



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2014년01월22일  
 (11) 등록번호 10-1354727  
 (24) 등록일자 2014년01월16일

(51) 국제특허분류(Int. Cl.)  
 H04L 29/06 (2006.01) H04L 12/46 (2006.01)  
 (21) 출원번호 10-2009-7017519  
 (22) 출원일자(국제) 2008년02월21일  
 심사청구일자 2013년01월30일  
 (85) 번역문제출일자 2009년08월21일  
 (65) 공개번호 10-2009-0123867  
 (43) 공개일자 2009년12월02일  
 (86) 국제출원번호 PCT/US2008/054485  
 (87) 국제공개번호 WO 2008/106355  
 국제공개일자 2008년09월04일  
 (30) 우선권주장  
 11/712,123 2007년02월28일 미국(US)  
 (56) 선행기술조사문헌  
 US7068669 A  
 US7406087 A  
 전체 청구항 수 : 총 20 항

(73) 특허권자  
**마이크로소프트 코포레이션**  
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
 마이크로소프트 웨이  
 (72) 발명자  
**허조그, 샤이**  
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
 소프트 웨이 마이크로소프트 코포레이션 국제 특  
 허부 내  
**해그만, 마리**  
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
 소프트 웨이 마이크로소프트 코포레이션 국제 특  
 허부 내  
 (74) 대리인  
**제일특허법인**

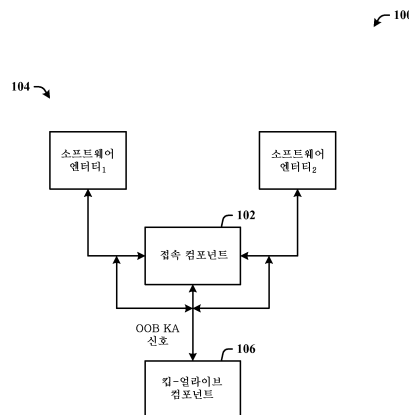
심사관 : 장석환

(54) 발명의 명칭 **NAT 시스템과 연관된 클라이언트에 대한 OOB 킵-얼라이브 메커니즘**

**(57) 요약**

기본적인 네이티브 애플리케이션(들) 상에 추가적인 조건을 가하지 않고도, 애플리케이션 접속에 외부적으로 적용될 수 있는 OOB(out-of-band) 기술을 사용함으로써 NAT(network address translation) 장치의 접속 상태를 연결 유지하기 위한 아키텍처가 제공된다. OOB 솔루션은 애플리케이션 프로토콜에 수정을 필요로 하지 않고서 임의의 접속에 적용될 수 있고, TCP 및 UDP에서 사용된다. 킵-얼라이브(KA) 애플리케이션은 NAT 장치에게 네이티브 접속으로부터 오고 있는 것으로 보이는 KA 패킷을 주입하는 OOB 메커니즘으로서 사용된다. 이 주입된 패킷은 그 접속에 대한 비활성화 타이머를 리셋하도록 NAT 장치를 속이지만, 스푸핑을 감지할 수 없는 네이티브 애플리케이션을 속이거나 혼란에 빠뜨리지는 않는다. 따라서, 접속은 NAT 타임아웃으로 인해 종료되지 않고, 따라서, 예를 들어, 클라이언트/서버 프로토콜은 접속 연결 유지를 위하여 가짜 활동 패킷을 생성할 필요가 없게 된다.

**대표도 - 도1**



**특허청구의 범위**

**청구항 1**

접속 관리(connection management)를 용이하게 하는 컴퓨터 구현 시스템으로서,

비활성화 타이머(inactivity timer)를 구비한 네트워크 장치를 포함하는 접속 컴포넌트- 상기 접속 컴포넌트는 제1 컴퓨터 시스템 상의 제1 소프트웨어 엔티티와 제2 컴퓨터 시스템 상의 제2 소프트웨어 엔티티 간의 통신 접속을 제공하고, 상기 접속 컴포넌트는, 상기 통신 접속을 위해 상기 네트워크 장치에 의해 개시된 상기 비활성화 타이머의 타임아웃 기간 내에 상기 네트워크 장치가 상기 제1 소프트웨어 엔티티와 상기 제2 소프트웨어 엔티티 간의 상기 통신 접속을 통해 통신되는 대역내(in-band) 패킷을 수신하지 못하는 경우, 상기 통신 접속을 차단하도록(drop) 구성됨 -; 및

상기 제1 소프트웨어 엔티티 및 상기 제2 소프트웨어 엔티티와는 별개로서, 상기 제1 소프트웨어 엔티티와 상기 제2 소프트웨어 엔티티 간의 상기 통신 접속을 유지하기 위하여 상기 제1 소프트웨어 엔티티와 상기 제2 소프트웨어 엔티티 간의 상기 통신 접속을 통한 대역내 트래픽으로서 상기 네트워크 장치에 의해 인식되고 상기 비활성화 타이머의 타임아웃 기간을 리셋하는 OOB(out-of-band) 스푸핑된(spoofed) 패킷들을 상기 접속 컴포넌트에 전송하기 위한 킵-얼라이브(keep-alive) 컴포넌트를 포함하고,

상기 통신 접속을 통해 상기 제1 소프트웨어 엔티티로부터 상기 제2 소프트웨어 엔티티로 통신되는 것으로 보이는 OOB 스푸핑된 패킷들은 상기 킵 얼라이브 컴포넌트에 의해 상기 제1 소프트웨어 엔티티와 상기 네트워크 장치 간의 상기 통신 접속에 삽입되고, 상기 비활성화 타이머의 타임아웃 기간을 리셋하도록 상기 네트워크 장치에 의해 처리되며, 상기 네트워크 장치에 의해 처리된 후 그리고 상기 제2 소프트웨어 엔티티에 도달하기 전에 상기 킵 얼라이브 컴포넌트에 의해 상기 통신 접속으로부터 제거되며,

상기 킵 얼라이브 컴포넌트는 상기 타임아웃 기간이 만료되기 전에 상기 통신 접속이 더 이상 필요하지 않다는 표시를 상기 제1 소프트웨어 엔티티 및 상기 제2 소프트웨어 엔티티 중 적어도 하나로부터 수신할 때까지 OOB 스푸핑된 패킷들을 상기 통신 접속에 삽입함으로써 상기 제1 소프트웨어 엔티티와 상기 제2 소프트웨어 엔티티 간의 상기 통신 접속을 유지하는,

시스템.

**청구항 2**

제1항에 있어서,

상기 통신 접속은 전송 계층 접속 지향형(connection-oriented) 단-대-단 프로토콜을 수용하는 시스템.

**청구항 3**

제1항에 있어서,

상기 통신 접속은 전송 계층 비접속형(connection-less) 프로토콜을 수용하는 시스템.

**청구항 4**

제1항에 있어서,

상기 접속 컴포넌트는 개인 네트워크와 공개 네트워크의 연결을 용이하게 하는 NAT(network address translation) 장치를 포함하는 시스템.

**청구항 5**

제1항에 있어서,

상기 OOB 스푸핑된 패킷들은 킵-얼라이브 패킷들인 시스템.

**청구항 6**

제1항에 있어서,

상기 킵-얼라이브 컴포넌트는 TCP/IP 시스템 테이블의 테이블 활동(activity)에 기초하여 상기 OOB 스푸핑된 패킷들을 삽입하는 애플리케이션인

시스템.

**청구항 7**

제1항에 있어서,

상기 킵-얼라이브 컴포넌트는 상기 비활성화 타이머의 타임아웃 기간을 모니터하고, 상기 타임아웃 기간에 기초하여 킵-얼라이브 리프레시 기간을 채용하고, 상기 리프레시 기간에 기초하여, 상기 OOB 스푸핑된 패킷들을 생성하고 상기 OOB 스푸핑된 패킷들을 상기 통신 접속에 삽입하는 시스템.

**청구항 8**

제1항에 있어서,

상기 제1 컴퓨터 시스템은 클라이언트와 서버 중 하나를 포함하는 시스템.

**청구항 9**

제8항에 있어서,

상기 킵-얼라이브 컴포넌트는 서버 기반 킵-얼라이브 애플리케이션 및 클라이언트 기반 킵-얼라이브 애플리케이션을 포함하고,

상기 서버 기반 킵-얼라이브 애플리케이션 및 상기 클라이언트 기반 킵-얼라이브 애플리케이션은 상기 접속을 유지하기 위하여 상기 접속 컴포넌트를 통해 OOB 스푸핑된 패킷들을 전송하는 시스템.

**청구항 10**

제1항에 있어서,

상기 접속 컴포넌트에 의해 처리된 상기 OOB 스푸핑된 패킷들은 제로-페이로드(zero-payload) 패킷들인 시스템.

**청구항 11**

접속을 관리하는 컴퓨터 구현 방법으로서,

비활성화 타이머를 구비한 NAT(Network Address Translation) 장치를 통해 제1 컴퓨팅 시스템의 제1 네이티브(native) 애플리케이션과 제2 컴퓨팅 시스템의 제2 네이티브 애플리케이션 간에 대역내 패킷들을 통신하기 위해 NAT 접속을 수립하는 단계- 상기 NAT 접속은, 상기 NAT 접속을 위해 상기 NAT 장치에 의해 개시된 상기 비활성화 타이머의 타임아웃 기간 내에 상기 NAT 장치가 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 통해 통신되는 대역내 패킷을 수신하지 못하는 경우, 상기 NAT 장치에 의해 차단됨 -; 및

상기 네이티브 애플리케이션들과는 별개로, 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 유지하기 위하여 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 통한 대역내 트래픽으로서 상기 NAT 장치에 의해 인식되고 상기 비활성화 타이머의 타임아웃 기간을 리셋하는 OOB(out-of-band) 스푸핑된(spoofed) 패킷들을 상기 NAT 장치에 자동적으로 전송하도록 구성된 킵-얼라이브(keep-alive) 애플리케이션을 런칭(launching)하는 단계를 포함하고,

상기 NAT 접속을 통해 상기 제1 네이티브 애플리케이션으로부터 상기 제2 네이티브 애플리케이션으로 통신되는 것으로 보이는 상기 OOB 스푸핑된 패킷들은 상기 킵 얼라이브 애플리케이션에 의해 상기 제1 네이티브 애플리케이션과 상기 NAT 장치 간의 상기 NAT 접속에 삽입되고, 상기 비활성화 타이머의 타임아웃 기간을 리셋하도록 상기 NAT 장치에 의해 처리되며, 상기 NAT 장치에 의해 처리된 후 그리고 상기 제2 네이티브 애플리케이션에 도달하기 전에 상기 킵 얼라이브 애플리케이션에 의해 상기 접속으로부터 제거되며,

상기 킵 얼라이브 애플리케이션은 상기 타임아웃 기간이 만료되기 전에 상기 NAT 접속이 더 이상 필요하지 않다는 표시를 상기 제1 네이티브 애플리케이션 및 상기 제2 네이티브 애플리케이션 중 적어도 하나로부터 수신할 때까지 OOB 스푸핑된 패킷들을 상기 NAT 접속에 삽입함으로써 상기 제1 네이티브 애플리케이션과 상기 제2 네이

티브 애플리케이션 간의 상기 NAT 접속을 유지하는, 방법.

**청구항 12**

제11항에 있어서,  
 미리 정해진 접속 정책에 기초하여 상기 OOB 스푸핑된 패킷들을 상기 접속에 자동적으로 삽입하는 단계를 더 포함하는 방법.

**청구항 13**

제11항에 있어서,  
 상기 NAT 접속은 TCP(transmission control protocol) 통신 전송 기술을 통해 용이해지는 방법.

**청구항 14**

제11항에 있어서,  
 상기 제1 네이티브 애플리케이션 및 상기 제2 네이티브 애플리케이션에 대한 대응하는 킵-얼라이브 애플리케이션들을 런칭하는 단계를 더 포함하고,  
 상기 대응하는 킵-얼라이브 애플리케이션들은 TCP/IP 시스템 테이블 내의 새로운 엔트리에 기초하여 OOB 스푸핑된 패킷들을 자동적으로 전송하여 상기 접속을 유지하는 방법.

**청구항 15**

제14항에 있어서,  
 상기 대응하는 킵-얼라이브 애플리케이션들은 접속 상태에 대한 대응하는 시스템 테이블들을 모니터하고, 다수의 NAT 접속들에 대한 접속 상태를 처리하도록 동작하는, 방법.

**청구항 16**

제11항에 있어서,  
 상기 네이티브 애플리케이션들 중 적어도 하나의 네이티브 애플리케이션과 연관된 프로토콜 번호, 소스 IP 어드레스, 소스 포트 번호, 대상(destination) IP 어드레스, 또는 대상 포트 번호 중 적어도 둘을 포함하는 튜플(tuple)을 발견하는 단계; 및  
 상기 튜플을 사용하는 OOB 스푸핑된 패킷들에 기초하여 상기 비활성화 타이머를 리셋하는 단계를 더 포함하는 방법.

**청구항 17**

제11항에 있어서,  
 상기 NAT 장치와 상기 제2 컴퓨팅 시스템 간의 상기 접속으로부터 상기 킵-얼라이브 애플리케이션에 의해 제거된 상기 OOB 스푸핑된 패킷들은 상기 제2 네이티브 애플리케이션과 연관된 대상 IP 어드레스를 이용하는, 방법.

**청구항 18**

제11항에 있어서,  
 상기 네이티브 애플리케이션들 중 적어도 하나의 네이티브 애플리케이션의 대역내(in-band) 활동에 기초하여 상기 킵-얼라이브 애플리케이션을 자동적으로 런칭하는 단계를 더 포함하는 방법.

**청구항 19**

제11항에 있어서,

하나 이상의 킵-얼라이브 애플리케이션을 자동적으로 런칭하고, 채용된 전송 프로토콜의 유형에 기초하여 상기 하나 이상의 킵-얼라이브 애플리케이션을 통해 상기 OOB 스푸핑된 패킷들을 삽입하는 단계를 더 포함하는 방법.

**청구항 20**

처리 장치 및 메모리를 포함하는 컴퓨터 구현 시스템으로서,

비활성화 타이머를 구비한 NAT(Network Address Translation) 장치를 통해 제1 컴퓨팅 시스템의 제1 네이티브 애플리케이션과 제2 컴퓨팅 시스템의 제2 네이티브 애플리케이션 간에 대역내 패킷들을 통신하기 위해 NAT 접속을 자동적으로 수립하기 위한 컴퓨터 구현 수단- 상기 NAT 접속을 위해 상기 NAT 장치에 의해 개시된 상기 비활성화 타이머의 타임아웃 기간 내에 상기 NAT 장치가 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 통해 통신되는 대역내 패킷을 수신하지 못하는 경우, 상기 NAT 장치에 의해 차단됨 -;

상기 제1 네이티브 애플리케이션 및 상기 제2 네이티브 애플리케이션과는 별개인 하나 이상의 킵-얼라이브 애플리케이션을 자동적으로 런칭하기 위한 컴퓨터 구현 수단;

시스템 테이블에 기초하여 비활성화에 대해 상기 NAT 접속을 모니터링하기 위한 컴퓨터 구현 수단; 및

상기 비활성화를 감지한 것에 응답하여, 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 유지하기 위해 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 통한 대역내 트래픽으로서 상기 NAT 장치에 의해 인식되고 상기 비활성화 타이머의 타임아웃 기간을 리셋하는 OOB 스푸핑된 패킷들을 상기 NAT 장치로 자동적으로 전송하기 위한 컴퓨터 구현 수단을 포함하고,

상기 NAT 접속을 통해 상기 제1 네이티브 애플리케이션으로부터 상기 제2 네이티브 애플리케이션으로 통신되는 것으로 보이는 OOB 스푸핑된 패킷들은 TCP 패킷 및 UDP 패킷 중 하나에 기초하여 킵 얼라이브 애플리케이션에 의해 상기 제1 네이티브 애플리케이션과 상기 NAT 장치 간의 상기 NAT 접속에 삽입되고, 상기 비활성화 타이머의 타임아웃 기간을 리셋하도록 상기 NAT 장치에 의해 처리되며, 상기 NAT 장치에 의해 처리된 후 그리고 상기 제2 네이티브 애플리케이션에 도달하기 전에 상기 킵 얼라이브 애플리케이션에 의해 상기 NAT 접속으로부터 제거되며,

상기 킵 얼라이브 애플리케이션은 상기 타임아웃 기간이 만료되기 전에 상기 NAT 접속이 더 이상 필요하지 않다는 표시를 상기 제1 네이티브 애플리케이션 및 상기 제2 네이티브 애플리케이션 중 적어도 하나로부터 수신할 때까지 OOB 스푸핑된 패킷들을 상기 NAT 접속에 삽입함으로써 상기 제1 네이티브 애플리케이션과 상기 제2 네이티브 애플리케이션 간의 상기 NAT 접속을 유지하는,

컴퓨터 구현 시스템.

**명세서**

**배경 기술**

[0001] 컴퓨팅 장치 및 네트워크에서의 기술적 진보는 광범위한 정보 및 서비스에 대한 액세스를 용이하게 하여, 사실상 세계 어느 곳으로부터도 액세스할 수 있게 되었다. 또한, 컴퓨터 및 휴대용 장치의 수가 계속해서 증가함에 따라, 접속성은 각각의 장치가 네트워크 상에 있을 때 고유하게 식별될 것을 요구할 수 있다. 각각의 네트워크 장치에 대한 별개의 (또는 정적) IP 어드레스를 얻기 위하여 추가 비용을 들여 가입을 하기 보다는, NAT(network address translation)라고 불리는 기술이 라우터 뒤 또는 내부 (또는 개인) 네트워크 상의 복수의 IP 노드들로 하여금 단일의 공개 IP 어드레스를 공유하게 할 수 있다. 즉, 내부 네트워크 트래픽에 사용되는 미등록 IP 어드레스들의 하나의 세트 및 외부 또는 공개 트래픽에 사용되는 IP 어드레스들의 다른 세트를 가능하게 하는 표준이 제공된다.

[0002] 통상적으로, NAT 장치는 네이티브 애플리케이션(native application)들 간의 접속 상태를 매핑하기 위하여 구성 가능한 타임아웃 기간을 갖는 접속 타임아웃 타이머를 사용한다. 특정 NAT 포트 매핑 테이블 엔트리가 타임아

웃 기간보다 오랫동안 인바운드 또는 아웃바운드에 의해 사용되지 않으면, 그 접속에 대한 NAT 타이머가 만료되고, 엔트리는 테이블로부터 제거된다(purged). 엔트리가 제거되고 나면, NAT 뒤의 공유 노드는 더 이상 이 접속을 통해 도달될 수 없고, 새로운 접속이 (예를 들어, 공유 노드에 의해) 개시되어야 한다.

[0003] NAT 타이머가 타임아웃(또는 만료)되는 것을 방지하기 위한 통상적인 메커니즘이 "KA(킵-얼라이브 : keep-alive)" 또는 "하트비트(heartbeat)" 프로세싱으로서 알려져 있다. 킵 얼라이브 하에서는, 타이머를 리셋하기 (또는 리프레시하기) 위하여 쓸모없는 트래픽이 NAT 타임아웃 기간보다 짧은 구간에서 접속을 통해 생성되고, 이에 의해, 접속을 활성화 상태로 유지한다. 주 전원으로서 배터리 전력을 사용하는 휴대용 장치(예를 들어, 스마트폰)의 경우, 통상적인 킵-얼라이브 기술은 배터리 수명에 영향을 미치고, 상당한 무선 활동을 생성하여 접속을 연결 유지(keeping alive)한다.

[0004] NAT를 통해 긴 수명의 접속을 제공하기 위한 솔루션은 네이티브 애플리케이션 프로토콜의 일부로서 킵-얼라이브 메커니즘을 구축하는 것이다(대역내(in-band) 솔루션). 그러나, 통상적인 메커니즘의 단점으로는, 기본적인 네이티브 애플리케이션 프로토콜이 KA 메커니즘을 수용하도록 수정되어야 하는 점, KA 메커니즘이 기존 애플리케이션(legacy application)으로 갱신될 수는 없고, 애플리케이션 업그레이드가 전개될 것을 요구하는 점, 및 KA 메커니즘에 대한 모든 업데이트가 코어 애플리케이션 프로토콜에 영향을 미치고, 그에 따라 테스트되어야 하는 점이 있다.

[0005] 또한, 통상적인 KA 메커니즘을 최적화하는 것은 대역내 제약 때문에 어려운데, 대역내 제약으로는, 계층화된 네이티브 애플리케이션 헤더(예를 들어, 많은 HTTP-하이퍼텍스트 마크업 언어 및 SOAP-심플 객체 액세스 프로토콜 헤더)를 수용하기 위하여 KA 패킷 크기를 불필요하게 크게 강제하는 점, 애플리케이션 로직이 KA 로직을 제약하고 추가적인 네트워크 접속을 필요로 할 수 있다는 점, 애플리케이션 레벨이 실패 모드 및 회복의 빠른 검출을 지원할 수 없다는 점, 애플리케이션 개발자가 기본적으로 시스템 레벨 솔루션인 것을 완전하게 하는 데에 전념하는 데에 필요한 자원, 시간 또는 전문적 지식을 갖지 못할 수 있다는 점, 및 모바일 로밍 및 상이한 네트워킹 환경에 대해 적응하고 응답하는 데에 따른 어려움이 있을 수 있다.

[0006] <발명의 개요>

[0007] 이하에는 여기에 개시된 창의적인 실시예의 기본적인 이해를 제공하기 위하여 간략화된 개요를 나타내었다. 본 개요는 광범위한 개요는 아니며, 핵심적인/중요한 구성요소를 식별하거나 그 범위를 정하기 위한 것이 아니다. 단지 후술되는 보다 상세한 설명에 대한 전제부로서 일부 개념을 간략화된 형태로 나타내는 것을 목적으로 한다.

[0008] 개시된 아키텍처는 기본적인 네이티브 애플리케이션 및/또는 애플리케이션 프로토콜 상에 임의의 조건 또는 수정을 가하지 않고도, 네이티브 애플리케이션 접속에 외부적으로 적용될 수 있는 대역외(out-of-band; OOB) 기술을 사용함으로써 NAT(network address translation) 장치의 접속 상태를 연결 유지하기 위한 솔루션을 제공한다. 본 기술은 접속을 유지하기 위하여 KA 패킷을 네이티브 애플리케이션 접속에 주입하고, 그 후에 KA 패킷이 네이티브 애플리케이션에 도달하기 전에 KA 패킷을 제거함(예를 들어, 드롭시킴)으로써 OOB 방식으로 네이티브 애플리케이션에서 동작하는 별도의 킵-얼라이브(KA) 애플리케이션을 제공한다. 대체적인 구현에서, KA 패킷은 제거되지는 않지만, 네이티브 애플리케이션에 의해 처리된다(예를 들어, 필터링, 드롭, ...).

[0009] 본 아키텍처는 공통의 NAT 상태 관리 메커니즘에서의 "논리 구멍(logical holes)"을 레버리지 한다. 이들 논리 구멍은 "버그"가 아니며 그것만으로는 보안 구멍으로서 고려되지 않고 단순히 인라인 네트워킹 장치에 위치한 제약의 아티팩트(artifact)이다. 하나의 논리 구멍의 속성은 접속 활동(activity)에 기초하여 NAT 상태를 유지하는 것에 관련된다. NAT 장치는 각각의 활성화된 접속에 대한 타임아웃 타이머를 개시하고, 접속이 트래픽(예를 들어, 패킷 활동)을 겪을 때마다 접속에 대한 타임아웃(또는 비활성화) 타이머를 리셋한다. 주어진 접속에서 어떠한 트래픽도 검출되지 않으면, 타임아웃 타이머가 그 접속에 대해 만료되고, 접속은 실패한다. 또한, NAT는 통상적으로 트래픽의 타당성(legitimacy)(예를 들어, 트래픽이 일어나는 종점)을 확인하지는 않는다.

[0010] 이 논리 구멍의 이점을 취하면, 하나의 예시적인 클라이언트/서버 환경에서, 클라이언트 및/또는 서버 상에서 실행 중인 제2 네트워크 KA 애플리케이션은 네이티브 접속(native connection)으로부터 오고 있는 것으로 보이는 스푸핑(spoofing)된 패킷을 주입하기 위하여 OOB 메커니즘으로서 사용될 수 있다. 이들 주입된 패킷은 그 접속에 대한 비활성화 타이머를 리셋하도록 NAT 장치를 속이지만(fool), 스푸핑을 감지할 수 없는 네이티브 애플리케이션을 속이거나 혼란스럽게(confuse) 하지 않는다. 따라서, 접속은 NAT 타임아웃으로 인해 종료되지 않고, 그러므로, 클라이언트/서버 프로토콜은 접속을 연결 유지하기 위하여 가짜 활동 패킷을 생성할 필요가

없다.

[0011] 본 아키텍처는 NAT 장치를 통해 임의의 비활성화 네트워크 접속에 지속성(persistency)을 제공하고, TCP(전송 제어 프로토콜) 등의 접속 지향형(connection-oriented) 단-대-단 전송 프로토콜 및 UDP(사용자 데이터그램 프로토콜) 등의 비접속형(connection-less) 전송 프로토콜에서 동작한다.

[0012] 상세한 관련 목적들을 달성하기 위하여, 개시된 창의적인 아키텍처의 임의의 예시적인 실시양태가 이하의 상세한 설명 및 첨부 도면과 관련하여 여기서 설명된다. 그러나, 이들 양태는 여기에 개시된 원리가 사용될 수 있는 각종 방식 중 단지 일부의 예시이며, 모든 이러한 양태 및 그 등가물을 포함하는 것으로 의도된다. 다른 이점 및 창의적인 특징은 도면과 함께 고려될 때 이하의 상세한 설명으로부터 명백해질 것이다.

**실시예**

[0023] 개시된 아키텍처는 네이티브 애플리케이션 상에 임의의 조건이나 수정을 가하지 않고도 네이티브 애플리케이션 접속에 대해 외부적으로 적용될 수 있는 OOB(out-of-band) 기술을 사용함으로써, NAT(network address translation) 장치 및/또는 소프트웨어의 접속을 연결 유지하기 위한 솔루션을 제공한다. 아키텍처는 NAT 장치의 관점에서 네이티브 접속(들)의 일부인 것으로 보이는 OOB 소스(예를 들어, 애플리케이션)로부터 스푸핑(spoofing)된 (또는 킵-얼라이브(KA)) 패킷을 주입함으로써 공통의 NAT 상태 관리 메커니즘에서의 "논리 구멍(logical hole)"을 레버리시한다. 이 주입된 패킷은 NAT 장치로 하여금 그 접속에 대한 비활성화 타이머를 리셋하게 하지만, 스푸핑을 감지할 수 없는 네이티브 애플리케이션(들)을 속이거나(fool) 또는 혼란스럽게(confuse) 하지는 않는다. 따라서, (예를 들어, TCP(전송 제어 프로토콜) 또는 UDP(사용자 데이터그램 프로토콜)에 기초한) 접속이 NAT 타임아웃으로 인해 종료되지 않을 것이고, 따라서, 예를 들어, 클라이언트/서버 프로토콜이 접속을 연결 유지하기 위하여 가짜의 대역내(in-band) KA 패킷을 생성할 필요는 없을 것이다.

[0024] 이제부터 도면을 참조하며, 본 도면에서 유사한 참조 부호는 전반적으로 유사한 구성요소를 참조하는 데 사용된다. 설명을 위하여, 이하의 설명에서는, 다양한 상세가 그 완전한 이해를 제공하기 위하여 개시된다. 그러나, 창의적인 실시예가 이들 특정 상세가 없어도 실시될 수 있음은 자명할 것이다. 다른 예에서는, 잘 알려진 구조 및 장치가 그 설명을 용이하게 하기 위하여 블록도 형태로 도시되어 있다.

[0025] 우선 도면을 참조해보면, 도 1은 일 실시예에 따른 접속 관리를 용이하게 하는 컴퓨터 구현 시스템(100)을 도시한다. 시스템(100)은 소프트웨어 엔터티들(104)(소프트웨어 엔터티<sub>1</sub> 및 소프트웨어 엔터티<sub>2</sub>라고 표시되어 있음) 간의 임의의 통신 접속을 제공하기 위한 접속 컴포넌트(102)(예를 들어, NAT 장치)를 포함한다. 또한, 시스템(100)은 접속을 유지하기 위하여 접속 컴포넌트(102)에 의해 처리되는 OOB KA 패킷(또는 신호(들))을 전송하기 위하여 접속 컴포넌트(102)에 인터페이스하는 킵-얼라이브 컴포넌트(106)를 포함한다.

[0026] 일 구현에서, KA 컴포넌트(106)는 소프트웨어 엔터티(104)가 존재하는 컴퓨팅 시스템 상에서 TCP/IP 스택 시스템 테이블 활동을 모니터링한다. 예를 들어, 새로운 테이블 엔트리는 KA 패킷이 삽입될 수 있는 새로운 접속을 나타낸다. TCP/IP 테이블은 소프트웨어 엔터티가 존재하는 시스템에서 모니터링될 수 있다. 예를 들어, 클라이언트 시스템 테이블이 모니터링될 수 있다. 마찬가지로, 서버가 포함된 경우, 서버 시스템 TCP/IP 테이블은 엔트리 활동(예를 들어, 엔트리 제거 또는 새로운 엔트리)에 대해 모니터링될 수 있다.

[0027] 대체적이고 선택적인 구현에서, 소프트웨어 엔터티들(104) 중 하나 또는 둘다의 접속 패킷 활동은 KA 컴포넌트(106)에 의해 모니터링될 수 있고, 그 활동에 기초하여, KA 패킷이 접속에 삽입된다. KA 패킷은 KA 컴포넌트(106)에 의해 접속 컴포넌트(102)의 대응하는 접속에 (예를 들어, 전송을 통해 패킷을 라우팅하는 다른 네트워크 장치를 통해) 외부적으로, 또한 접속 컴포넌트(102)의 어느 한측 또는 양측에 삽입되어, 삽입된 KA 패킷이 정상적인 대역내 트래픽으로서 접속 컴포넌트(102)에 의해 인지되게 한다. 그리고, KA 패킷은 접속 컴포넌트(102)에 의한 패킷(대역내 및/또는 연결 유지)의 각각의 검출된 발생(occurrence)에 대해 그 접속에 대한 접속 타이머를 리셋한다.

[0028] 시스템 테이블(들)은 KA 컴포넌트(106)에 의해 직접적으로 및/또는 접속 컴포넌트(102)를 통해 간접적으로 모니터링될 수 있다. 예를 들어, 테이블 활동이 접속이 유지되어야 하는 것으로 나타내면, KA 컴포넌트(106)는 KA 패킷을 접속에 삽입하여, 접속을 유지하기 위하여 접속 타이머를 리셋하도록 KA 패킷이 접속 컴포넌트(102)에 의해 처리되게 한다.

[0029] 대체적이고 선택적인 예에서, 소프트웨어 엔터티들(104) 중 하나 이상에 의해 모니터링되는 접속 패킷 활동(또는 그 부족)이 KA 컴포넌트(106)로 통신되어, KA 컴포넌트(106)로 하여금 KA 패킷을 접속 컴포넌트(102)로 전송하

게 하고 나서, KA 패킷을 자가 처리(self-processing)를 위해 접속에 삽입하고, 타임아웃 타이머가 리셋되게 할 수 있다. 또 다른 예에서, KA 컴포넌트(106)에 의해 직접 모니터링되는 것과 같은 접속 패킷 비활성화, 및 접속을 연결 유지하기 위한 바램에 기초하여, KA 컴포넌트(106)는 소프트웨어 엔티티들(104) 중 어느 하나 또는 둘다에 신호를 보내서, 접속을 통해 대역내 KA 패킷을 생성하여 그 접속에 대한 타임아웃 타이머를 리셋할 수 있다. 소프트웨어 엔티티들(104)이 KA 애플리케이션인 경우, KA 컴포넌트(106)는 소프트웨어 엔티티들(104)의 어느 하나 또는 둘다에 신호를 보내서 OOB 패킷을 접속에 삽입하여 접속을 유지할 수 있다. KA 패킷이 테이블(들)의 비활성화 또는 대역내 패킷 트래픽에 관계없이 주기적으로 전송될 수 있음을 이해하여야 한다.

[0030] 시스템(100)의 일 구현은 NAT 장치를 접속 컴포넌트(102)의 일부로서 포함하여, 소프트웨어 엔티티들(104)이 NAT 장치를 통해 서로 통신하게 한다. (네이티브 애플리케이션으로서의) 소프트웨어 엔티티들(104)은 접속을 통해 NAT를 거쳐 통신함으로써, 대역내(단지 네이티브 애플리케이션들 간) 패킷의 규칙적인 통신에 의해 활성화 상태의 접속을 생성한다. 타임아웃 타이머를 포함하는 NAT 장치는 네이티브 애플리케이션(104)으로부터의 대역내 패킷의 수신에 기초하여 타이머를 계속적으로 리셋한다.

[0031] 그러나, 통상적으로, 대역내 패킷이 타임아웃 기간 내에 NAT 장치에 의해 수신되지 않으면, NAT는 접속을 드롭시키고, NAT 장치를 통한 접속을 재구축하기 위한 애플리케이션이 필요하다. 아키텍처는 네이티브 애플리케이션들(104) 중 하나 이상에서 런칭하는 적어도 하나의 KA 애플리케이션을 (예를 들어, KA 컴포넌트(106)의 일부로서) 제공하여, KA 애플리케이션이 KA 패킷을 생성하고 KA 패킷을 접속에 삽입하여, NAT 장치가 타임아웃 타이머를 자동적으로 리셋하게 하고, 그에 의해 접속을 유지하게 함으로써 이러한 문제점을 해결한다. 테이블 엔트리(들)가 시스템 테이블(들)로부터 제거되면, KA 패킷은 더 이상 접속에 삽입되지 않는다.

[0032] 일 구현에서, KA 컴포넌트(106)는 운영 체제 TCP/IP 테이블에서 새로운 엔트리를 감지하는 것에 응답하여 KA 애플리케이션(들)을 런칭한다. 따라서, NAT 장치를 통해 동작하는 복수의 상이한 접속들이 관리될 수 있다. 다른 구현에서는, 하나의 KA 애플리케이션이 각각의 네이티브 애플리케이션(또는 소프트웨어 엔트리(104))에 대해 런칭된다. 여기서, 네이티브 애플리케이션은 대역내 패킷 통신을 처리하고, KA 애플리케이션은 접속이 연결해제 상태로 남겨져야 하는 것으로 판정될 때까지 접속을 유지하기 위하여 KA 패킷을 적절한 NAT 접속에 삽입함으로써 OOB 패킷 활동을 계속한다.

[0033] 도 2는 접속의 각각의 네이티브 애플리케이션에 대하여 KA 애플리케이션을 사용하는 시스템(200)을 도시한다. 여기서, KA 컴포넌트(106)는 2개의 KA 애플리케이션(202), 즉 제1 KA 애플리케이션(204)(KA APP<sub>1</sub>이라고 표시됨) 및 제2 KA 애플리케이션(206)(KA APP<sub>2</sub>라고 표시됨)을 포함한다. 제1 KA 애플리케이션(204) 또는/및 제2 KA 애플리케이션(206) 중 하나 또는 둘다는 시스템 테이블 엔트리 활동을 모니터링한다. 새로운 테이블 엔트리가 (예를 들어, TCP/IP 시스템 테이블에서) 검출되면, KA 애플리케이션(204 또는/및 206) 중 하나 또는 둘다는 KA 패킷을 접속에 삽입하는 것을 시작한다.

[0034] 대체적이고 선택적인 실시예에서, 접속 유지가 테이블 활동보다는 패킷 트래픽에 기초하는 경우, 제1 KA 애플리케이션(204)은 접속 컴포넌트(102)(예를 들어, NAT 장치)와 제1 네이티브 애플리케이션(208)(네이티브 APP<sub>1</sub>이라고 표시됨) 간의 네이티브 패킷 트래픽을 모니터링하고/하거나, 제2 KA 애플리케이션(206)은 접속 컴포넌트(102)와 제2 네이티브 애플리케이션(210)(네이티브 APP<sub>2</sub>라고 표시됨) 간의 네이티브 패킷 트래픽을 모니터링한다. 여기서 설명됨에 있어서, 네이티브 애플리케이션(208 및 210)은 대역내 통신하는 것으로 참조되고, KA 애플리케이션(204 및 206)은 OOB 통신하는 것으로 참조된다. 제1 네이티브 애플리케이션 시스템과 연관된 시스템 테이블(들)을 모니터링함으로써, 제1 네이티브 애플리케이션의 네트워크 5-튜플(예를 들어, 프로토콜 번호, 소스 IP 어드레스, 소스 포트, 대상(destination) IP 어드레스 및 대상 포트)이 발견될 수 있다. 이 정보에 기초하여, 제1 KA 애플리케이션(204)은 접속 컴포넌트(102)의 관찰된 타임아웃에 기초하여 원하는 KA 리프레시 기간을 관찰하고 사용할 수 있다. KA 리프레시 기간은 접속 컴포넌트(102)의 타임아웃 기간보다는 짧으므로, 그 접속에 대한 타임아웃 기간이 만료되기 전에 KA 패킷이 송신되게 된다. 예를 들어, 타임아웃 기간이 15분이면, 리프레시는 10분(또는 타임아웃 기간보다 짧은 임의의 다른 적절한 값)으로 선택될 수 있다.

[0035] 하나의 대체적이고 선택적인 동작에서는, 리프레시 기간에 기초하여, 접속 컴포넌트(102)의 각각의 타임아웃 기간이 만료되기 전에, 제1 KA 애플리케이션(204)이 KA 패킷을 접속에 주입할 것이다. 물론, 이는 제1 네이티브 애플리케이션(208)의 패킷 활동에 기초한다. 즉, 제1 네이티브 애플리케이션(208)이 예를 들어, 제1 네이티브 애플리케이션(208)으로부터 제1 KA 애플리케이션(204)으로의 신호에 의해 더 이상 제2 네이티브 애플리케이션(210)에서의 통신이 더 이상 원해지지 않는 것으로 나타내면, 제1 KA 애플리케이션(204)은 KA 패킷을 접속에 주

입하는 것을 그만둘 것이다. 따라서, 접속 컴포넌트(102)는 접속을 타임아웃하고, 접속은 실패할 것이다.

- [0036] 일 실시예에서, 제1 KA 애플리케이션(204)이 KA 패킷을 전송 중인 경우, 제2 KA 애플리케이션(206)은 KA 패킷을 제거할 것이다. 따라서, 제2 네이티브 애플리케이션(210)은 불필요하게 KA 패킷을 다룰 필요가 없게 된다.
- [0037] 유사하고 선택적인 동작에 의하면, 시스템 테이블보다는, 제2 네이티브 애플리케이션(210)과 접속 컴포넌트(102) 간의 패킷 트래픽을 모니터링함으로써, 제2 KA 애플리케이션(206)은 제2 네이티브 애플리케이션의 네트워크 5-튜플(예를 들어, 프로토콜 번호, 소스 IP 어드레스, 소스 포트, 대상 IP 어드레스 및 대상 포트)을 발견한다. 이 정보에 기초하여, 제2 KA 애플리케이션(206)은 접속 컴포넌트(102)의 관찰된 타임아웃에 기초하여 적절한 KA 리프레시 기간을 관찰하고 사용할 수 있다. 리프레시 정보에 기초하여, 제2 KA 애플리케이션(206)은 각각의 타임아웃 기간이 만료되기 전에 KA 패킷을 접속에 주입할 것이다. 물론, 이는 제2 네이티브 애플리케이션(210)의 활동에 기초한다. 즉, 제2 네이티브 애플리케이션(210)이 제1 네이티브 애플리케이션(208)에서의 통신이 더 이상 원해지지 않는 것으로 나타내면, 제2 KA 애플리케이션(206)은 KA 패킷을 접속에 주입하는 것을 그만둘 것이다. 따라서, 접속 컴포넌트(102)는 접속을 타임아웃하고, 접속은 실패할 것이다.
- [0038] 다른 구현에서, 제2 KA 애플리케이션(206)이 KA 패킷을 전송 중이면, 제1 KA 애플리케이션(204)은 KA 패킷을 제거할 것이다. 따라서, 제1 네이티브 애플리케이션(208)은 KA 패킷을 다룰 필요가 없게 된다.
- [0039] 다른 동작에 의하면, 제1 및 제2 네이티브 애플리케이션(208 및 210) 둘다와 접속 컴포넌트(102) 간의 패킷 트래픽을 모니터링함으로써, 대응하는 제1 및 제2 KA 애플리케이션(204 및 206)은 네이티브 애플리케이션의 네트워크 5-튜플을 발견한다. 이 정보에 기초하여, 제1 및 제2 KA 애플리케이션(204 및 206)은 접속 컴포넌트(102)의 관찰된 타임아웃에 기초하여 원하는 KA 리프레시 기간을 관찰하고 사용할 수 있다. 리프레시 기간에 기초하여, 제1 및/또는 제2 KA 애플리케이션(204 및 206)은 각각의 타임아웃 기간이 만료되기 전에 KA 패킷을 접속에 주입할 것이다. 물론, 이는 대응하는 제1 및 제2 네이티브 애플리케이션(208 및 210)의 활동에 기초한다. 또 다른 구현에서, 두 KA 애플리케이션(204 및 206)은 NAT 타임아웃이 만료되기 전의 어떤 지점에서 접속 컴포넌트(102)의 양측이 리프레시되는 한 독립적으로 동작할 수 있다.
- [0040] 즉, 이 선택적인 구현에 의하면, 제2 네이티브 애플리케이션(210)이 예를 들어, 제2 네이티브 애플리케이션(210)으로부터 제2 KA 애플리케이션(206)으로의 신호에 의해 제1 네이티브 애플리케이션(208)에서의 통신이 더 이상 원해지지 않는 것으로 나타내면, 제2 KA 애플리케이션(206)은 KA 패킷을 접속에 주입하는 것을 그만둘 것이다. 마찬가지로, 제1 네이티브 애플리케이션(208)이 제2 네이티브 애플리케이션(210)에서의 통신이 더 이상 원해지지 않는 것으로 나타내면, 제1 KA 애플리케이션(204)은 KA 패킷을 접속에 주입하는 것을 그만둘 것이다. 따라서, 접속 컴포넌트(102)는 접속을 타임아웃하고, 접속은 실패할 것이다. 제1 및 제2 KA 애플리케이션(204 및 206) 각각이 KA 패킷을 전송 중인 경우, 반대되는(opposing) 제2 및 제1 KA 애플리케이션(206 및 204) 각각은 수신된 KA 패킷을 제거할 수 있다. 따라서, 제1 및 제2 네이티브 애플리케이션(208 및 210)은 KA 패킷을 처리하도록 구성될 필요가 없다.
- [0041] 도 3은 OOB KA 패킷을 사용하여 접속을 유지하기 위한 클라이언트/서버 시스템(300)을 도시한다. 보다 구체적으로는, 클라이언트/서버 시나리오에서, 클라이언트(304)의 클라이언트 네이티브 애플리케이션(302)은 NAT 장치(306)를 통해 서버(310)의 서버 네이티브 애플리케이션(308)으로의 긴 수명의 UDP 또는 TCP 접속을 개방한다. 아키텍처는 TCP 등의 연결 지향형(connection-oriented) 단-대-단 전송 프로토콜 및 UDP 등의 비접속형(connection-less) 전송 프로토콜에서 동작한다. 또한, 클라이언트(304)는 방화벽, 필터링 또는 멀티플렉싱 컴포넌트(312)(이하, 일반적으로 방화벽(312)이라고 부름)을 포함하는데, 이를 통해 통신이 발생한다. 통신은 NAT 장치(306)의 접속을 통한 클라이언트 TCP/IP 스택 및 테이블(들)(314), 서버 TCP/IP 스택 및 테이블(들)(316) 및 서버 방화벽(318)을 통해 서버 네이티브 애플리케이션(308)으로 진행된다. 스택(314 및 316)은 구축된 각각의 새로운 접속에 대한 새로운 테이블 엔트리로 업데이트하고 드롭되는 접속에 대한 테이블 엔트리를 드롭시키는 TCP/IP 시스템 테이블과 연관되었다.
- [0042] 일 구현에 의하면, KA 애플리케이션(KA 컴포넌트(106)로서 협력적으로 개시됨)은 클라이언트(304) 및 서버(310) 상에서 런칭된다. 클라이언트 KA 애플리케이션(320)은 클라이언트(304) 상에서 런칭하고, 서버 KA 애플리케이션(322)은 서버(310) 상에서 런칭한다. KA 컴포넌트 애플리케이션(320 및 322)이 백그라운드 프로세스로서 연속적으로 실행되는 운영 체제(클라이언트 및 서버)에서 런칭될 수 있음을 이해할 것이다. 상술한 바와 같이, 네트워킹 5-튜플(예를 들어, 프로토콜 번호, 소스 IP 어드레스, 소스 포트, 대상 IP 어드레스, 및 대상 포트)이 TCP/IP 스택(314 및 316)과 연관된 TCP/IP 테이블을 통해 KA 컴포넌트(106)(클라이언트 및 서버 KA 애플리케이션(320 또는/및 322) 중 하나 또는 둘다)에 의해 발견될 수 있다.

- [0043] KA 애플리케이션(320 및 322)은 협력적으로 또는 독립적으로 (관찰된 NAT 타임아웃에 기초하여) 원하는 KA 리프레시 기간을 관찰하고 사용할 수 있다. 예를 들어, NAT 장치(306) 뒤의 (예를 들어, 개인 측의) 클라이언트(304)가 (공개 측의) 서버(310)에 대한 TCP 접속을 개방한 후, 침묵(패킷 활동이 없음)을 유지한다고 가정한다. 이 접속을 구축하면 클라이언트 TCP/IP 시스템 테이블에 새로운 엔트리가 생겨난다. KA 컴포넌트(106)가 없으면, 통상적인 구현에서와 같이, NAT 장치(306)는 접속 상태를 타임아웃하고, TCP 접속을 쓸모없게 할 것이다. 설명된 OOB 방식으로 동작하는 KA 컴포넌트(106)를 사용하면, TCP 접속이 NAT 타임아웃으로 인해 종료하지 않을 것이고, 따라서 클라이언트/서버 프로토콜이 접속을 연결 유지하기 위하여 "가짜" 대역내 활동을 발생할 필요가 없음을 보장한다. 관찰된 NAT 타임아웃이 예를 들어, 15분이면, 리프레시 기간(또는 값)은 15분보다 작을 수 있다(예를 들어, 10분). 일반적으로, KA 리프레시 기간은 관찰된 NAT 장치 타임아웃 기간보다 짧도록 사용된다.
- [0044] 동작에서, KA 애플리케이션(320 및 322)은 (전송 프로토콜에 따라) 협력적으로 또는 독립적으로 클라이언트(304), 서버(310) 또는 클라이언트(304) 및 서버(310) 둘다로부터 KA (또는 스푸핑된) 접속 패킷을 전송한다. KA 컴포넌트(106)는 NAT 장치(306)의 수신 측에서 그 스푸핑된 패킷을 제거하고, 그에 따라 KA 패킷을 처리함으로써 수신측 네이티브 애플리케이션의 혼란을 제거하도록 기능한다.
- [0045] 대체적인 구현에서는, 네이티브 애플리케이션(클라이언트 네이티브 애플리케이션(302) 또는 서버 네이티브 애플리케이션(308))이 혼란되지 않도록 (또는 에러를 일으키지 않도록) 스푸핑된 패킷을 처리할 정도로 충분히 견고한 경우, 스푸핑된 패킷의 KA 컴포넌트(106)에 의한 제거가 수행되지 않는다. 이는 스푸핑된 패킷을 인식하고 드롭시키는 것을 포함할 수 있다. 예를 들어, KA 패킷은 스푸핑된 패킷을 고유하게 정의하는 정보에 대한 패킷 데이터를 조사함으로써 수신 측상에서 필터링되고/되거나 제거될 수 있다. KA 패킷은 (헤더만을 갖는) 제로-페이로드(zero-payload) 패킷일 수 있다. 다른 방식들 또한 사용될 수 있다. NAT 장치(306)는 스푸핑된 패킷의 각각의 수신시에 (또한 네이티브 패킷의 수신시에) NAT 접속 타임아웃을 리셋한다. TCP/IP 프로토콜의 속성에 의해, 스푸핑은 미가공(raw) IP 계층에서 수행되는데, 왜냐하면 UDP/TCP 프로토콜은 복수의 애플리케이션이 전송기 및 수신기 둘다에 대하여 동일한 5-튜플에 바인딩되게 할 수 없기 때문이다.
- [0046] KA 컴포넌트(106)는 각각의 방화벽(312 및 318) 및/또는 시스템 TCP/IP 스택 및 테이블(314 및 316)과 통신함으로써 네이티브 애플리케이션 5-튜플을 발견할 수 있다. 또한, KA 패킷 제거는 필터링 기능을 사용하여 각각의 수신 방화벽(312 및 318)에 의해 달성될 수 있다. 또한, 시스템(3)은 클라이언트/서버 시나리오에 제한되지 않고, 피어-투-피어 토폴로지에도 적용가능하다.
- [0047] 도 4는 KA 애플리케이션이 복수의 접속에 대한 접속 상태를 처리하도록 동작하는 대체적인 시스템(400)을 도시한다. 시스템(400)은 NAT 장치(306)를 통해 제2 시스템(404)으로 통신하고 싶어하는 제1 시스템(402)(예를 들어, 휴대용 컴퓨터)을 포함한다. 제1 시스템(402)은 2개의 네이티브 애플리케이션, 즉 제1 네이티브 애플리케이션(406) 및 제2 네이티브 애플리케이션(408)을 포함한다. 또한, 제1 시스템(402)은 제1 시스템(402)의 TCP/IP 시스템 테이블 엔트리를 통해 제1 시스템(402)의 네이티브 애플리케이션 상호시스템(intersystem)(402 및 404) 활동을 모니터링하기 위하여 제1 및 제2 네이티브 애플리케이션(406 및 408)에 인터페이스하는 제1 KA 애플리케이션(410)을 포함한다. 또한, 선택적으로, 제1 KA 애플리케이션(410)은 NAT 장치(306)를 통해 접속의 접속 상태를 모니터링할 수 있다.
- [0048] 마찬가지로, 제2 시스템(404)(예를 들어, 웹 서버)은 NAT 장치(306)를 통해 제1 시스템(402)에 통신한다. 이 구체적인 예에서, 제2 시스템(404)은 2개의 네이티브 애플리케이션, 즉 제3 네이티브 애플리케이션(412) 및 제4 네이티브 애플리케이션(414)을 포함한다. 또한, 제2 시스템(404)은 제2 시스템(404)의 TCP/IP 시스템 테이블 엔트리를 통해 제2 시스템(404)의 네이티브 애플리케이션 활동을 모니터링하기 위하여 제3 및 제4 네이티브 애플리케이션(412 및 414)에 인터페이스하는 제2 KA 애플리케이션(416)을 포함한다. 선택적으로, 제2 KA 애플리케이션(416)은 NAT 장치(306)를 통해 NAT 접속의 접속 상태를 모니터링할 수 있다.
- [0049] 이 예에서, 제1 및 제3 네이티브 애플리케이션(406 및 412)은 NAT 장치(306)를 통해 제1 접속(CON<sub>1</sub>이라고 표시함)을 개방하고, 제2 및 제4 네이티브 애플리케이션(408 및 414)은 NAT 장치(306)를 통해 제2 접속(CON<sub>2</sub>라고 표시함)을 개방한다. 제1 시스템(402)의 새로운 TCP/IP 시스템 테이블 엔트리 및/또는 제2 시스템(404)의 새로운 TCP/IP 시스템 테이블 엔트리에 기초하여, 제1 및 제2 KA 애플리케이션(410 및 416)은 원하는 제1 및/또는 제2 접속을 위한 접속 상태를 유지하기 위하여 KA 패킷을 대응하는 제1 및 제2 접속에 제공한다.
- [0050] 초기 상태에서, 어느 시스템(402 또는 404)의 네이티브 애플리케이션도 활성화 상태이지 않은 경우, KA 애플리

케이션(410 및 416)은 런칭되지 않는다. 제1 네이티브 애플리케이션(406)이 NAT 장치(306)를 통해 제1 접속을 개방하면, 제1 KA 애플리케이션(410)은 제1 접속 KA 패킷에 대한 리프레시 기간을 런칭하고 사용한다. 제1 접속은 예를 들어, 제3 네이티브 애플리케이션(412)에 대해서는 활성화 상태이지만, 접속이 비활성화 상태이면서 제1 접속은 비활성화 상태가 아니기를 원하면, 제1 KA 애플리케이션(410)은 제1 접속을 유지하기 위하여 제1 접속 KA 패킷을 제1 접속으로 자동적으로 삽입할 것이다. 제3 네이티브 애플리케이션(412)에서의 활동의 검출시, 제2 KA 애플리케이션(416), 이제는 수신자 KA 애플리케이션은 패킷 스트림으로부터 수신된 KA 패킷을 필터링할 것이다. 따라서, 제3 네이티브 애플리케이션(412)은 KA 패킷 처리 또는 필터링에 의해 방해받지 않는 패킷 스트림을 수신할 수 있다.

[0051] 이때, 제2 네이티브 애플리케이션(408)이 활성화되고, 제4 네이티브 애플리케이션(414)에 대한 제2 접속을 개방하면, 제1 KA 애플리케이션(410)은 이미 NAT 장치(306)의 리프레시 기간을 알기 때문에, 비활성화 제어를 원한다면, 제2 접속(CON<sub>2</sub>)에 대한 NAT 장치(306)에 KA 접속 유지를 적용한다. 따라서, 제1 KA 애플리케이션(410)은 단일의 NAT 장치(306)를 통해 복수의 접속을 관리할 수 있다. 대체적인 동작에서, 제1 KA 애플리케이션(410)은 제1 접속을 관리하고, 제2 시스템(404)의 제2 KA 애플리케이션(416)은 제2 접속을 관리한다. 복수의 접속에 대한 복수의 포트를 갖는 NAT 장치의 통상적인 구현에서, 복수의 접속 KA 관리가 수행될 수 있음을 알 수 있다.

[0052] 도 5는 OOB KA 관리의 방법을 도시한다. 설명의 간략함을 위하여, 예를 들어, 플로우차트 또는 흐름도의 형태로 여기서 도시된 하나 이상의 방법론(methodology)은 일련의 단계들로서 도시되고 설명되지만, 일부 단계들이 그에 따라 여기서 도시되고 설명되는 다른 단계들과 상이한 순서로 및/또는 동시에 일어날 수 있는 것과 같이 본 방법론은 단계들의 순서에 제한되지 않음을 이해할 것이다. 예를 들어, 당업자는 본 방법론이 상태도 등과 같은 일련의 상호연관된 상태 또는 이벤트로서 대체적으로 나타낼 수 있음을 이해할 것이다. 또한, 방법론의 모든 예시되는 단계들이 창의적인 구현을 위해 필요한 것은 아니다.

[0053] 단계(500)에서, NAT 접속은 상이한 시스템들의 네이티브 애플리케이션들 간에 개방된다. 단계(502)에서, NAT 접속에 대한 리프레시 기간이 판정된다. 즉, KA 애플리케이션은 KA 패킷이 전송되는 것에 기초하여 리프레시 값의 테이블을 포함할 수 있다. 리프레시 값은 KA 애플리케이션에 (예를 들어, 30초마다) 하드코딩(hard-coding)될 수도 있고, 또는 KA 애플리케이션이 미리 계산된 값을 사용할 수도 있다. 선택적으로, 리프레시 기간은 NAT 타임아웃 기간에 기초하여 연관된 KA 애플리케이션에 의해 자동적으로 계산될 수 있다. 단계(504)에서, 접속은 시스템 테이블 활동에 기초하여 모니터링된다. 단계(506)에서, KA 패킷은 선택된 리프레시 값에 기초하여 KA 애플리케이션을 사용하여 전송측 상의 접속에 자동적으로 삽입된다. 단계(508)에서, KA 패킷은 원하는 대로 수신 측에서 제거된다. 즉, KA 패킷이 수신하는 KA 애플리케이션에 의해 수신 측에서 제거될 필요는 없다.

[0054] 도 6은 접속 정책에 기초한 접속 관리의 방법을 도시한다. 단계(600)에서, NAT 접속은 네이티브 애플리케이션들 간에 개방된다. 단계(602)에서, 리프레시 값은 NAT 타임아웃 기간에 기초하여 연관된 KA 애플리케이션에 의해 선택된다. 단계(604)에서, 접속과 연관된 정책이 획득되고 처리된다. 단계(606)에서, KA 패킷은 선택된 리프레시 값에 기초하여 KA 애플리케이션을 사용하여 전송측 상의 접속에 자동적으로 삽입될 것이다. 단계(608)에서, 접속은 정책에 따라 KA 패킷을 이용하여 동작된다. 즉, 정책은 대역내 또는 OOB 패킷 트래픽이 종료되었는지 여부에 따라, 미리 정해진 기간 동안 접속이 개방 상태로 유지되고, 또한 시간이 만료되면 접속이 차단됨을 나타낼 수 있다.

[0055] 도 7은 유형 전송 프로토콜에 기초하여 NAT 접속 비활성화를 관리하는 방법을 도시한다. 단계(700)에서, NAT 접속은 네이티브 애플리케이션들 간에 개방된다. 단계(702)에서, 네이티브 애플리케이션의 활동에 기초하여, 대응하는 KA 애플리케이션들 중 하나 이상이 런칭된다. 단계(704)에서, KA 애플리케이션들 중 하나 이상은 NAT 타임아웃 기간에 기초하여 리프레시 기간을 선택한다. 단계(706)에서, KA 애플리케이션은 스택 테이블(들) 활동에 기초하여 대응하는 네이티브 애플리케이션 접속을 모니터링한다. 단계(708)에서, TCP 패킷의 이전 통신에 기초하여, KA 패킷은 KA 애플리케이션들 각각으로부터 접속에 자동적으로 삽입된다. 다르게는, 단계(710)에서, UDP 패킷의 이전 통신에 기초하여, KA 패킷은 KA 애플리케이션들 중 하나 또는 둘다로부터 접속에 자동적으로 삽입된다. TCP가 접속 지향형 단-대-단 전송 프로토콜이기 때문에, KA 애플리케이션 둘다는 KA 패킷을 삽입하고 KA 패킷을 제거하도록 동작하는 것이 바람직하다. UDP가 비접속형 전송 프로토콜이기 때문에, 대부분의 경우, KA 애플리케이션들 중 하나만이 KA 프로토콜을 NAT 접속에 삽입하도록 동작할 수 있으면 된다.

[0056] 도 8은 접속 관리를 위하여 KA 패킷을 생성하고 사용하는 방법을 도시한다. 단계(800)에서, NAT 접속은 하나의 네이티브 애플리케이션에 의해 다른 네이티브 애플리케이션으로 NAT 장치를 통해 개방된다. 단계(802)에서, KA

애플리케이션을 런칭한 후, 리프레시 기간이 KA 애플리케이션들 중 하나 이상에 의해 사용된다. 단계(804)에서, 네이티브 애플리케이션(들)의 네트워킹 5-튜플을 발견하기 위한 발견 프로세스가 개시된다. 이는 TCP/IP 테이블을 통해서일 수 있다. 단계(806)에서, KA 패킷은 5-튜플 정보를 사용하여 특정 접속에 대해 구조화된다. 단계(808)에서, KA 패킷은 리프레시 기간에 기초하여 NAT 타임아웃 타이머를 리셋하기 위하여 접속에 삽입된다. 단계(810)에서, NAT 장치들 통해 처리되었던 KA 패킷은 네이티브 애플리케이션에 도달하기 전에 제거된다.

[0057] 본원에서 사용됨에 있어서, "컴포넌트" 및 "시스템"이란 용어는 컴퓨터 관련 엔터티, 어느 하드웨어, 하드웨어 및 소프트웨어의 조합, 소프트웨어, 또는 실행 중인 소프트웨어를 참조하도록 하였다. 예를 들어, 컴포넌트는 프로세서 상에서 실행 중인 프로세스, 프로세서, 하드디스크 드라이브, (광학 및/또는 자기 저장 매체의) 복수의 저장 장치, 객체, 실행 파일(executable), 실행 스레드, 프로그램 및/또는 컴퓨터일 수 있지만, 이에 제한되지 않는다. 예시로서, 서버 상에서 실행 중인 애플리케이션 및 서버 둘다 컴포넌트일 수 있다. 하나 이상의 컴포넌트는 프로세스 및/또는 실행 스레드 내에 존재할 수 있고, 컴포넌트는 하나의 컴퓨터 상에 로컬화되고/되거나 2 이상의 컴퓨터들 간에 분산될 수 있다.

[0058] 이제, 도 9를 참조하면, 개시된 KA 아키텍처를 실행하도록 동작가능한 컴퓨팅 시스템(900)의 블록도가 도시되어 있다. 그 다양한 양태에 대한 추가적인 콘텍스트를 제공하기 위하여, 도 9 및 이하의 논의는 본 아키텍처의 다양한 양태가 구현될 수 있는 적절한 컴퓨팅 시스템(900)의 간략하고 포괄적인 설명을 제공하기 위한 것이다. 상술한 설명은 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어들의 일반적인 문맥에서 행해졌지만, 당업자는 본 아키텍처가 또한 다른 프로그램 모듈과의 조합 및/또는 하드웨어 및 소프트웨어의 조합으로서 구현될 수 있음을 인식할 것이다.

[0059] 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조 등을 포함한다. 또한, 당업자는 본 발명의 방법이 단일 프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터뿐만 아니라, 퍼스널 컴퓨터, 핸드-헬드 컴퓨팅 장치, 마이크로프로세서 기반 또는 프로그램가능 가전 제품 등(이들 각각은 하나 이상의 연관된 장치에 동작적으로 연결될 수 있음)을 포함한 다른 컴퓨터 시스템 구성에서 실시될 수 있음을 이해할 것이다.

[0060] 또한, 예시된 양태는 통신 네트워크를 통해 연결되는 원격 처리 장치에 의해 어떤 태스크가 수행되는 분산 컴퓨팅 환경에서도 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 배치될 수 있다.

[0061] 컴퓨터는 통상적으로 다양한 컴퓨터 관독가능 매체를 포함한다. 컴퓨터 관독가능 매체는 컴퓨터에 의해 액세스될 수 있는 모든 이용가능한 매체일 수 있으며, 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 포함한다. 예로서, 컴퓨터 관독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만, 이에 제한되지 않는다. 컴퓨터 저장 매체는 컴퓨터 관독가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 데이터 등의 정보를 저장하기 위하여 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체 모두를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스되고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다.

[0062] 도 9를 다시 참조하면, 다양한 실시양태를 구현하기 위한 예시적인 컴퓨팅 시스템(900)은 처리 장치(904), 시스템 메모리(906) 및 시스템 버스(908)를 포함하는 컴퓨터(902)를 포함한다. 시스템 버스(908)는 처리 장치(904)에 대하여 시스템 메모리(906)(이에 제한되지 않음)를 포함한 시스템 컴포넌트에 대한 인터페이스를 제공한다. 처리 장치(904)는 다양하게 상업적으로 이용가능한 프로세서들 중 임의의 것일 수 있다. 또한, 듀얼 마이크로프로세서 및 다른 멀티프로세서 아키텍처도 처리 장치(904)로서 사용될 수 있다.

[0063] 또한, 시스템 버스(908)는 메모리 버스(메모리 컨트롤러를 포함하거나 포함하지 않는 것), 주변 장치 버스 및 각종 상업적으로 이용가능한 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스에 상호접속될 수 있는 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리(906)는 ROM(read-only memory)(910) 및 RAM(random access memory)(912)을 포함한다. 시동 중과 같은 때에, 컴퓨터(902) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템(BIOS)은 통상적으로 ROM, EPROM, EEPROM 등과 같은 비휘발성 메모리(910)에 저장되어 있다. 또한, RAM(912)은 데이터를 캐싱하기 위하여 정적 RAM 등의 고속 RAM을 포함한다.

- [0064] 컴퓨터(902)는 적절한 본체(도시 생략)에서의 외장형 사용을 위해 구성될 수도 있는 내장형 하드디스크 드라이브(HDD)(914)(예를 들어, EIDE, SATA), 자기 플로피 디스크 드라이브(FDD)(916)(예를 들어, 이동식 디스켓(918)으로부터 판독하거나 그에 기입함), 및 광 디스크 드라이브(920)(예를 들어, CD-ROM 디스크(922)를 판독하거나, DVD 등의 기타 고용량 광학 매체로부터 판독하거나 그에 기입함)를 더 포함한다. 하드디스크 드라이브(914), 자기디스크 드라이브(916) 및 광 디스크 드라이브(920)는 하드디스크 드라이브 인터페이스(924), 자기디스크 드라이브 인터페이스(926) 및 광 드라이브 인터페이스(928) 각각에 의하여 시스템 버스(908)에 접속될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(924)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 둘다를 포함한다.
- [0065] 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어 등의 비휘발성 저장을 제공한다. 컴퓨터(902)에 있어서, 드라이브 및 매체는 적절한 디지털 포맷의 임의의 데이터의 저장을 수용한다. 상술한 컴퓨터 판독가능 매체의 설명은 HDD, 이동식 자기 디스켓, 및 CD 또는 DVD와 같은 이동식 광학 매체를 참조하였지만, 당업자는 집 드라이브, 자기 카세트, 플래시 메모리 카드, 카트리리지 등과 같이 컴퓨터에 의해 판독가능한 다른 유형의 매체 또한 예시적인 운영 환경에서도 사용될 수 있고, 또한 임의의 이러한 매체가 개시된 방법을 수행하기 위한 컴퓨터 실행가능 명령어들을 포함할 수 있음을 이해하여야 한다.
- [0066] 운영 체제(930), 하나 이상의 애플리케이션 프로그램(932), 기타 프로그램 모듈(934) 및 프로그램 데이터(936)를 포함한 다수의 프로그램 모듈은 드라이브 및 RAM(912)에 저장될 수 있다. 또한, 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 일부는 RAM(912)에 캐싱될 수 있다. 아키텍처는 각종 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있음을 이해할 수 있다. 애플리케이션(932) 및/또는 모듈(934)은 이전에 설명한 네이티브 애플리케이션, KA 애플리케이션 및/또는 KA 컴포넌트를 포함할 수 있다.
- [0067] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(938) 및 마우스(940)와 같은 포인팅 장치를 통해 명령 및 정보를 컴퓨터(902)에 입력할 수 있다. 다른 입력 장치(도시 생략)로는, 마이크로폰, IR 원격 컨트롤, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린 등이 있다. 이들 및 기타 입력 장치는 때로는 시스템 버스(908)에 연결되는 입력 장치 인터페이스(942)를 통해 처리 장치(904)에 접속되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스 등과 같은 다른 인터페이스에 의해 접속될 수 있다.
- [0068] 또한, 모니터(944) 또는 기타 유형의 표시 장치는 비디오 어댑터(946) 등의 인터페이스를 통해 시스템 버스(908)에 접속된다. 모니터(944) 외에, 컴퓨터는 통상적으로 스피커, 프린터 등의 다른 주변 출력 장치(도시 생략)를 포함한다.
- [0069] 컴퓨터(902)는 원격 컴퓨터(들)(948)와 같은 하나 이상의 원격 컴퓨터로의 유선 및/또는 무선 통신을 통한 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(948)는 워크스테이션, 서버 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서 기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 통상적으로 컴퓨터(902)와 관련하여 상술된 구성요소들의 다수 또는 그 전부를 포함하지만, 간략화를 위하여, 메모리/저장 장치(950)만이 도시되어 있다. 도시된 논리적 접속으로는 LAN(local area network)(952) 및/또는 보다 큰 네트워크, 예를 들어, WAN(wide area network)(954)으로의 유선/무선 접속을 포함한다. 이러한 LAN 및 WAN 네트워킹 환경은 사무실 및 기업체에서 일반적이며, 그 전부가 글로벌 통신 네트워크, 예를 들어, 인터넷에 접속할 수 있는 인트라넷과 같은 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 한다.
- [0070] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(902)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(956)를 통해 LAN(952)에 접속된다. 어댑터(956)는 LAN(952)으로의 유선 또는 무선 통신을 용이하게 할 수 있으며, LAN은 또한 무선 어댑터(956)와 통신하기 위하여 그에 배치된 무선 액세스 지점을 포함할 수 있다.
- [0071] WAN 네트워킹 환경에서 사용될 때, 컴퓨터(902)는 모뎀(958)을 포함할 수 있거나, WAN(954) 상에서 통신 서버에 접속되거나, 인터넷을 통하는 것과 같이 WAN(954)을 통해 통신을 설정하기 위한 기타 수단을 포함한다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(958)은 직렬 포트 인터페이스(942)를 통해 시스템 버스(908)에 접속된다. 네트워크화된 환경에서, 컴퓨터(902) 또는 그의 일부와 관련하여 기술된 프로그램 모듈은 원격 메모리/저장 장치(950)에 저장될 수 있다. 도시된 네트워크 접속은 예시적인 것이며 이 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.
- [0072] 컴퓨터(902)는 임의의 무선 장치 또는 무선 통신에 동작적으로 배치되는 엔터티, 예를 들어, 프린터, 스캐너, 데스크탑 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 임의의 부분의 장비 또는 무선으

로 검출가능한 태그와 연관된 위치(예를 들어, 공중전화 박스, 신문 가판대, 화장실), 및 전화와 통신하도록 동작할 수 있다. 이는 적어도 Wi-Fi 및 Bluetooth™ 무선 기술을 포함한다. 따라서, 통신은 통상적인 네트워크에 의하는 것과 같은 미리 정해진 구조 또는 적어도 두 장치간의 간단한 애드혹(ad hoc) 통신일 수 있다.

[0073] 이제 도 10을 참조하면, OOB KA 처리를 사용할 수 있는 예시적인 컴퓨팅 환경(1000)의 개략적인 블록도가 도시되어 있다. 시스템(100)은 하나 이상의 클라이언트(들)(1002)를 포함한다. 클라이언트(들)(1002)는 하드웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 클라이언트(들)(1002)는 예를 들어, 쿠키(들) 및/또는 연관된 콘텍스트 정보를 하우징할 수 있다.

[0074] 또한, 시스템(1000)은 하나 이상의 서버(들)(1004)를 포함한다. 또한, 서버(들)(1004)는 하드웨어 및/또는 소프트웨어(예를 들어, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 서버(1004)는 예를 들어, 아키텍처를 사용하는 것에 의한 변형(transformation)을 수행하기 위하여 스레드를 하우징할 수 있다. 클라이언트(1002)와 서버(1004) 간의 한가지 가능한 통신은 2 이상의 컴퓨터 프로세스들 간에 전송되도록 구성되는 데이터 패킷의 형태가 있을 수 있다. 데이터 패킷은 예를 들어, 쿠키 및/또는 연관된 콘텍스트 정보를 포함할 수 있다. 시스템(1000)은 클라이언트(들)(1002)와 서버(들)(1004) 간의 통신을 용이하게 하는 데에 사용될 수 있는 통신 프레임워크(1006)(예를 들어, 인터넷과 같은 글로벌 통신 네트워크)를 포함한다.

[0075] 통신은 유선(광 섬유 포함) 및/또는 무선 기술을 통해 용이하게 될 수 있다. 클라이언트(들)(1002)는 클라이언트(들)(1002)에 로컬한 정보(예를 들어, 쿠키(들) 및/또는 연관된 콘텍스트 정보)를 저장하는 데에 사용될 수 있는 하나 이상의 클라이언트 데이터 저장소(들)에 동작적으로 접속된다. 마찬가지로, 서버(들)(1004)는 서버(1004)에 로컬한 정보를 저장하는 데에 사용될 수 있는 하나 이상의 서버 데이터 저장소(들)(1010)에 동작적으로 접속된다.

[0076] 클라이언트(1002) 및 서버(1004)는 둘다 NAT 라우터, 게이트웨이 등과 같은 네트워크 인터페이스 장치(도시 생략)를 모니터링하는 KA 애플리케이션을 포함할 수 있다. 앞서 설명된 바와 같이, 클라이언트(1002)는 로컬 KA 애플리케이션(들)을 사용함으로써 접속 상태 관리가 클라이언트들 중 하나 또는 모두에 내장형일 수 있는 피어 투 피어 방식으로 상호접속될 수 있다.

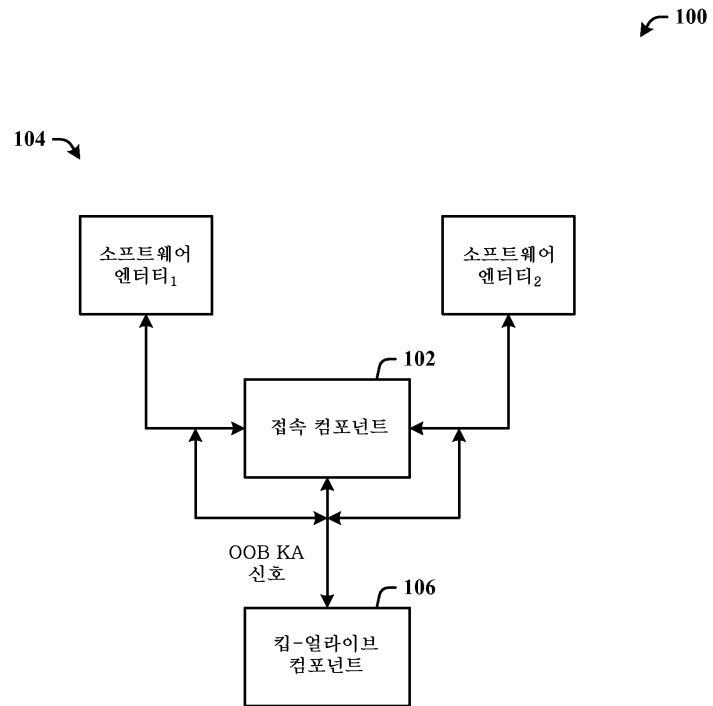
[0077] 상술한 것은 개시된 아키텍처의 예시를 포함한다. 물론, 컴포넌트들 및/또는 방법론들의 모든 생각할 수 있는 조합을 설명하는 것은 가능하지 않지만, 당업자는 여러 다른 조합 및 치환이 가능함을 인식할 수 있다. 따라서, 창의적인 아키텍처는 첨부되는 청구범위의 사상과 범주 내에 있는 모든 이러한 변경, 수정 및 변형을 포함하는 것으로 의도된다. 또한, "구비하다(comprising)"가 청구항에서 전환된 언어로 사용될 때 해석됨에 있어서, 상세한 설명 또는 청구범위 중 어느 것에서도 "포함하다(includes)"란 용어가 사용될 정도로, 이러한 용어는 "구비하다(comprising)"란 용어와 마찬가지로의 방식으로 포함적인 것으로 의도된다.

**도면의 간단한 설명**

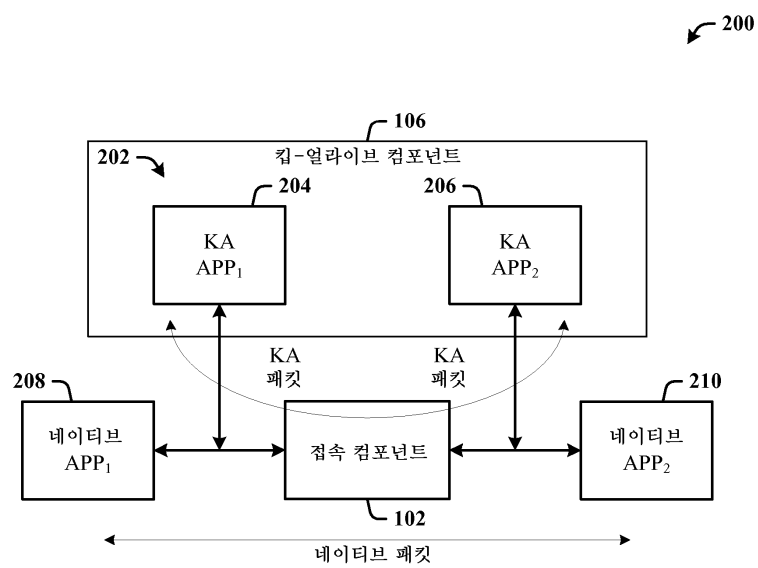
- [0013] 도 1은 일 실시예에 따른 접속 관리를 용이하게 하는 컴퓨터 구현 시스템을 도시하는 도면.
- [0014] 도 2는 접속의 각각의 네이티브 애플리케이션에 대하여 킵-얼라이브(KA) 애플리케이션을 사용하는 시스템을 도시하는 도면.
- [0015] 도 3은 OOB(out-of-band) KA 패킷을 사용하여 접속을 유지하기 위한 클라이언트/서버 시스템을 도시하는 도면.
- [0016] 도 4는 KA 애플리케이션이 복수의 접속에 대한 접속 상태를 처리하도록 동작하는 다른 시스템을 도시하는 도면.
- [0017] 도 5는 OOB KA 관리의 방법을 도시하는 도면.
- [0018] 도 6은 접속 정책에 기초한 접속 관리의 방법을 도시하는 도면.
- [0019] 도 7은 전송 프로토콜에 기초하여 NAT 접속 비활성화를 관리하는 방법을 도시하는 도면.
- [0020] 도 8은 접속 관리를 위하여 KA 패킷을 생성하고 사용하는 방법을 도시하는 도면.
- [0021] 도 9는 개시된 KA 아키텍처를 실행하도록 동작가능한 컴퓨팅 시스템을 도시하는 블록도.
- [0022] 도 10은 OOB KA 프로세싱을 사용할 수 있는 예시적인 컴퓨팅 환경을 도시하는 개략적인 블록도.

도면

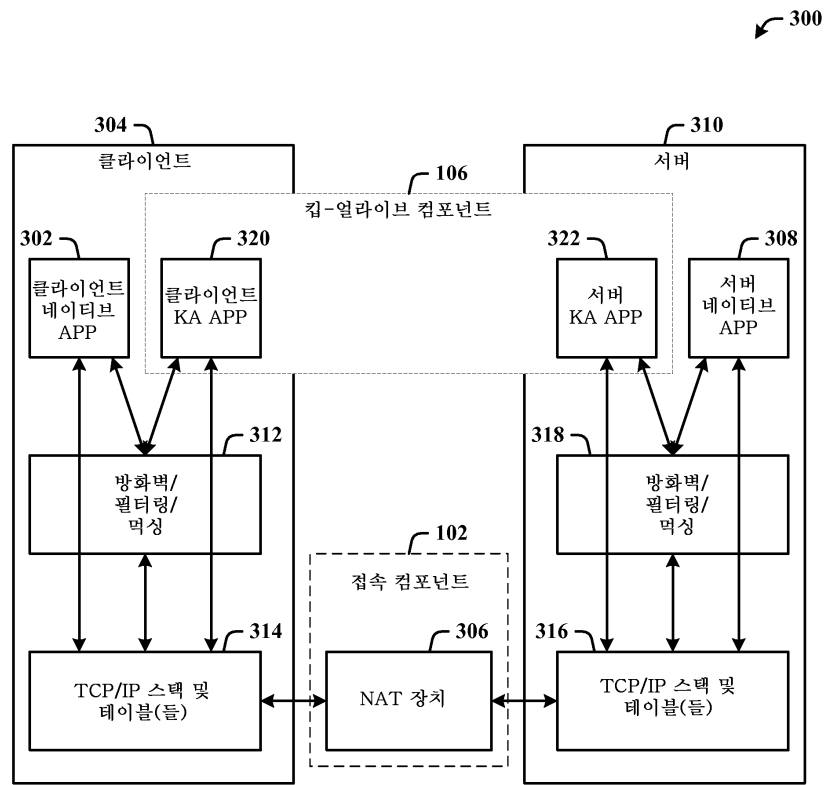
도면1



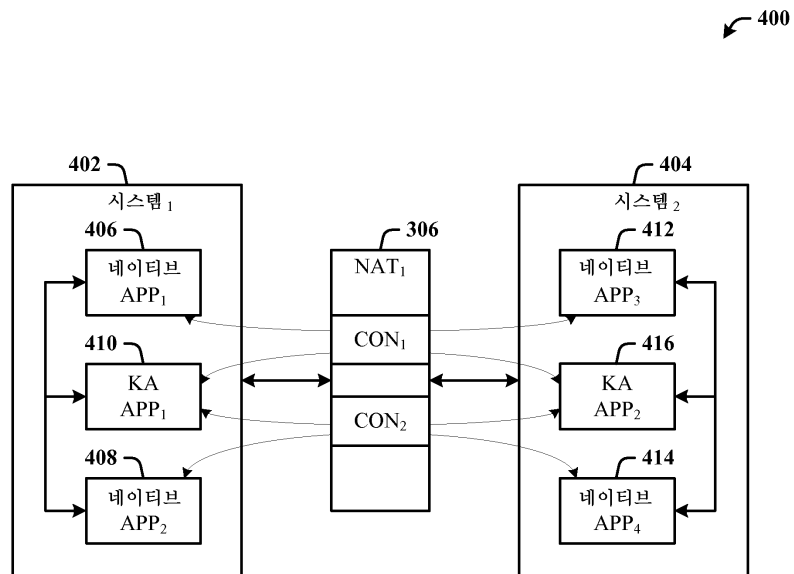
도면2



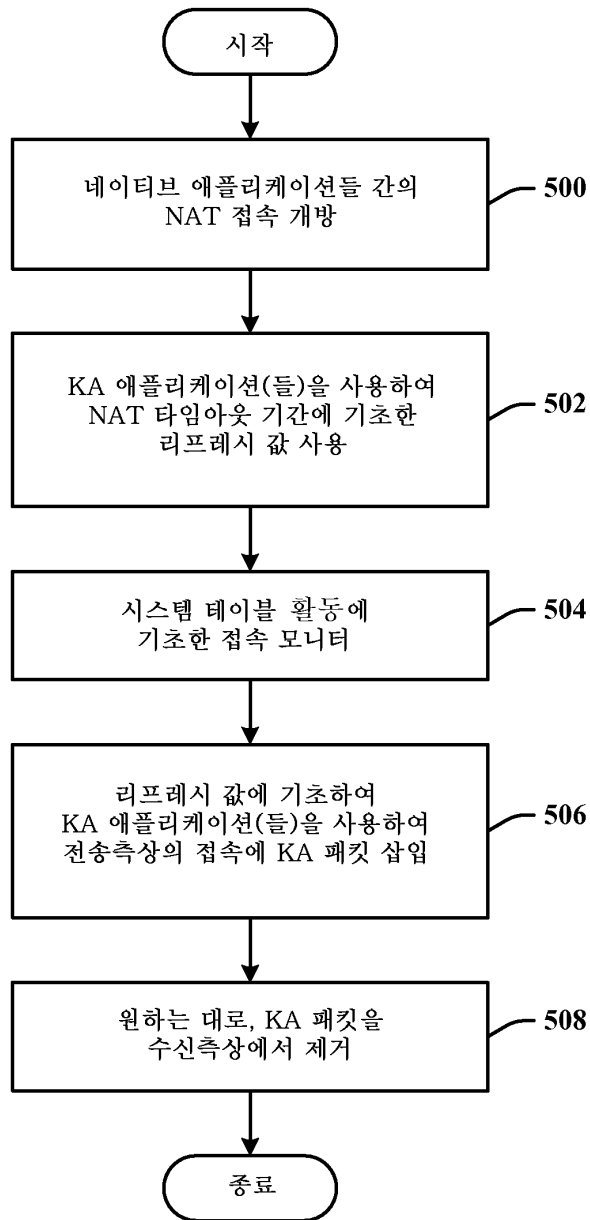
도면3



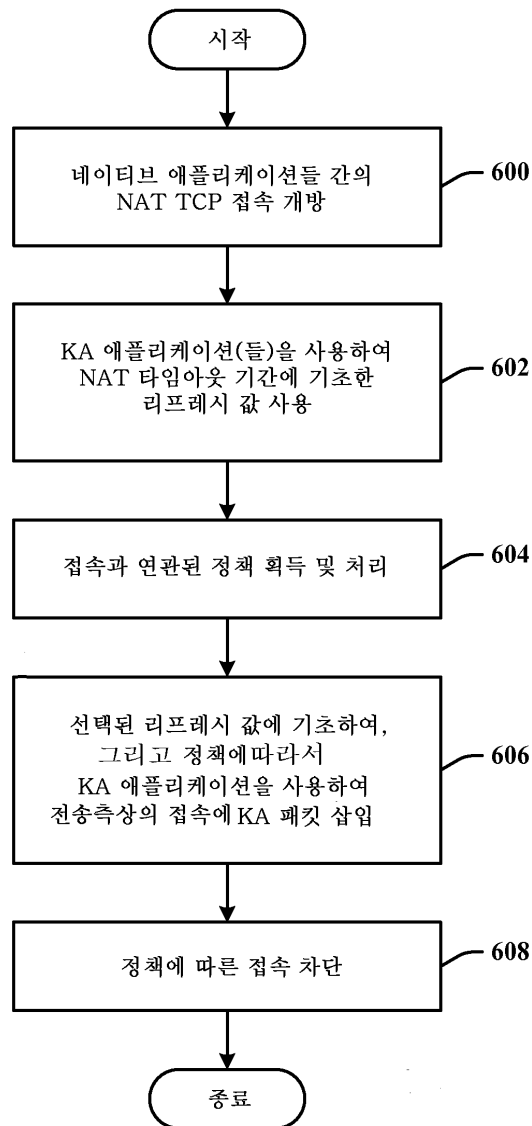
도면4



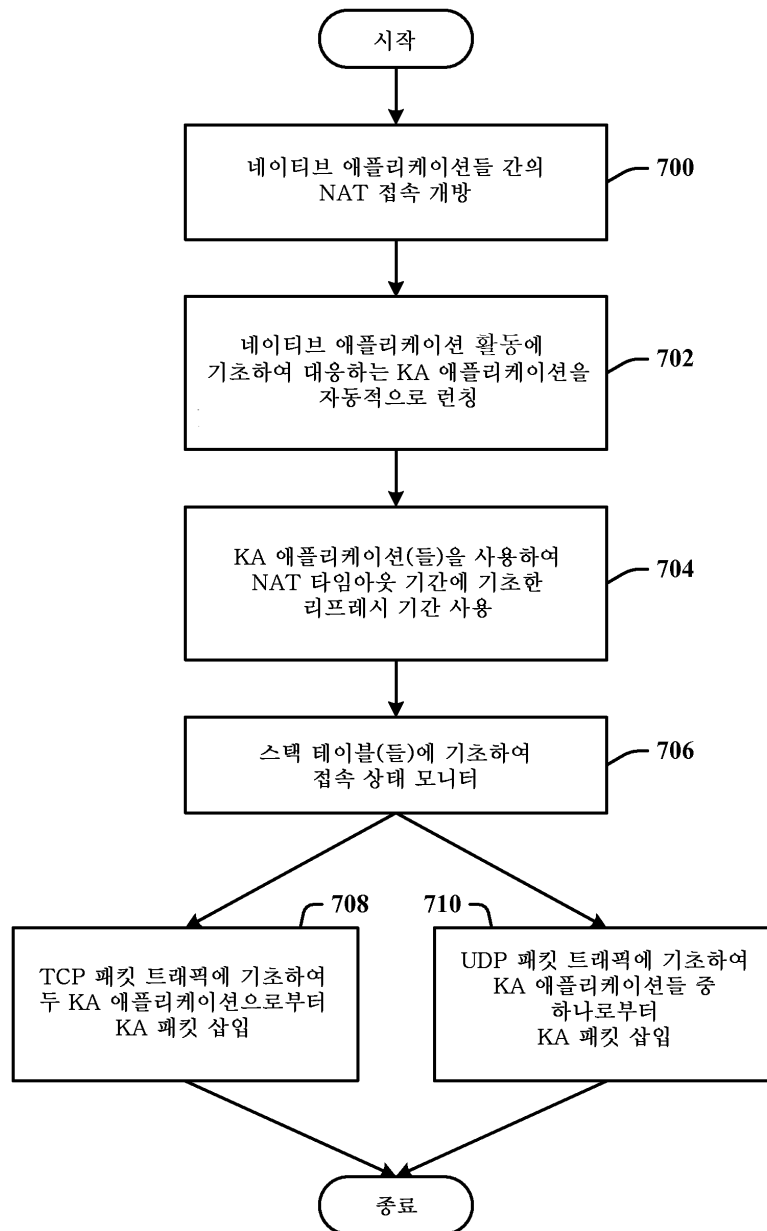
도면5



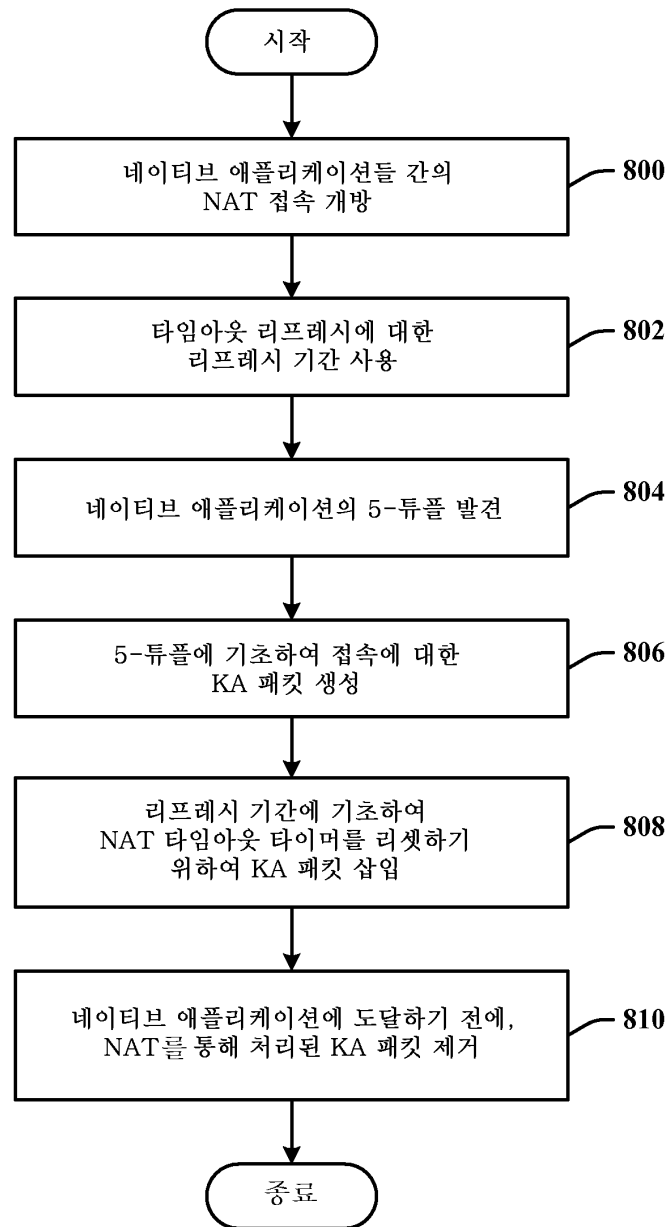
도면6



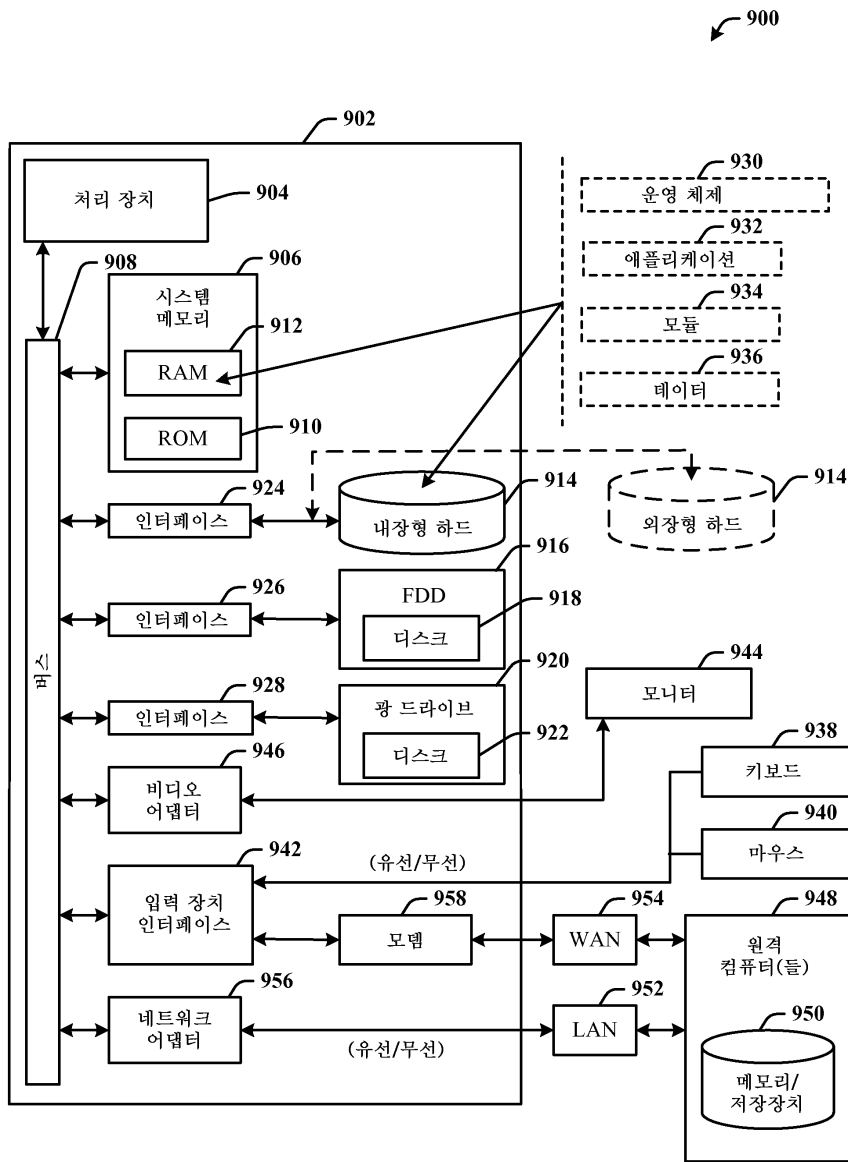
도면7



도면8



도면9



도면10

