 동작을 실행하는 동안 진행 정보를 제공하기 위한 일례의 프로토콜을 도시한 프로토콜 도면.

[0026] 도 17a는 메시지 집합을 스트리밍하기 위한 프로시저를 도시한 플로우챠트.

[0027] 도 17b는 본 발명의 한 양상에 따라 진행 정보와 함께 메시지 집합을 스트리밍하기 위한 프로시저를 도시한 플로우챠트.

[0028] 도 18은 이메일 서버 컴포넌트에서의 동일한 데이터 객체에 대한 변경 결과를 통지받는 다수의 이메일 클라이언트 컴포넌트들의 개략도.

[0029] 도 19a는 다수의 가입자들에게 통지하기 위한 프로시저를 도시한 플로우챠트.

[0030] 도 19b는 본 발명의 한 양상에 따라 다수의 가입자들에게 통지하기 위한 프로시저를 도시한 플로우챠트.

[0031] 도 20은 본 발명의 한 양상에 따라 희망 코드 페이지를 사용하는 이메일 메시지를 제공하기 위한 프로시저를 도시한 플로우챠트.

[0032] <도면의 주요 부분에 대한 부호의 설명>

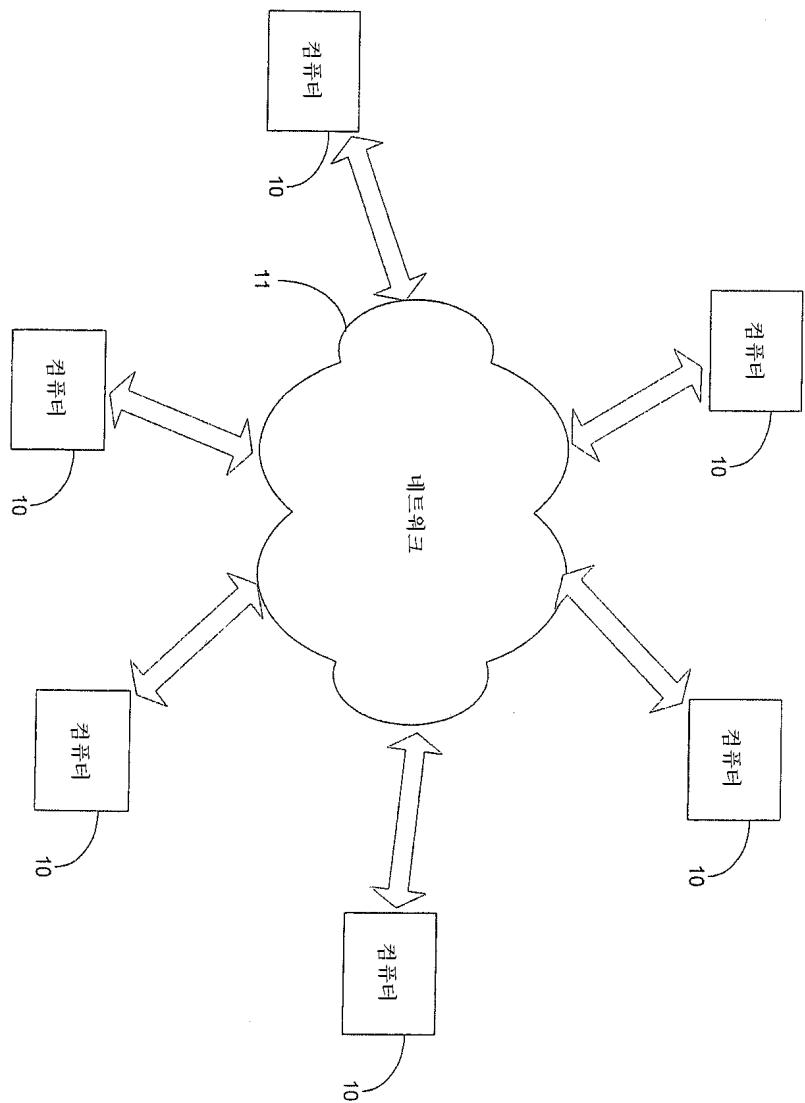
[0033] 10 : 컴퓨터

[0034] 11 : 네트워크

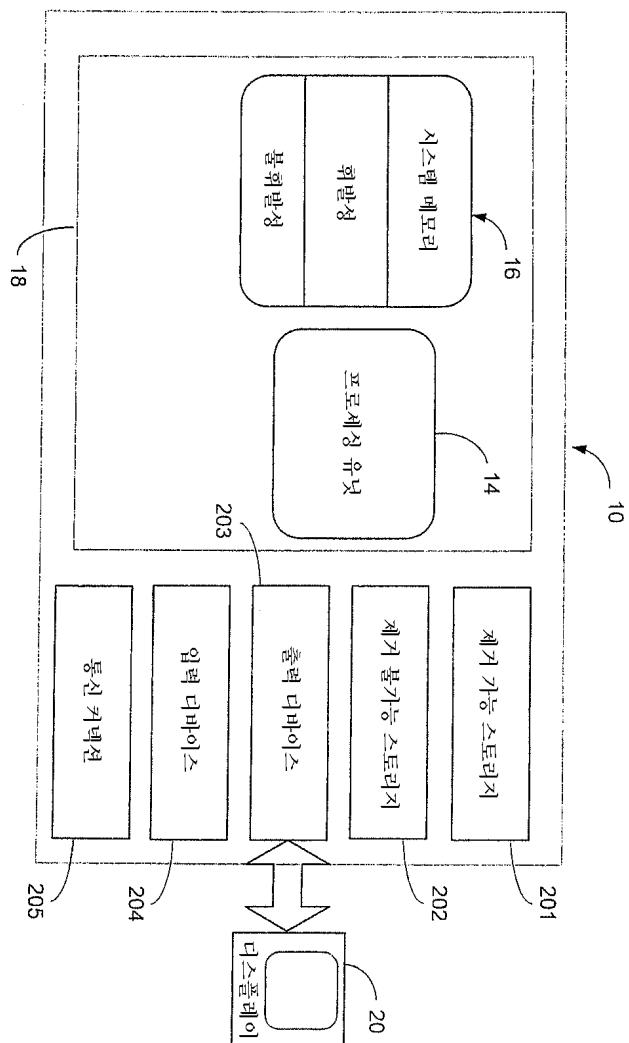
- [0035] 307, 308, 309, 310, 311 : 프로토콜
- [0036] 501 : 이메일 클라이언트 컴포넌트
- [0037] 502 : 이메일 서버 컴포넌트
- [0038] 503, 504, 505, 506 : 버퍼
- [0039] 1001, 1003 : 요청
- [0040] 1002, 1004 : 응답

도면

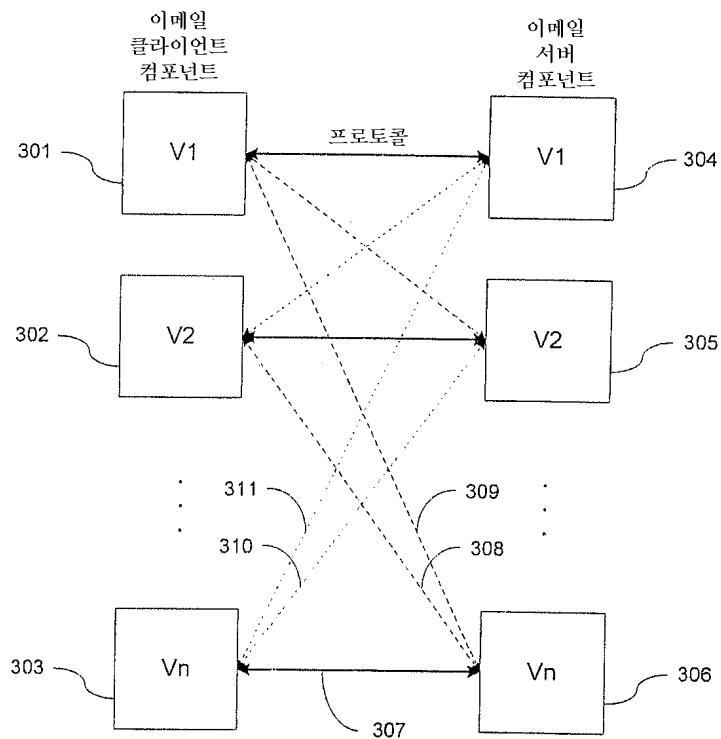
도면1



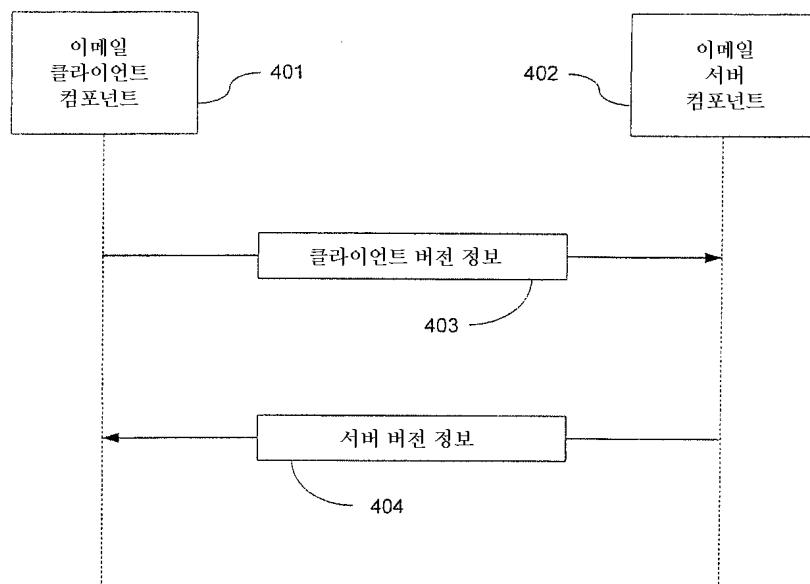
도면2



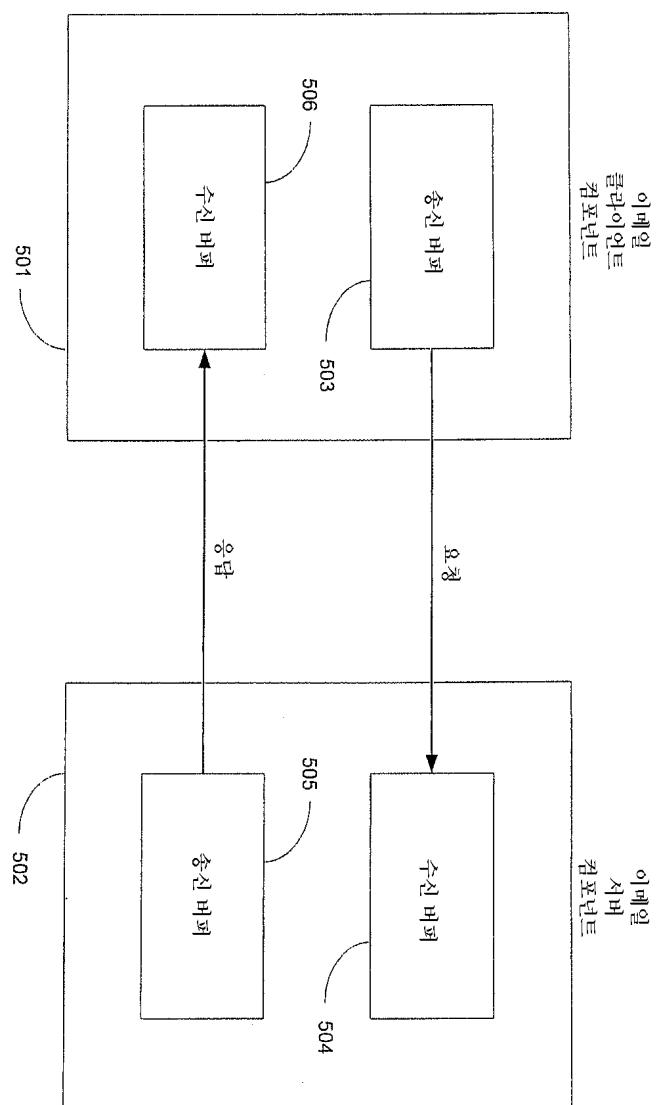
도면3



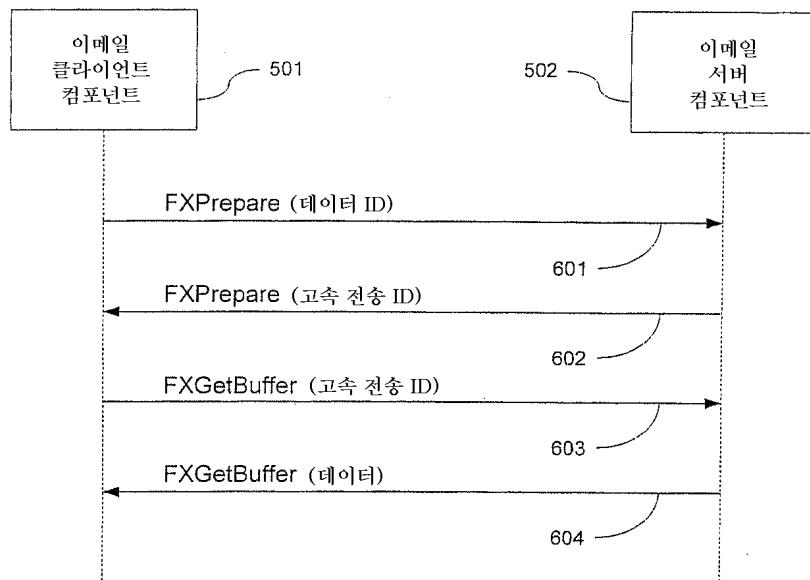
도면4



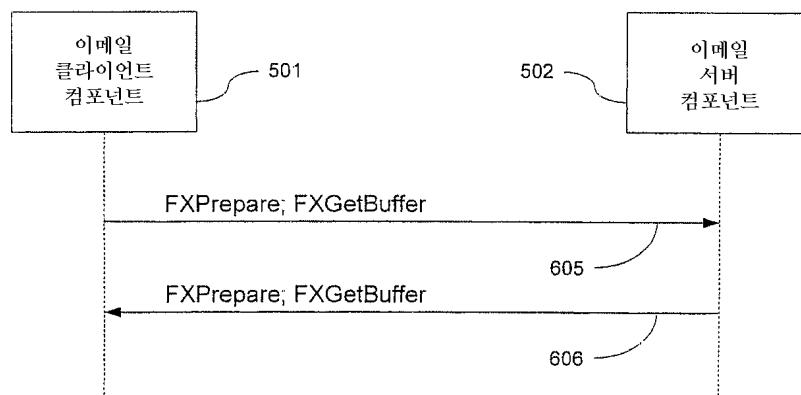
도면5



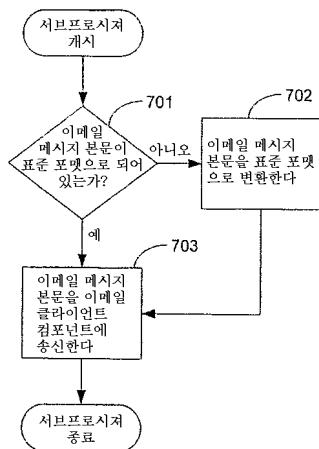
도면6a



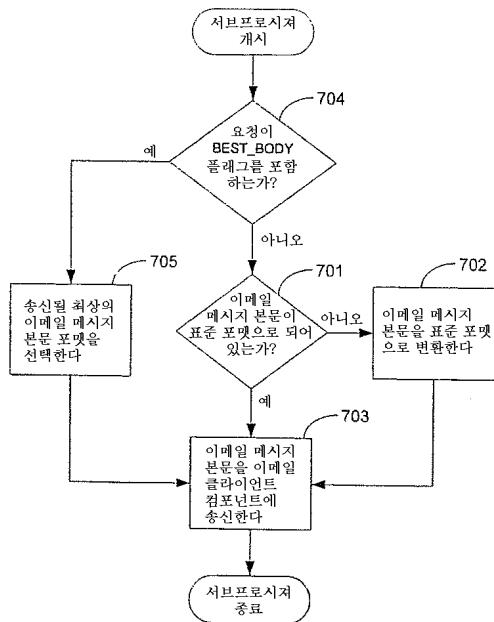
도면6b



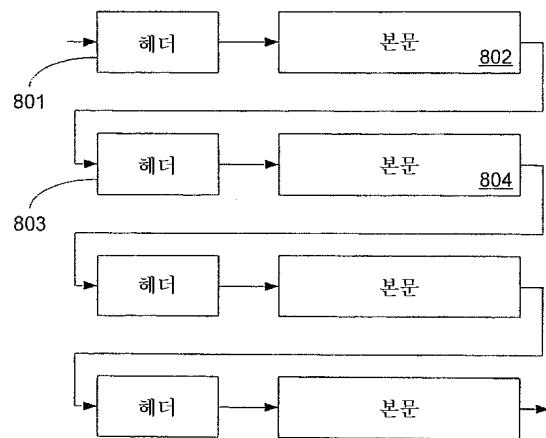
도면7a



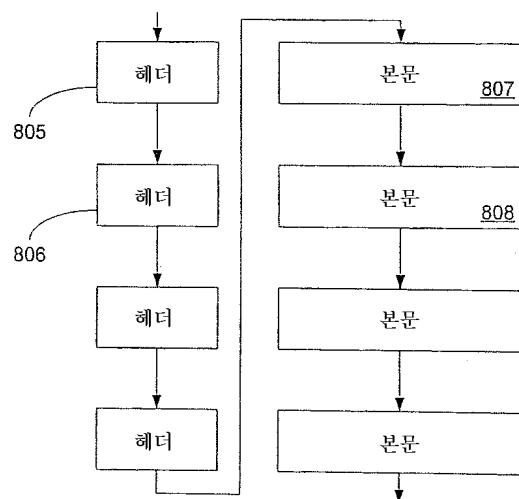
도면7b



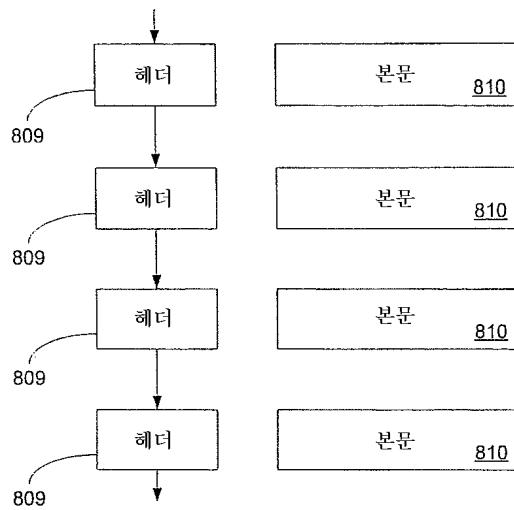
도면8a



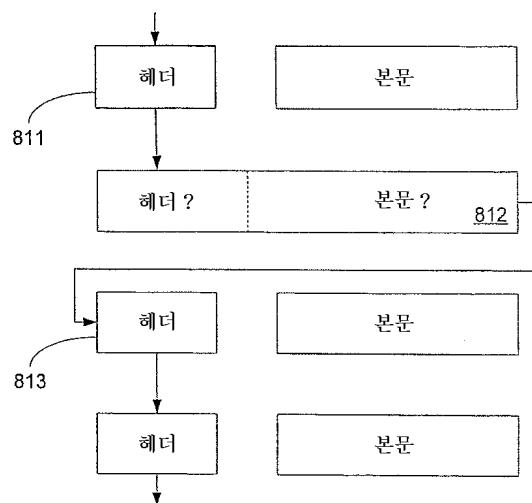
도면8b



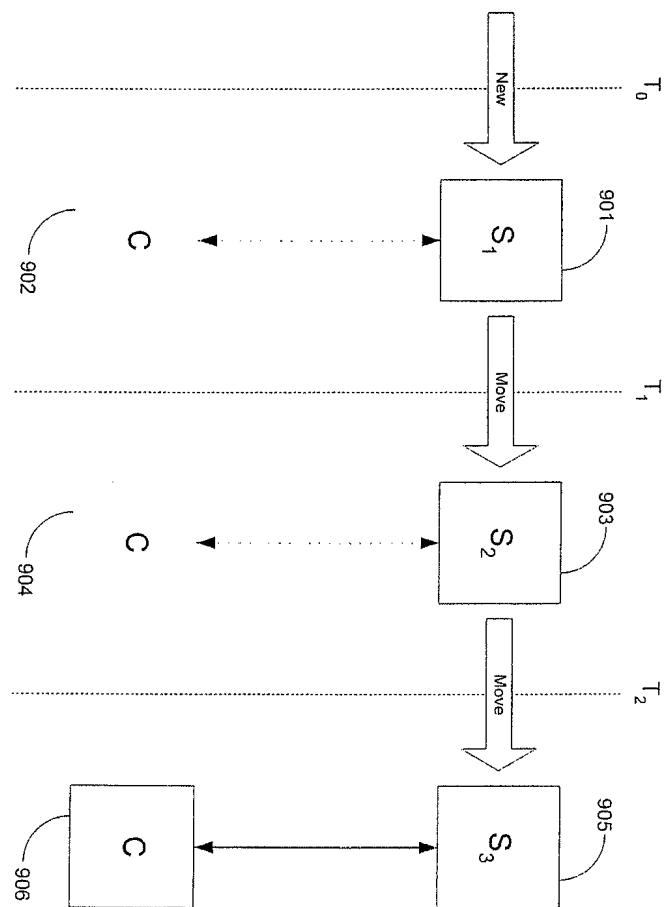
도면8c



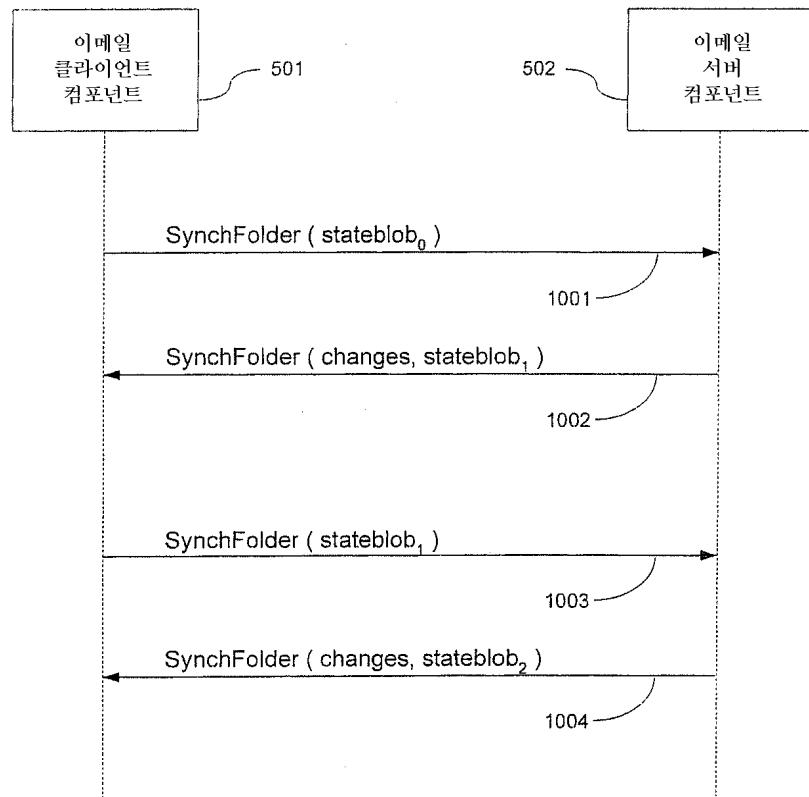
도면8d



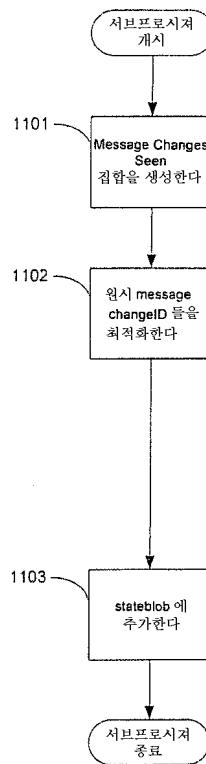
6면도



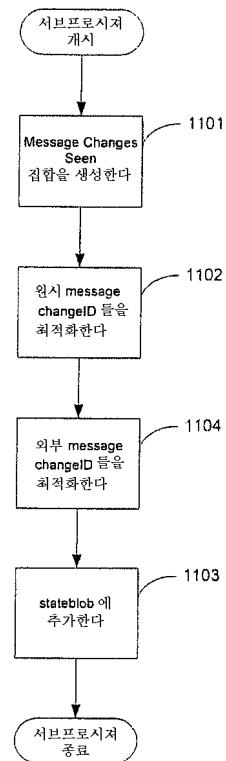
도면10



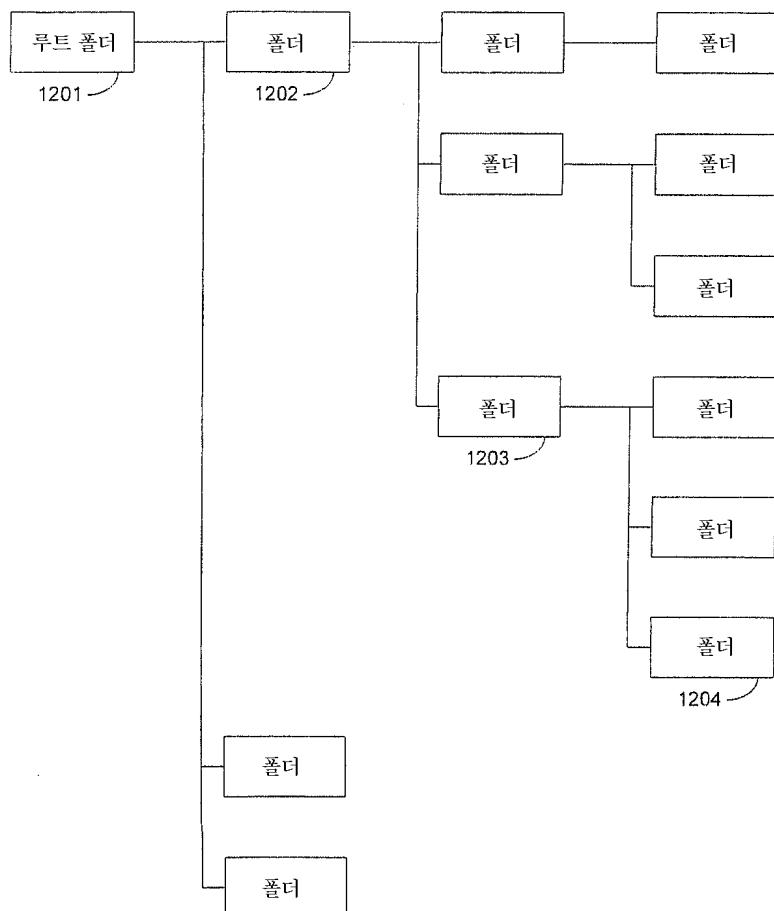
도면11a



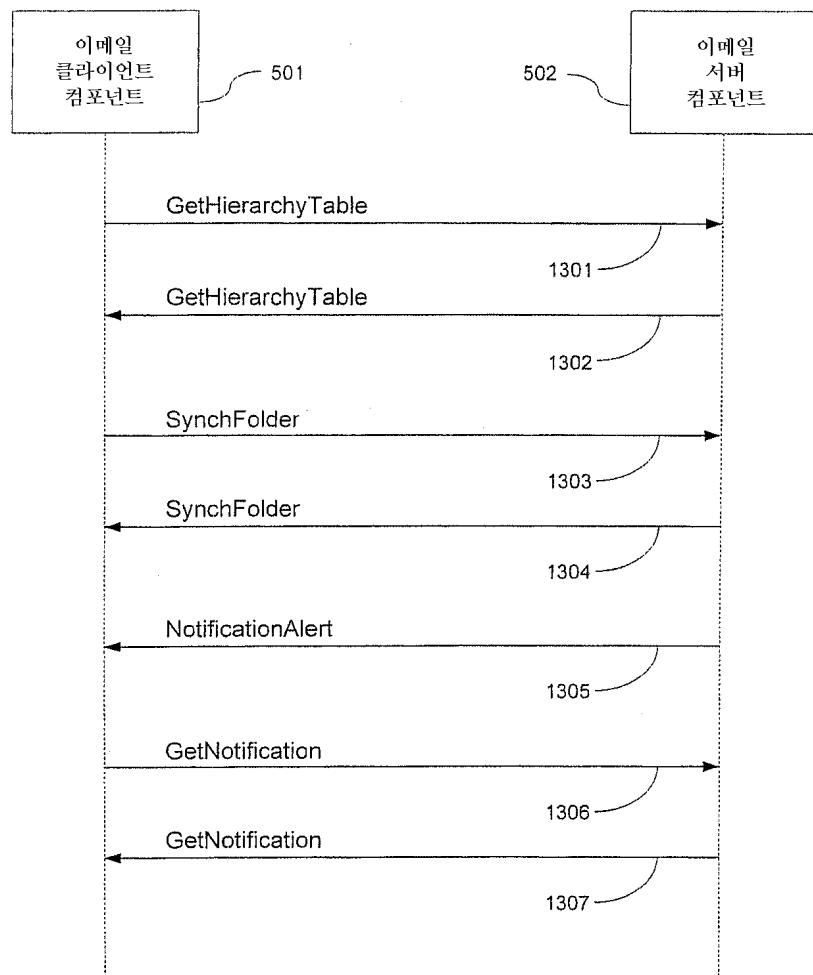
도면11b



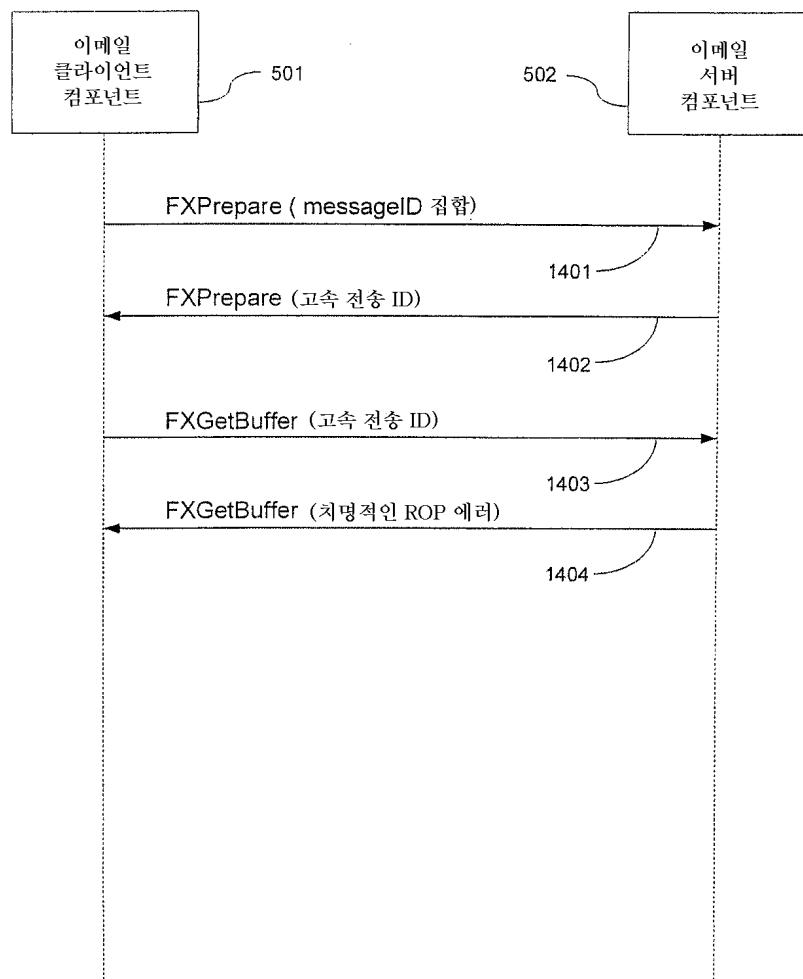
도면12



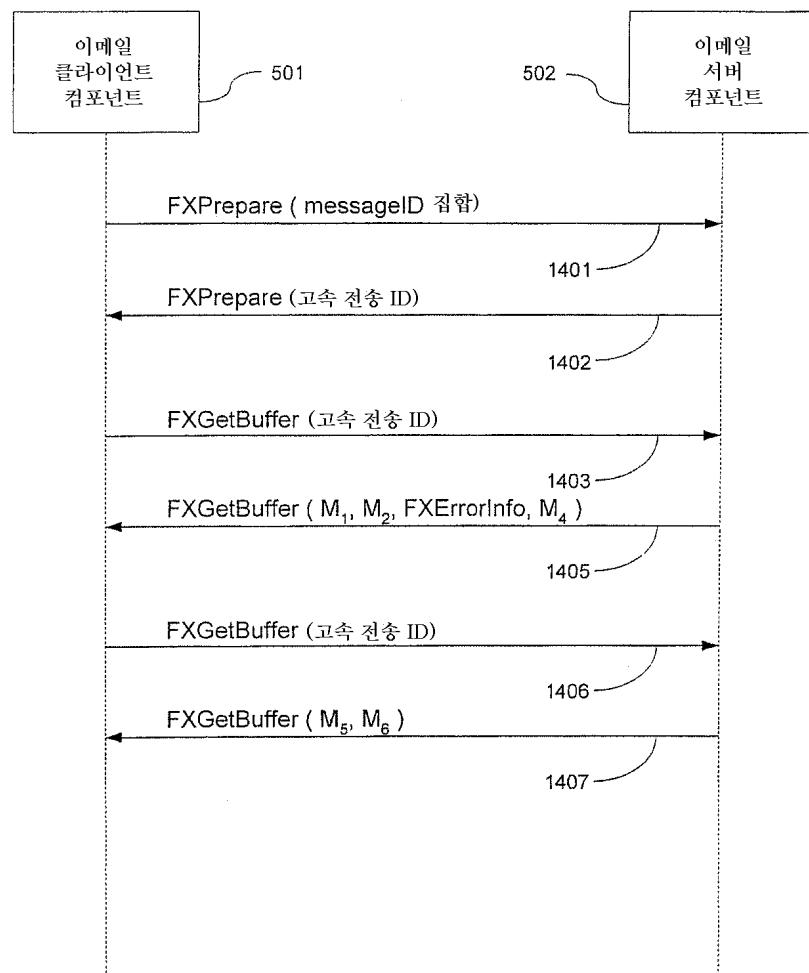
도면13



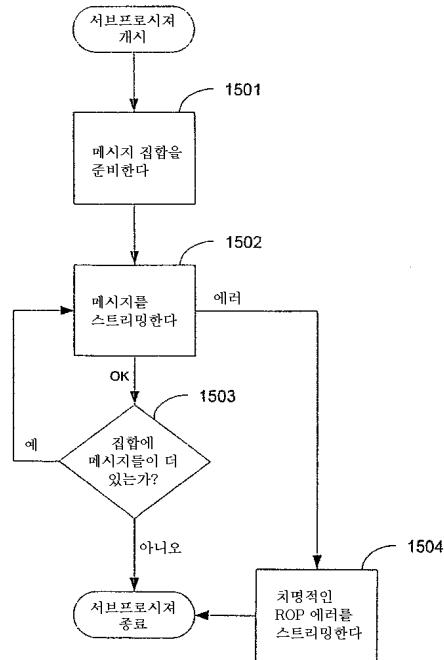
도면14a



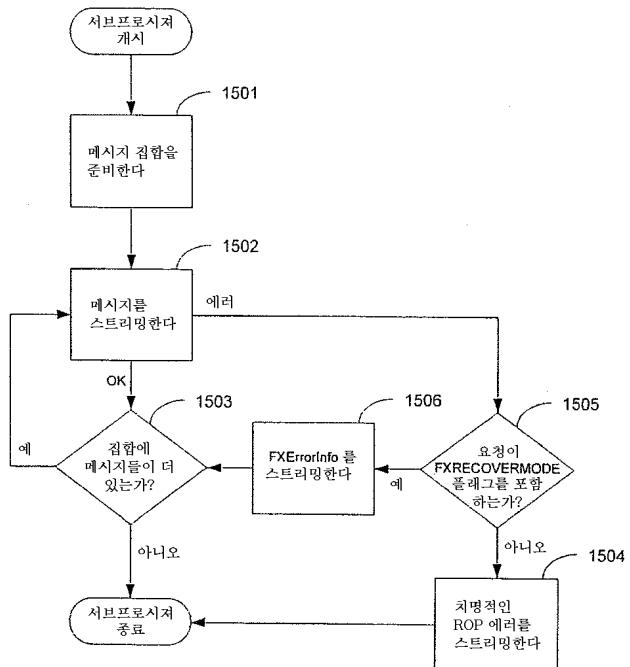
도면14b



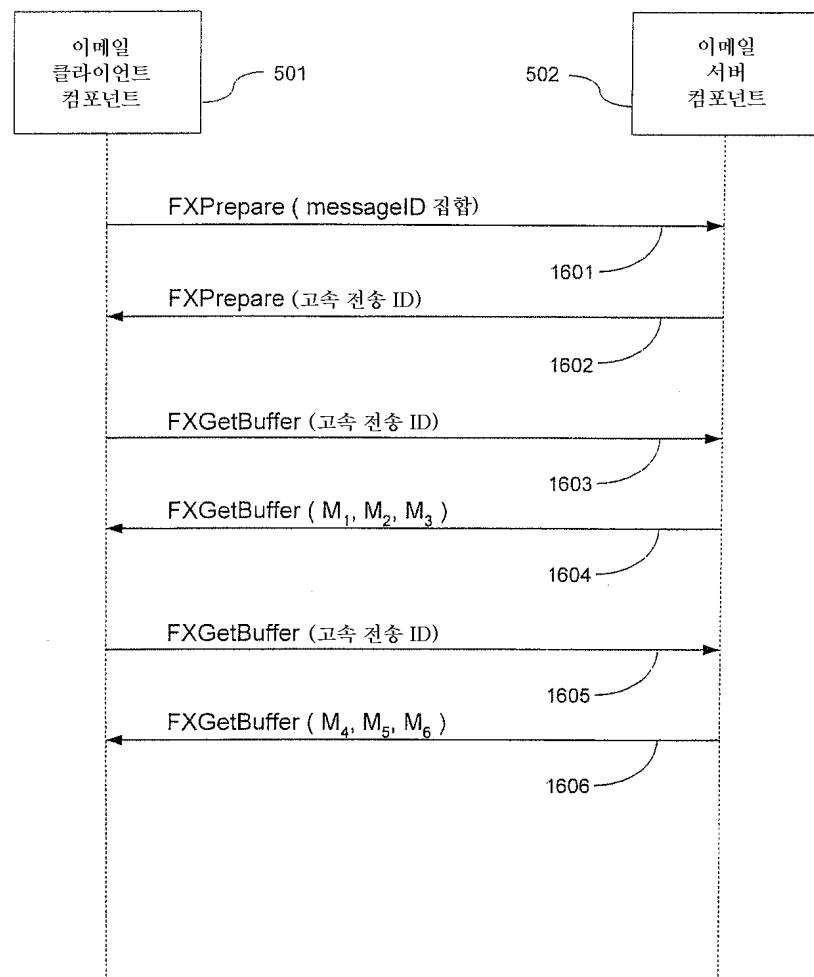
도면15a



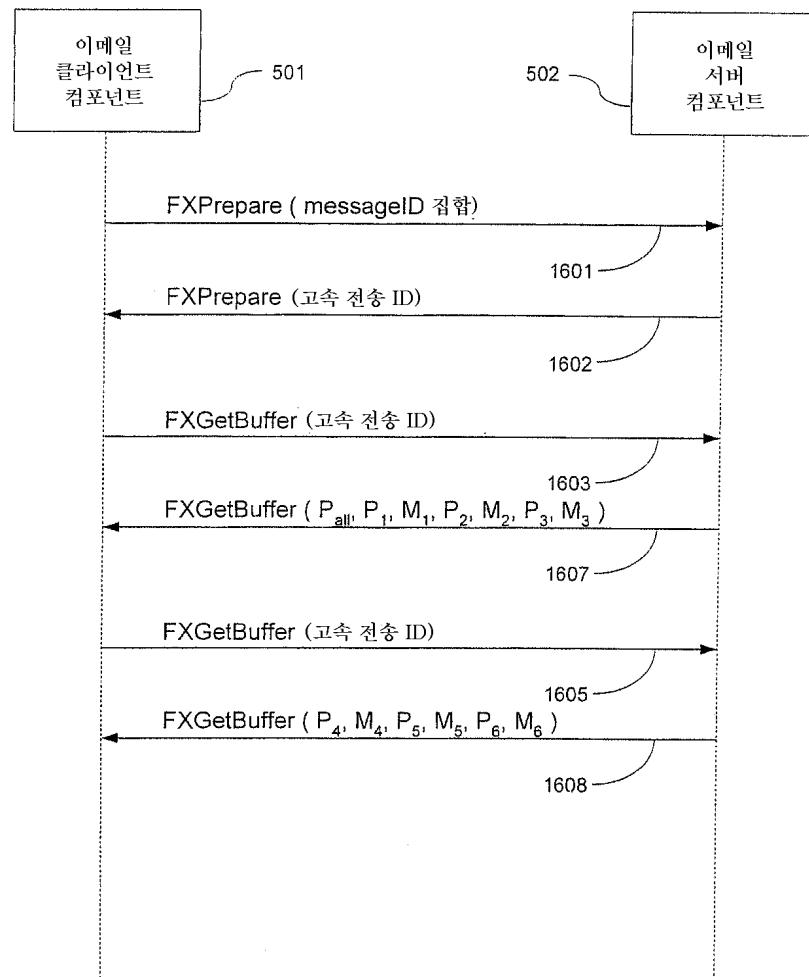
도면15b



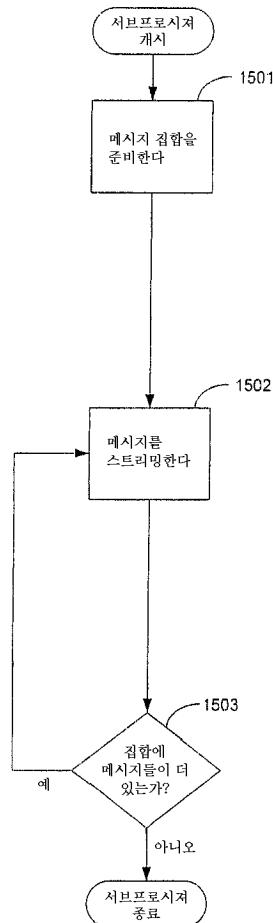
도면16a



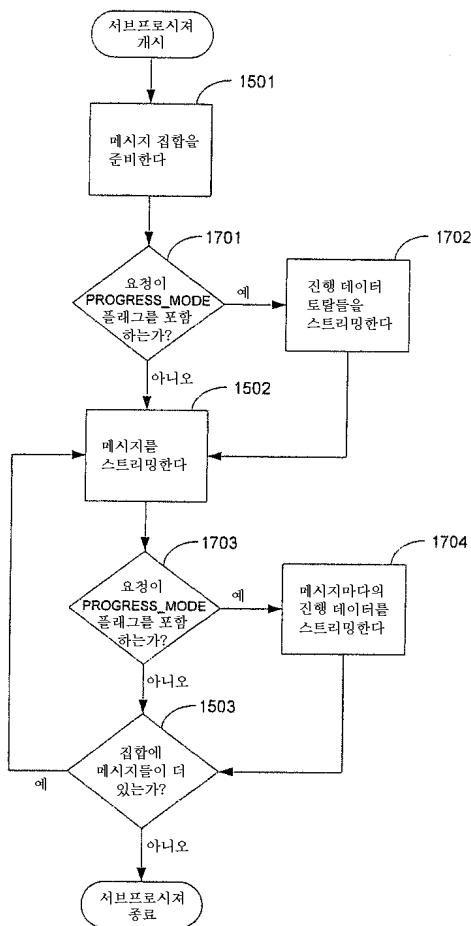
도면16b



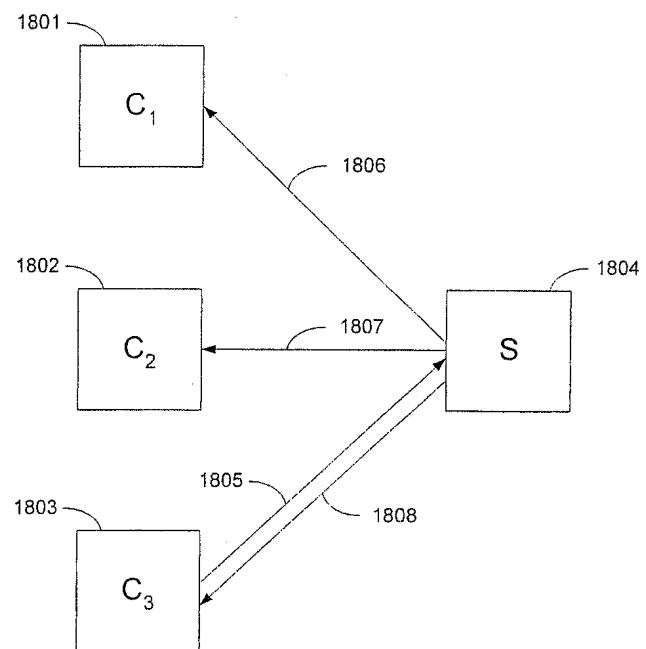
도면17a



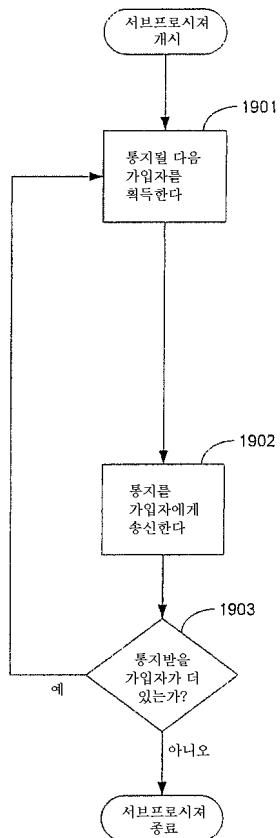
도면17b



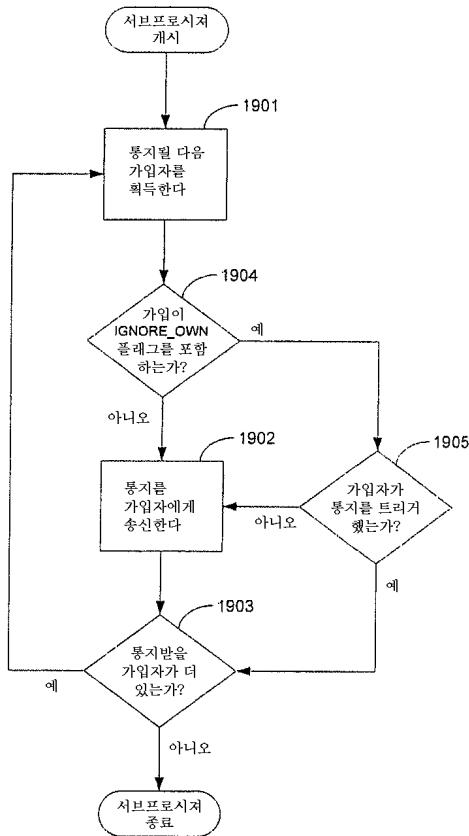
도면18



도면19a



도면19b



도면20

