



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0124249  
(43) 공개일자 2023년08월25일

(51) 국제특허분류(Int. Cl.)  
H04L 69/163 (2022.01) H04L 69/16 (2022.01)  
(52) CPC특허분류  
H04L 69/163 (2022.05)  
H04L 69/162 (2013.01)  
(21) 출원번호 10-2022-0021310  
(22) 출원일자 2022년02월18일  
심사청구일자 2022년02월18일

(71) 출원인  
고려대학교 산학협력단  
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)  
(72) 발명자  
이준호  
서울특별시 송파구 가락로 187, 7동 213호  
양경식  
서울특별시 동대문구 무학로45길 40, 401호(용두동)  
(뒷면에 계속)  
(74) 대리인  
김등용

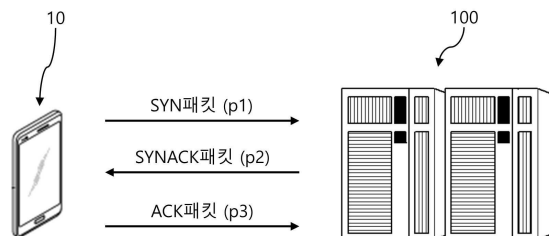
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 전송 제어 프로토콜 연결 처리 장치 및 방법

(57) 요약

전송 제어 프로토콜 연결 처리 장치 및 방법에 관한 것으로, 전송 제어 프로토콜 연결 처리 장치는 적어도 하나의 SYNACK 패킷 및 상기 적어도 하나의 SYNACK 패킷에 대응하는 적어도 하나의 리퀘스트 소켓을 보관하는 보관소 및 클라이언트 장치로부터 SYN 패킷이 수신되면, 상기 보관소를 열람하고, 상기 보관소로부터 상기 SYN 패킷에 대응하는 SYNACK 패킷 및 상기 SYN 패킷에 대응하는 리퀘스트 소켓을 검출하는 처리부를 포함하고, 상기 SYN 패킷은 상기 클라이언트 장치로 전송되고, 상기 리퀘스트 소켓은 상기 클라이언트 장치가 상기 SYN 패킷에 대한 응답으로 전송한 ACK 패킷을 수신함으로써, 상기 클라이언트 장치와 전송 제어 프로토콜 연결을 수립할 수 있다.

대표도 - 도1



(52) CPC특허분류

**H04L 69/28** (2022.05)

(72) 발명자

**유연호**

서울특별시 성북구 고려대로24가길 14-6, 102호

**유혁**

서울특별시 용산구 한강대로 205, A-803호

이 발명을 지원한 국가연구개발사업

|             |                         |
|-------------|-------------------------|
| 과제고유번호      | 1345342146              |
| 과제번호        | 2021R1A6A1A13044830     |
| 부처명         | 교육부                     |
| 과제관리(전문)기관명 | 한국연구재단                  |
| 연구사업명       | 이공학학술연구기반구축(R&D)        |
| 연구과제명       | 융합소프트웨어연구소              |
| 기 여 율       | 45/100                  |
| 과제수행기관명     | 고려대학교 산학협력단             |
| 연구기간        | 2021.06.01 ~ 2024.02.29 |

이 발명을 지원한 국가연구개발사업

|             |                                                |
|-------------|------------------------------------------------|
| 과제고유번호      | 1711081050                                     |
| 과제번호        | 2015-0-00280-005                               |
| 부처명         | 과학기술정보통신부                                      |
| 과제관리(전문)기관명 | 정보통신기획평가원                                      |
| 연구사업명       | SW컴퓨팅산업원천기술개발(R&D)                             |
| 연구과제명       | (SW 스타랩) 성능 및 보안 SLA 보장이 가능한 차세대 클라우드 인프라SW 개발 |
| 기 여 율       | 45/100                                         |
| 과제수행기관명     | 고려대학교 산학협력단                                    |
| 연구기간        | 2015.03.01 ~ 2022.12.31                        |

이 발명을 지원한 국가연구개발사업

|             |                         |
|-------------|-------------------------|
| 과제고유번호      | 1345335279              |
| 과제번호        | 4199990214667           |
| 부처명         | 교육부                     |
| 과제관리(전문)기관명 | 한국연구재단                  |
| 연구사업명       | 4단계두뇌한국21사업(R&D)        |
| 연구과제명       | 데이터주도과학 미래인재 교육연구단      |
| 기 여 율       | 10/100                  |
| 과제수행기관명     | 고려대학교 산학협력단             |
| 연구기간        | 2020.09.01 ~ 2027.08.31 |

## 명세서

### 청구범위

#### 청구항 1

적어도 하나의 SYNACK 패킷 및 상기 적어도 하나의 SYNACK 패킷에 대응하는 적어도 하나의 리퀘스트 소켓을 보관하는 보관소; 및

클라이언트 장치로부터 SYN 패킷이 수신되면, 상기 보관소를 열람하고, 상기 보관소로부터 상기 SYN 패킷에 대응하는 SYNACK 패킷 및 상기 SYN 패킷에 대응하는 리퀘스트 소켓을 검출하는 처리부;를 포함하고,

상기 SYN 패킷은 상기 클라이언트 장치로 전송되고, 상기 리퀘스트 소켓은 상기 클라이언트 장치가 상기 SYN 패킷에 대한 응답으로 전송한 ACK 패킷을 수신함으로써, 상기 클라이언트 장치와 전송 제어 프로토콜 연결이 수립되는 전송 제어 프로토콜 연결 처리 장치.

#### 청구항 2

제1항에 있어서,

상기 처리부는, 상기 클라이언트 장치와의 반복 연결이 미리 정의된 임계 값을 초과하면, 상기 클라이언트 장치와의 전송 제어 프로토콜 연결에 이용된 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 전송 제어 프로토콜 연결 처리 장치.

#### 청구항 3

제2항에 있어서,

상기 처리부는, SYN 패킷 및 리퀘스트 소켓의 재사용 가능 여부를 판단하고, 재사용 가능 여부에 따라서 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 전송 제어 프로토콜 연결 처리 장치.

#### 청구항 4

제1항에 있어서,

상기 처리부는, 상기 보관소에 저장 가능한 SYN 패킷 및 리퀘스트 소켓의 개수를 동시 수립이 가능한 전송 제어 프로토콜의 최대 개수로 유지하는 전송 제어 프로토콜 연결 처리 장치.

#### 청구항 5

제1항에 있어서,

상기 처리부는, 타이머의 구동 및 만료가 적어도 일 회 수행되는 동안, 적어도 하나의 클라이언트 장치와의 연결이 부재한지 여부를 판단하고, 만약 상기 적어도 하나의 클라이언트 장치와의 연결이 부재하다면, 상기 적어도 하나의 클라이언트 장치와의 연결에 이용된 SYN 패킷 및 리퀘스트 소켓을 상기 보관소에서 제거하는 전송 제어 프로토콜 연결 처리 장치.

#### 청구항 6

제1항에 있어서,

상기 적어도 하나의 SYNACK 패킷에 상기 적어도 하나의 리퀘스트 소켓의 대응 여부에 대한 정보를 저장하는 연결 리스트;를 더 포함하는 연결 처리 장치.

#### 청구항 7

클라이언트 장치로부터 SYN 패킷이 수신되는 단계;

SYN 패킷의 수신에 따라 보관소를 열람하고, 상기 보관소는 적어도 하나의 SYNACK 패킷 및 상기 적어도 하나의 SYNACK 패킷에 대응하는 적어도 하나의 리퀘스트 소켓을 보관하는 단계;

상기 보관소로부터 상기 SYN 패킷에 대응하는 SYNACK 패킷 및 상기 SYN 패킷에 대응하는 리퀘스트 소켓을 검출하는 단계;

상기 SYN 패킷이 상기 클라이언트 장치로 전송되는 단계;

상기 클라이언트 장치가 상기 SYN 패킷에 대한 응답으로 ACK 패킷을 전송하는 단계;

상기 리퀘스트 소켓을 통해 상기 ACK 패킷이 수신되는 단계; 및

상기 클라이언트 장치와 전송 제어 프로토콜 연결이 수립되는 단계;를 포함하는 전송 제어 프로토콜 연결 처리 방법.

## 청구항 8

제7항에 있어서,

상기 클라이언트 장치와의 반복 연결이 미리 정의된 임계 값을 초과하면, 상기 클라이언트 장치와의 전송 제어 프로토콜 연결에 이용된 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계;를 더 포함하는 전송 제어 프로토콜 연결 처리 방법.

## 청구항 9

제8항에 있어서,

상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계는,

SYN 패킷 및 리퀘스트 소켓의 재사용 가능 여부를 판단하는 단계; 및

재사용 가능 여부에 따라서 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계;를 포함하는 전송 제어 프로토콜 연결 처리 방법.

## 청구항 10

제7항에 있어서,

상기 보관소에 저장 가능한 SYN 패킷 및 리퀘스트 소켓의 개수를 동시 수립이 가능한 전송 제어 프로토콜의 최대 개수로 유지하는 단계;를 더 포함하는 전송 제어 프로토콜 연결 처리 방법.

## 청구항 11

제7항에 있어서,

타이머의 구동 및 만료가 적어도 일 회 수행되는 동안, 적어도 하나의 클라이언트 장치와의 연결이 부재한지 여부를 판단하는 단계; 및

만약 상기 적어도 하나의 클라이언트 장치와의 연결이 부재하다면, 상기 적어도 하나의 클라이언트 장치와의 연결에 이용된 SYN 패킷 및 리퀘스트 소켓을 상기 보관소에서 제거하는 단계;를 더 포함하는 전송 제어 프로토콜 연결 처리 방법.

## 발명의 설명

## 기술 분야

[0001] 전송 제어 프로토콜 연결 처리 장치 및 방법에 관한 것이다.

## 배경 기술

[0002] 전송 제어 프로토콜(TCP: Transmission Control Protocol)은 컴퓨터나 서버 장치와 같은 둘 이상의 정보 처리 장치 간의 데이터 통신을 위한 통신 프로토콜로, 통상 인터넷 프로토콜(IP: Internet Protocol)과 함께 사용된다. 전송 제어 프로토콜은, 송수신 데이터에 대한 순서를 보장하고 에러 없이 신뢰성 있는 연결을 맺을 수 있어 통상적으로 많이 사용되고 있다. 전송 제어 프로토콜은 연결 수립 과정을 필요로 하는데, 이러한 연결 수립 과정은 클라이언트로부터의 SYN 패킷의 전송, 서버 장치로부터의 SYNACK 패킷의 전송 및 클라이언트의 ACK 패킷

전송의 과정(3-way handshake)을 거쳐 수행된다. 이 경우, 서버 장치 내에서는 메모리 할당 과정이 수행된다. 구체적으로 리눅스 커널의 경우, 클라이언트가 서버로 SYN 패킷을 보내 전송 제어 프로토콜 연결을 요청하면, 서버는 해당 패킷을 수신할 리스 소켓(listen socket)을 결정하고, 메모리 할당자는 연결 지속을 위해 두 개의 인스턴스(instance)(연결 수립에 이용될 패킷인 SYNACK 패킷과, SYNACK 패킷에 대한 ACK 패킷을 수신하기 위한 소켓인 Request 소켓)를 할당한다. 두 인스턴스의 상태가 초기화되면, 서버는 클라이언트와 SYNACK 패킷과 ACK 패킷을 각각 교환하고, 수신한 ACK 패킷에 이상이 없으면 클라이언트와 연결을 생성한다. 이때, 두 인스턴스는 초기화된 상태들을 해제하고 메모리 할당자로 반환된다. 이와 같은 연결 수립 과정은 서버의 중앙 처리 장치(CPU: Central Processing Unit) 부하의 원인으로 알려져 있다. 특히 이러한 전송 제어 프로토콜의 연결 수립 과정은, 하이퍼텍스트 전송 프로토콜(HTTP: hypertext transfer protocol)을 이용하는 경우에 더 큰 문제가 될 수 있다. 하이퍼텍스트 전송 프로토콜은 단기 연결(short-lived connection) 특성을 가질 뿐만 아니라, 상술한 전송 제어 프로토콜 연결 수립 과정을 반복적으로 발생시키는 특성도 가지고 있기 때문에, 실제 데이터 패킷 교환 성능을 저하시킨다. 종래에는 이러한 문제점들을 해결하기 위해서 소켓 인터페이스의 자료 구조를 경량화하거나, 시스템 콜(System call)을 일괄 처리하고 특정한 장치를 사용하여 전송 제어 프로토콜 처리를 오프-로드하는 등의 방법으로 단기 연결 시 발생하는 부하를 해결하려고 하였다. 그러나 이들은 전송 제어 프로토콜은 유지한 상태에서 외부적인 다른 요소를 이용하는 것이므로, 다른 기능에도 영향을 미쳐 이들의 성능 저하를 야기할 뿐만 아니라, 기존 애플리케이션과의 호환성이 부족한 문제점이 있었다.

## 발명의 내용

### 해결하려는 과제

[0003] 전송 제어 프로토콜 연결의 반복 생성에 기인한 처리 장치의 불필요한 처리 병목 현상을 해결하기 위한 전송 제어 프로토콜 연결 처리 장치 및 방법을 제공하는 것을 해결하고자 하는 과제로 한다.

### 과제의 해결 수단

[0004] 상술한 과제를 해결하기 위하여 전송 제어 프로토콜 연결 처리 장치 및 방법이 제공된다.

[0005] 전송 제어 프로토콜 연결 처리 장치는, 적어도 하나의 SYNACK 패킷 및 상기 적어도 하나의 SYNACK 패킷에 대응하는 적어도 하나의 리퀘스트 소켓을 보관하는 보관소 및 클라이언트 장치로부터 SYN 패킷이 수신되면, 상기 보관소를 열람하고, 상기 보관소로부터 상기 SYN 패킷에 대응하는 SYNACK 패킷 및 상기 SYN 패킷에 대응하는 리퀘스트 소켓을 검출하는 처리부를 포함하고, 상기 SYN 패킷은 상기 클라이언트 장치로 전송되고, 상기 리퀘스트 소켓은 상기 클라이언트 장치가 상기 SYN 패킷에 대한 응답으로 전송한 ACK 패킷을 수신함으로써, 상기 클라이언트 장치와 전송 제어 프로토콜 연결이 수립될 수 있다.

[0006] 상기 처리부는, 상기 클라이언트 장치와의 반복 연결이 미리 정의된 임계 값을 초과하면, 상기 클라이언트 장치와의 전송 제어 프로토콜 연결에 이용된 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관할 수도 있다.

[0007] 상기 처리부는, SYN 패킷 및 리퀘스트 소켓의 재사용 가능 여부를 판단하고, 재사용 가능 여부에 따라서 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 것도 가능하다.

[0008] 상기 처리부는, 상기 보관소에 저장 가능한 SYN 패킷 및 리퀘스트 소켓의 개수를 동시 수립이 가능한 전송 제어 프로토콜의 최대 개수로 유지할 수도 있다.

[0009] 상기 처리부는, 타이머의 구동 및 만료가 적어도 일 회 수행되는 동안, 적어도 하나의 클라이언트 장치와의 연결이 부재한지 여부를 판단하고, 만약 상기 적어도 하나의 클라이언트 장치와의 연결이 부재하다면, 상기 적어도 하나의 클라이언트 장치와의 연결에 이용된 SYN 패킷 및 리퀘스트 소켓을 상기 보관소에서 제거할 수도 있다.

[0010] 전송 제어 프로토콜 연결 처리 장치는, 상기 적어도 하나의 SYNACK 패킷에 상기 적어도 하나의 리퀘스트 소켓의 대응 여부에 대한 정보를 저장하는 연결 리스트를 더 포함할 수도 있다.

[0011] 전송 제어 프로토콜 연결 처리 방법은, 클라이언트 장치로부터 SYN 패킷이 수신되는 단계, SYN 패킷의 수신에 따라 보관소를 열람하고, 상기 보관소는 적어도 하나의 SYNACK 패킷 및 상기 적어도 하나의 SYNACK 패킷에 대응하는 적어도 하나의 리퀘스트 소켓을 보관하는 단계, 상기 보관소로부터 상기 SYN 패킷에 대응하는 SYNACK 패킷 및 상기 SYN 패킷에 대응하는 리퀘스트 소켓을 검출하는 단계, 상기 SYN 패킷이 상기 클라이언트 장치로 전송되

는 단계, 상기 클라이언트 장치가 상기 SYN 패킷에 대한 응답으로 ACK 패킷을 전송하는 단계, 상기 리퀘스트 소켓을 통해 상기 ACK 패킷이 수신되는 단계 및 상기 클라이언트 장치와 전송 제어 프로토콜 연결이 수립되는 단계를 포함할 수도 있다.

[0012] 전송 제어 프로토콜 연결 처리 방법은, 상기 클라이언트 장치와의 반복 연결이 미리 정의된 임계 값을 초과하면, 상기 클라이언트 장치와의 전송 제어 프로토콜 연결에 이용된 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계를 더 포함할 수도 있다.

[0013] 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계는, SYN 패킷 및 리퀘스트 소켓의 재사용 가능 여부를 판단하는 단계 및 재사용 가능 여부에 따라서 상기 SYN 패킷 및 상기 리퀘스트 소켓을 상기 보관소에 보관하는 단계를 포함할 수도 있다.

[0014] 전송 제어 프로토콜 연결 처리 방법은, 상기 보관소에 저장 가능한 SYN 패킷 및 리퀘스트 소켓의 개수를 동시에 수립이 가능한 전송 제어 프로토콜의 최대 개수로 유지하는 단계를 더 포함하는 것도 가능하다.

[0015] 전송 제어 프로토콜 연결 처리 방법은, 타이머의 구동 및 만료가 적어도 일 회 수행되는 동안, 적어도 하나의 클라이언트 장치와의 연결이 부재한지 여부를 판단하는 단계 및 만약 상기 적어도 하나의 클라이언트 장치와의 연결이 부재하다면, 상기 적어도 하나의 클라이언트 장치와의 연결에 이용된 SYN 패킷 및 리퀘스트 소켓을 상기 보관소에서 제거하는 단계를 더 포함할 수도 있다.

### 발명의 효과

[0016] 상술한 전송 제어 프로토콜 연결 처리 장치 및 방법에 의하면, 전송 제어 프로토콜 연결의 반복 생성에 기인한 프로세서의 불필요한 처리 병목 현상을 해결할 수 있게 되고, 이에 따라 더 많은 전송 제어 프로토콜 연결과 패킷을 처리할 수 있게 되어, 서버 장치 등과 같은 장치의 성능을 보다 개선할 수 있게 되는 장점을 얻을 수 있다.

[0017] 상술한 전송 제어 프로토콜 연결 처리 장치 및 방법에 의하면, 전송 제어 프로토콜 연결 수립에 사용하는 자료 구조를 재사용함으로써, 메모리 할당과 초기화에 의한 오버헤드를 줄이고, 캐싱(Caching) 효과를 극대화하여 프로세서의 퍼포먼스를 향상시킬 수 있게 된다.

[0018] 상술한 전송 제어 프로토콜 연결 처리 장치 및 방법에 의하면, 하이퍼텍스트 전송 프로토콜과 같이 전송 제어 프로토콜 연결이 대량으로 반복 발생하여 전송 제어 프로토콜 트래픽이 큰 경우에도, 서버 장치는 성능 저하 없이 클라이언트가 요구하는 서비스를 안정적으로 제공할 수 있게 되는 효과도 얻을 수 있다.

### 도면의 간단한 설명

[0019] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 전송 제어 프로토콜 연결 처리 시스템의 일 실시예에 대한 개요도이다.

도 2는 전송 제어 프로토콜 연결 처리 장치의 일 실시예에 대한 블록도이다.

도 3은 코어마다 인스턴스 보관소를 할당하는 일례를 도시한 도면이다.

도 4는 전송 제어 프로토콜 연결 처리 방법의 일 실시예에 대한 흐름도이다.

도 5는 반복 사용되는 인스턴스의 보관 여부를 결정하기 위한 과정의 일 실시예를 설명하기 위한 흐름도이다.

도 6은 사용되지 않는 인스턴스를 제거하기 위한 과정의 일 실시예를 설명하기 위한 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0020] 이하 명세서 전체에서 동일 참조 부호는 특별한 사정이 없는 한 동일 구성요소를 지칭한다. 이하에서 사용되는 '부'가 부가된 용어는, 소프트웨어 및/또는 하드웨어로 구현될 수 있으며, 실시예에 따라 하나의 '부'가 하나의 물리적 또는 논리적 부품으로 구현되거나, 복수의 '부'가 하나의 물리적 또는 논리적 부품으로 구현되거나, 하나의 '부'가 복수의 물리적 또는 논리적 부품들로 구현되는 것도 가능하다. 명세서 전체에서 어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 어떤 부분과 다른 부분이 상호 간에 물리적으로 연결되었음을 의미할 수도 있고, 및/또는 전기적으로 연결되었음을 의미할 수도 있다. 또한, 어떤 부분이 다른 부분을 포함한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 부분 이외의 또 다른 부분을 제외하는 것이 아니며, 설계자의 선택

에 따라서 또 다른 부분을 더 포함할 수 있음을 의미한다. 제1 내지 제N(N은 1 이상의 자연수) 등의 표현은, 적어도 하나의 부분(들)을 다른 부분(들)으로부터 구분하기 위한 것으로, 특별한 기재가 없는 이상 이들이 순차적임을 반드시 의미하지는 않는다. 또한 단수의 표현은 문맥상 명백하게 예외가 있지 않는 한, 복수의 표현을 포함할 수 있다.

[0021] 이하 도 1 내지 도 3을 참조하여 전송 제어 프로토콜 연결 처리 시스템 및 전송 제어 프로토콜 연결 처리 장치의 일 실시예에 대해 설명하도록 한다.

[0022] 도 1은 전송 제어 프로토콜 연결 처리 시스템의 일 실시예에 대한 개요도이다.

[0023] 도 1에 도시된 바에 의하면, 전송 제어 프로토콜 연결 처리 시스템(1, 이하 처리 시스템)은, 일 실시예에 있어서, 클라이언트 장치(10)와, 클라이언트 장치(10)와 통신 가능하게 연결된 서버 장치를 포함하는 제어 프로토콜 연결 수립 제어 장치(100, 이하 TCP 연결 처리 장치라 함)를 포함할 수 있다. 클라이언트 장치(10)와 TCP 연결 처리 장치(100)는 상호 유선 통신 네트워크 및 무선 통신 네트워크 중 적어도 하나를 기반으로 통신을 수행할 수 있다. 여기서, 무선 통신 네트워크는 근거리 통신 네트워크(예를 들어, 와이파이(Wi-Fi), 와이파이 다이렉트(Wi-Fi Direct), 블루투스(Bluetooth), 저전력 블루투스(Bluetooth Low Energy), 지그비(zigbee) 통신, 초광대역(UWB: Ultra-WideBand) 통신, 알에프아이디(RFID: Radio-Frequency IDentification), 캔(CAN) 통신 및/또는 엔에프씨(NFC: Near Field Communication) 등) 또는 원거리 통신 네트워크(예를 들어, 3GPP, 3GPP2, 와이브로 또는 와이맥스 계열 등의 이동 통신 표준을 기반으로 구현된 이동 통신 네트워크 등)를 포함할 수도 있다.

[0024] 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 중 적어도 하나는, 실시예에 따라, 하나 또는 둘 이상의 정보 처리장치를 이용하여 구현된 것일 수도 있고, 또는 후술하는 동작 및 기능의 수행을 위해 별도로 특별히 고안된 장치를 포함할 수도 있다. 여기서, 정보처리장치는, 예를 들어, 서버용 하드웨어 장치, 데스크톱 컴퓨터, 랩톱 컴퓨터, 스마트 폰, 태블릿 피씨, 스마트 시계, 스마트 태그, 스마트 밴드, 두부 장착형 디스플레이(HMD: Head Mounted Display) 장치, 휴대용 게임기, 내비게이션 장치, 개인용 디지털 보조기(PDA: Personal Digital Assistant), 보이스 레코더, 동영상 촬영 장치(캠코더나 액션캠 등), 스캐너 장치, 프린터 장치, 삼차원 프린터 장치, 스마트 키, 원격 제어 장치(리모컨), 디지털 텔레비전, 셋 톱 박스, 디지털 미디어 플레이어 장치, 미디어 스트리밍 장치, 디브이디 재생 장치, 콤팩트 디스크 재생 장치, 음향 재생 장치(인공 지능 스피커 등), 가전 기기, 유인/무인 이동체(차량, 이동성 로봇 또는 무선 모형 차량 등), 유인/무인 비행체(항공기, 드론 또는 모형 비행기 등), 의료 기기, 가정용/산업용/군사용 로봇 또는 산업용/군사용 기계 등을 포함할 수 있으나 이에 한정되는 것은 아니다.

[0025] 도 1에 도시된 바에 의하면, 클라이언트 장치(10)와 TCP 연결 처리 장치(100)는 상호 간의 통신을 위해 먼저 전송 제어 프로토콜 기반 연결(이하 TCP 연결)을 수립한다. 이 경우, 클라이언트 장치(10)는 먼저 TCP 연결을 위한 SYN 패킷을 TCP 연결 처리 장치(100)로 전달하고(p1), TCP 연결 처리 장치(100)는 이에 응하여 후술하는 일련의 동작을 수행한 후, SYN 패킷을 수신하였음을 확인하기 위해, SYN 패킷에 대한 응답으로, SYN 패킷에 대응하는 SYNACK 패킷(도 2의 191)을 클라이언트 장치(10)로 전송한다(p2). 클라이언트 장치(10)는 SYNACK 패킷(191)에 대한 응답으로, SYNACK 패킷(191)을 수신하였음을 확인하기 위한 ACK 패킷을 TCP 연결 처리 장치(100)로 전송하고(p3), 이에 따라 클라이언트 장치(10)와 TCP 연결 처리 장치(100) 양자 간의 TCP 연결이 수립된다. 이후, 클라이언트 장치(10)와 TCP 연결 처리 장치(100)는 TCP 연결을 기반으로 일방으로 또는 쌍방으로 데이터, 명령/지시 또는 프로그램 등을 전송할 수 있게 된다.

[0026] 도 2는 전송 제어 프로토콜 연결 처리 장치의 일 실시예에 대한 블록도이고, 도 3은 코어마다 인스턴스 보관소를 할당하는 일례를 도시한 도면이다.

[0027] 도 2에 도시된 바와 같이, 일 실시예에 있어서, TCP 연결 처리 장치(100)는 SYNACK 패킷(191)의 전송(p2) 이전에 두 개의 인스턴스(191, 192)를 적어도 일회성으로 사용할 수 있다. 두 개의 인스턴스는 SYN 패킷에 대응하는 SYNACK 패킷(191)과, SYNACK 패킷(191)에 대응하는 ACK 패킷을 수신하기 위한 소켓(socket)인 리퀘스트 소켓(192, request socket)을 포함할 수 있다. 통상적으로 동일한 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 간의 연결에서 SYNACK 패킷(191)과 리퀘스트 소켓(192)의 상태는 큰 변경 없이 유지된다. 일 실시예에 의하면, SYNACK 패킷(191)은, 보관소(120)에 보관된 적어도 하나의 SYNACK 패킷(191-1 내지 191-k) 중에서 선택된 것이고, 리퀘스트 소켓(192)도 보관소(120)에 보관된 적어도 하나의 리퀘스트 소켓(192-1 내지 192-k) 중에서 선택된 것일 수 있다. 여기서, 보관소(120)에 보관된 SYNACK 패킷(191-1 내지 191-k) 및 보관소(120)에 보관된 적어도 하나의 리퀘스트 소켓(192-1 내지 192-k)은, 각각 상술한 클라이언트 장치(10) 또는 다른 클라이언트 장치와의 통신 과정에서 기 사용된 바 있는 SYNACK 패킷 및 리퀘스트 소켓일 수도 있다. 다시 말해서, 클라이언

트 장치(10)와의 연결 수립을 위한 SYNACK 패킷(191)과 리퀘스트 소켓(192)은 재사용되는 것일 수 있다. SYNACK 패킷(191)과 리퀘스트 소켓(192)의 재사용에 의해, 종래의 TCP 연결 과정에서 수행되었던 초기화 처리가 필요 없게 된다.

[0028] 도 2에 도시된 바를 참조하면, TCP 연결 처리 장치(100)는, 일 실시예에 있어서, 통신부(101)와, 보관소(120)와, 처리부(130)를 포함할 수 있다.

[0029] 통신부(101)는, 유선 또는 무선 통신 네트워크에 접속하여 클라이언트 장치(10)로부터 패킷(예를 들어, SYN 패킷)을 수신하고 이를 리슨 소켓(102)으로 전달하거나, 또는 클라이언트 장치(10)로 패킷(예를 들어, SYNACK 패킷(191))을 전송할 수 있다. 통신부(101)는, 외부에서 전달되는 전기적 신호나 전파 신호를 수신하고, 수신한 신호를 전기적 신호로 변환하여 출력하거나, 수신한 아날로그 신호를 디지털 신호로 변환하거나, 전기적 신호를 증폭하는 등의 동작을 수행할 수도 있다. 통신부(101)는, 예를 들어, 랜 카드나, 무선 통신 모듈 등을 이용해 구현될 수 있다.

[0030] 리슨 소켓(102, listen socket)은, 들어오는 연결을 수신하기 위한 소켓이다. 예를 들어, 리슨 소켓(102)은 클라이언트 장치(10)가 전송한 적어도 하나의 패킷(일례로 SYN 패킷 또는 ACK 패킷)이 전달되면, 이에 대응하는 정보를 처리부(130)에 전달함으로써, 처리부(130)가 클라이언트 장치(10)로부터 TCP 연결 요청이 왔음을 알 수 있게 한다.

[0031] 보관소(120)는, 적어도 하나의 SYNACK 패킷(191-1 내지 191-k)과, 하나의 리퀘스트 소켓(192-1 내지 192-k) 등과 같은 인스턴스를 일시적으로 또는 비일시적으로 보관할 수 있다. 이 경우, 보관소(120)는 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k)을 서로 대응하는 바에 따라 쌍으로 저장할 수도 있다. 일 실시예에 따르면, SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k)은 사전에 클라이언트 장치(10) 또는 다른 클라이언트 장치와의 TCP 연결 수립 과정에서 생성되었던 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k)을 포함할 수 있다. 다시 말해서, SYNACK 패킷(191-1 내지 191-k)은 과거에 클라이언트 장치(10) 또는 다른 클라이언트 장치로 전송된 패킷이고, 리퀘스트 소켓(192-1 내지 192-k)은 과거에 클라이언트 장치(10) 또는 다른 클라이언트 장치가 전송한 ACK 패킷을 수신한 소켓이다. 상황에 따라서, 보관소(120)에는 새로운 SYNACK 패킷 및/또는 리퀘스트 소켓이 더 저장될 수도 있고, 기존의 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k) 중 적어도 하나가 삭제될 수도 있다. 이들 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k)의 보관 또는 삭제는 처리부(130)에 동작에 의해 수행될 수 있다. 실시예에 따라서, 보관소는 멀티코어 시스템에서 확장성을 유지하기 위해 록리스(lock-less) 방법을 적용하여 최적화된 것일 수도 있다. 실시예에 따라서, 보관소(120)는 적어도 하나의 주기억장치(램 등)나 보조기억장치(플래시 메모리 장치나, 하드디스크 장치 등)를 이용하여 구현될 수 있다.

[0032] 처리부(130)는, TCP 연결 처리 장치(100)의 동작에 필요한 각종 연산 또는 제어 동작을 수행할 수 있으며, 예를 들어, 리슨 소켓(102)의 생성, SYNACK 패킷(191) 및 리퀘스트 소켓(192)의 탐색 및 결정, 보관소(120)의 관리 및/또는 통신부(101)에 대한 제어 등의 동작을 수행할 수 있다. 처리부(130)는, 저장 매체(미도시)에 저장된 프로그램(앱, 애플리케이션 또는 소프트웨어로 지칭 가능함)을 구동시켜 상술한 동작을 수행하는 것도 가능하다. 처리부(130)는, 실시예에 따라, 중앙 처리 장치(CPU: Central Processing Unit), 그래픽 처리 장치(GPU: Graphic Processing Unit), 마이크로 컨트롤러 유닛(MCU: Micro Controller Unit), 애플리케이션 프로세서(AP: Application Processor), 전자 제어 유닛(ECU: Electronic Controlling Unit), 및/또는 이외 각종 연산 및 제어 처리를 수행할 수 있는 적어도 하나의 전자 장치 등을 이용하여 구현 가능하다. 또한, 처리부(130)는, 중앙 처리 장치(CPU)의 다수의 코어(core, 도 3의 140-1 내지 140-n, 즉, 멀티 코어) 중 적어도 하나의 코어(140-1 내지 140-n 중 적어도 하나)를 이용하여 구현될 수도 있다.

[0033] 일 실시예에 따르면, 처리부(130)는, 보관소 처리부(131), 보관 처리부(132), 인스턴스 제거부(133) 및 재사용 판단부(134)를 포함할 수 있다.

[0034] 보관소 처리부(131)는 보관소(120) 내의 인스턴스의 검출 동작을 수행할 수 있다. 구체적으로 보관소 처리부(131)는, 먼저 수신한 SYN 패킷의 정보로부터 클라이언트 장치(10)에 대한 정보를 획득하고, 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 페어 정보와 일치하는 인스턴스(즉, SYNACK 패킷(191) 및 리퀘스트 소켓(192))가 보관소에 존재하는지를 확인할 수 있다. 만약 보관소(120)의 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k) 중에서, 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 페어 정보와 일치하는 SYNACK 패킷(191) 및 리퀘스트 소켓(192)이 존재한다면, 이를 사용할 SYNACK 패킷(191) 및 리퀘스트 소켓(192)로 하여 보관소(120)로부터 검출하고, TCP 연결 수립을 위해 재사용한다. 이후, TCP 연결 처리 장치(100)는,

도 1에 도시된 것처럼 클라이언트 장치(10)와 SYNACK 패킷과 ACK 패킷을 교환한다(p2, p3). 만약 TCP 연결이 정상적으로 생성되면, 해당 인스턴스(즉, SYNACK 패킷(191) 및 리퀘스트 소켓(192))는 인스턴스들을 보관소에 다시 보관된다. 다시 말해서, TCP 연결 과정에서, 메모리 할당자가 SYNACK 패킷(191) 및 리퀘스트 소켓(192)을 할당하고, TCP 연결 성립에 따라 SYNACK 패킷(191) 및 리퀘스트 소켓(192)이 해제되어 메모리 할당자로 반환되는 동작의 반복 수행이 불필요하게 된다. 이와 같이, 반복적인 할당, 해제 및 초기화 과정의 생략은 TCP 연결 수립 성능을 향상시킬 수 있게 된다. 일 실시예에 의하면, 보관소 처리부(131)는, SYNACK 패킷(191) 및 리퀘스트 소켓(192)의 검출을 위해서 소정의 해시 테이블을 사용할 수도 있다. 해시 테이블의 사용에 따라 보관소 처리부(131)는, 평균적으로 상수 시간 내에 SYNACK 패킷(191) 및 리퀘스트 소켓(192)을 찾을 수 있게 된다. 또한, 보관소 처리부(131)는 연결 리스트(199)를 참조하여 서로 대응하는 SYNACK 패킷(191) 및 리퀘스트 소켓(192)의 검출할 수도 있다. 연결 리스트(199)는 보관소(120)에 기록된 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k) 간의 대응 여부에 정보를 순차적으로 또는 비순차적으로 저장할 수 있으며, 보관소 처리부(131)는 이를 참조하여 보다 신속하게 상호 대응하는 SYNACK 패킷(191-1 내지 191-k) 및 리퀘스트 소켓(192-1 내지 192-k)을 검출할 수 있게 된다.

[0035] 보관 처리부(132)는 사용된 인스턴스(즉, 이전 또는 현재의 TCP 연결 수립 과정에서 이용된 SYNACK 패킷(일례로 191) 및 리퀘스트 소켓(일례로 192))의 보관 처리를 수행한다. 보다 구체적으로, 보관 처리부(132)는 해당 인스턴스(191, 192)를 보관할 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍을 결정하고, 보관소(120)에 어느 정도의 인스턴스(191, 192)를 보관할지를 결정할 수 있다. 일 실시예에 의하면, 보관 처리부(132)는 인스턴스(191, 192)를 보관할 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍을 결정하기 위해 임계 값을 이용할 수 있다. 이 경우, 보관 처리부(132)는 일정 시간 동안 미리 정의된 임계 값을 초과하여 반복 연결된 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍이 존재하면, 해당 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍이 반복적으로 연결된다고 판단하고, 해당 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍의 연결 수립을 위해 이용된 인스턴스(191, 192)를 저장할 수 있다. 또한, 보관 처리부(132)는 보관소(120)에 저장될 인스턴스의 개수, 즉 SYNACK 패킷(191-1 내지 191-k)의 개수 및/또는 리퀘스트 소켓의 개수(192-1 내지 192-k)를 결정하기 위해, 동시에 생성 가능한 TCP 연결의 개수를 관리할 수도 있다. 이 경우, 보관 처리부(132)는 보관소(120)에 저장될 인스턴스(191-1 내지 191-k, 192-1 내지 192-k)의 개수를, 일정 시간(일례로 타임 윈도우) 동안 동시에 수립 가능한 TCP 연결의 최대 개수로 유지할 수도 있다. 이는, 다수의 인스턴스(들)(191-1 내지 191-k, 192-1 내지 192-k) 각각의 상태는 재사용이 가능하나, 각 연결을 구별하는 상태 등은 공유가 되지 않으므로, 거의 동시에 다수의 TCP 연결 수립이 수행되고 있다면, 이들의 처리를 위해 TCP 연결 개수에 대응하는 개수의 인스턴스(191, 192)가 요청되기 때문이다.

[0036] 인스턴스 제거부(133)는 불필요한 인스턴스(즉, 더 이상 사용되지 않는 인스턴스)를 제거 대상으로 결정하고, 필요에 따라서 해당 인스턴스를 보관소(120)에서 제거할 수 있다. 인스턴스 제거부(133)는 특정한 적어도 한 쌍의 인스턴스(191-1 내지 191-k 중 적어도 하나, 192-1 내지 192-k 중 적어도 하나)에 대응하는 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍이 반복적으로 연결되지 않은 경우, 해당 쌍의 인스턴스(191-1 내지 191-k 중 적어도 하나, 192-1 내지 192-k 중 적어도 하나)를 제거 대상으로 판단할 수도 있다. 이를 위해 인스턴스 제거부(133)는 타이머를 이용하는 것도 가능하다. 구체적으로, 타이머는 일정한 주기마다 반복적으로 만료되고, 인스턴스 제거부(133)는 타이머의 만료 시마다 보관소(120)를 열람하여 어느 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍이 반복 연결 특성을 잃었는지 확인하고, 확인 결과를 기반으로 해당 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)의 쌍의 제거 여부를 결정할 수 있다. 예를 들어, 인스턴스 제거부(133)는 타이머가 만료되는 동안, 해당 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)가 상호 간에 전혀 연결되지 않았다면, 반복 연결 특성을 잃었다고 판단할 수도 있고, 또는 타이머가 2회 만료되는 동안, 해당 TCP 연결 처리 장치(100) 및 클라이언트 장치(10)가 상호 간에 전혀 연결되지 않았다면, 반복 연결 특성을 잃었다고 판단할 수도 있다. 타이머는 후술하는 다수의 코어(140-1 내지 140-n)마다 별도로 마련될 수도 있고, 또는 하나 또는 둘 이상의 타이머가 다수의 코어(140-1 내지 140-n)에 의해 전역적으로 이용될 수도 있다.

[0037] 재사용판단부(134)는 저장된 인스턴스, 일례로 SYNACK 패킷(191) 및/또는 리퀘스트 소켓(192)이 재사용 가능한지 여부를 판단할 수 있다. 재사용이 가능하다는 것은 해당 인스턴스(191, 192)가 값의 변경 없이 그대로 사용 가능한 상태를 의미하며, 구체적으로 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 사이의 연결이 반복적으로 수행되어야 하고, 이에 이용되는 각각의 인스턴스, 예를 들어 SYNACK 패킷(191) 및 리퀘스트 소켓(192)이 보관소(120)에 보관될 수 있어야 한다. 그런데, 특정한 데이터 등은 또 다른 TCP 연결에서 재사용이 가능하나, 다른 데이터 등은 재사용이 불가능할 수도 있다. 예를 들어, 프로토콜 패밀리, 인터넷 프로토콜(IP) 주소, 포트 번호 또는 라우팅 정보 등은 동일한 클라이언트 장치(10) 및 TCP 연결 처리 장치(100) 간의 연결에서 동일한 값

을 가지므로, 재사용이 가능하나, 패킷 시퀀스 번호나 타임스탬프 등은 연결 시마다 새로 초기화되어야 하므로 재사용이 불가능하다. 재사용판단부(134)는 이와 같이 반복 연결 여부나, 인스턴스(191, 192)의 재사용 가능 여부 등을 판단하고, 해당 인스턴스(191, 192)의 보관소(120) 저장 여부를 판단할 수 있다. 판단 결과는 보관 처리부(132)로 전달될 수도 있으며, 이 경우 보관 처리부(132)는 판단 결과를 기반으로 해당 인스턴스(191, 192)를 저장할 수 있다.

[0038] 처리부(130)는, 일 실시예에 있어서, 상술한 동작의 수행을 위해서 TCP 스택과 더불어, 소프트 인터럽트 요청(Soft IRQ: Soft Interrupt Request)를 이용할 수도 있다. 구체적으로 처리부(130)가 상술한 처리를 수행하기 위해서는, SYNACK 패킷(191) 및 리퀘스트 소켓(192) 구조에 대한 동기화가 요청되는데, 처리부(130)는 SYNACK 패킷(191) 및 리퀘스트 소켓(192)를 새로 추가하거나 삭제하는 경우에 있어서, 이러한 동기화를 위해 지역적 인터럽트 요청 해제(Local IRQ disabling)를 이용할 수도 있다. 통상 소프트 인터럽트 요청은 하드웨어 인터럽트에 의해 직접 수행되는 경우와, ksoftirqd 프로세스에 의해 처리되는 경우로 나뉠 수 있는데, ksoftirqd 프로세스는 사용자 프로세스이므로, 하드웨어 인터럽트로부터 선점될 수도 있다. 따라서, 동일 처리부(코어, 130: 130-1 내지 130-k) 내에서 최소한의 록(lock)이 발생될 수 있다. 처리부(130)는 새로운 인스턴스(191, 192)가 할당되는 경우, 코어(130-1 내지 130-k)가 점유한 지역적 인터럽트 요청을 단 시간 동안 비활성화함으로써, 이러한 록 발생을 방지할 수도 있다.

[0039] 일 실시예에 의하면, 처리부(130), 일례로 코어(130-1 내지 130-k)는 보관소(120)에서 SYNACK 패킷(191) 및 리퀘스트 소켓(192) 등의 인스턴스를 검출하거나 또는 다시 되돌려 보내는 경우, 동기화를 수행할 수도 있다. 만약 두 쌍의 인스턴스(일례로 (191, 192) 및 (191-1, 192-1))가 동시에 필요하다면, 이는 두 개의 TCP 연결 수립이 수행 중임을 의미하므로, 서로 상이한 인스턴스가 사용되어야 한다. 이를 위해 처리부(130)는 일반적인 원자적 동작(atomic operation) 기법을 이용할 수 있으며, 이에 따라 비지 웨이팅(busy waiting)이 불필요하게 되고, 동기화 비용이 절감될 수 있게 된다.

[0040] 일 실시예에 따르면, 도 3에 도시된 바와 같이, TCP 연결 처리 장치(100)는 복수의 보관소(120-1 내지 120-n)를 포함할 수도 있다. 이들 중 적어도 하나의 보관소(120-1 내지 120-n)는, 상술한 것처럼 적어도 하나의 SYNACK 패킷(191-1 내지 191-k) 및 이에 대응하는 적어도 하나의 리퀘스트 소켓(192-1 내지 192-k)을 저장할 수 있다. 한편, TCP 연결 제어 장치는, 복수의 처리부(130)를 포함할 수 있으며, 복수의 처리부(130) 각각은, 복수의 처리부(130) 각각에 대응하는 보관소(120)에 대한 인스턴스의 보관 또는 삭제 등의 동작을 제어할 수 있다. 각각의 처리부(130)는 상술한 바와 같이 각각의 코어(140-1 내지 140-n)를 이용하여 구현될 수도 있다. 이 경우, 일 실시예에 의하면, 각각의 보관소(120-1 내지 120-n)는 복수의 코어(140-1 내지 140-n) 각각에 대응하여 마련될 수도 있다. 상술한 바와 같이 보관소(120)를 이용함에 따라 할당, 해제 및/또는 초기화의 생략이 가능해지나, 이에 따른 전체적인 성능은 보관소(120)의 이용에 따른 성능 저하에 영향을 받는다. 보관소(120) 기반 동작에 따른 성능 저하 방지를 위해서 동일한 연결의 처리는 동일한 코어(140-1 내지 140-n)에 의해 수행되고, 이 경우 각각의 코어(140-1 내지 140-n)는 대응하는 보관소(120-1 내지 120-n)를 이용하여 해당 연결을 처리한다. 즉, 보관소(120-1 내지 120-n)는 코어(140-1 내지 140-n) 별로 분리된 후 코어(140-1 내지 140-n)에 고정된다. 일 실시예에 의하면, 이를 위해서 네트워크 인터페이스 컨트롤러(NIC: Network Interface Controller)의 수신측 스케일링(RSS: Receive-Side Scaling) 기술이 이용될 수 있다. 이 경우, SYN 패킷을 수신한 코어(140-1 내지 140-n 중 적어도 하나)는 해당 TCP 연결 처리가 모두 수행하면서, 사용하는 인스턴스를 다른 코어(140-1 내지 140-n 중 다른 적어도 하나)로 전달하거나 다른 코어(140-1 내지 140-n 중 다른 적어도 하나)로부터 다른 코어(140-1 내지 140-n 중 다른 적어도 하나)가 사용하는 인스턴스를 전달받지 않는다. 보관소(120)를 전역적으로 사용하는 경우, 인스턴스(191-1 내지 191-k, 192-1 내지 192-k)의 지역성을 저하시키고 록을 발생시킨다. 특히, 록의 발생은 코어 간의 비지 웨이팅으로 인한 딜레이 발생을 야기한다. 그러나, 상술한 바와 같이 서로 상이한 코어(140-1 내지 140-n)가 서로 상이한 보관소(120-1 내지 120-n)를 처리하는 경우, 캐시 일관성은 유지될 뿐만 아니라, 록이 사용되지 않으므로, 불필요한 성능 저하를 회피할 수 있게 된다.

[0041] 이하 도 4 내지 도 6을 참조하여, 전송 제어 프로토콜 연결 처리 방법의 일 실시예에 대해 설명한다.

[0042] 도 4는 전송 제어 프로토콜 연결 처리 방법의 일 실시예에 대한 흐름도이다.

[0043] 도 4에 도시된 바에 의하면, 전송 제어 프로토콜 연결 처리(TCP 연결 처리)를 위해서 리슨 소켓이 결정되고, 리슨 소켓을 통해 클라이언트 장치로부터 SYN 패킷을 수신하게 된다(400).

[0044] SYN 패킷이 수신되면, 이에 응하여 보관소가 열람될 수 있다(404). 보관소 내에는 사전에 하나 또는 둘 이상의 쌍의 인스턴스(SYNACK 패킷 및 리퀘스트 소켓)가 저장되어 있을 수 있다.

- [0045] 보관소의 열람을 통해, SYN 패킷에 대응하는 인스턴스(SYNACK 패킷 및 리퀘스트 소켓)가 보관소 내에 존재하는지 여부가 판단될 수 있다(406). 만약 보관소 내에 해당 SYNACK 패킷 및 리퀘스트 소켓이 존재한다면(406의 예), 해당 SYNACK 패킷 및 리퀘스트 소켓은, 수신한 SYN 패킷을 전송한 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결을 위해 이용된다(408). 반대로 보관소 내에 해당 SYNACK 패킷 및 리퀘스트 소켓이 존재하지 않는다면(406의 아니오), 새로운 인스턴스(SYNACK 패킷 및 리퀘스트 소켓)가 메모리 할당자 등에 의해 생성될 수도 있다.
- [0046] 보관소에 저장되었거나 또는 새로 생성된 SYNACK 패킷은 클라이언트 장치로 전달될 수 있다(410).
- [0047] 클라이언트 장치는 SYNACK 패킷의 수신에 응하여 ACK 패킷을 생성하여 발송하고, TCP 연결 처리 장치는 리퀘스트 소켓을 통해 발송된 ACK 패킷을 수신한다(412).
- [0048] 이에 따라 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결이 수립된다(414).
- [0049] 도 5는 반복 사용되는 인스턴스의 보관 여부를 결정하기 위한 과정의 일 실시예를 설명하기 위한 흐름도이다.
- [0050] 한편, TCP 연결 처리 장치는, 특정한 클라이언트 장치 및 TCP 연결 처리 장치 간의 반복된 연결에 사용되는 인스턴스의 보관 여부를 도 5에 도시된 바와 같이 결정할 수 있다.
- [0051] 구체적으로 예를 들어, 먼저 특정 클라이언트 장치 및 TCP 연결 처리 장치 사이에 TCP 연결이 수립되면, 연결 수립 개수가 계수된다(412).
- [0052] 계수 결과와 미리 정의된 임계 값이 상호 비교되고(422), 비교 결과 계수 결과가 임계 값보다 크다면(422의 예), 특정 클라이언트 장치 및 TCP 연결 처리 장치 사이에 TCP 연결이 반복 특성을 갖는 것으로 판단된다(423).
- [0053] 이 경우, 특정 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결에 이용된 인스턴스(즉, SYNACK 패킷 및 리퀘스트 소켓)은 보관소에 저장된다(424).
- [0054] 반대로 비교 결과 계수 결과가 임계 값보다 작다면(422의 아니오), 상술한 과정(423, 424)은 수행되지 않는다.
- [0055] 상술한 과정(421 내지 425)은 일정 시간(타임 윈도우)이 경과될 때까지 반복될 수 있다. 만약 특정한 특정 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결이 반복적이라고 판단되는 경우(즉, 계수 결과가 임계 값보다 큰 경우(422의 예))에는 반복이 중단될 수도 있다(425). 다시 말해서, 특정 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결이 임계 값보다 작다면(422의 아니오), 연결 생성 및 계수 과정 내지 임계 값 비교 결과에 따른 인스턴스의 보관소 저장 과정(421 내지 424)는 계속해서 반복 수행될 수 있다.
- [0056] 도 6은 사용되지 않는 인스턴스를 제거하기 위한 과정의 일 실시예를 설명하기 위한 흐름도이다.
- [0057] 또한, TCP 연결 처리 장치는, 도 6에 도시된 것처럼, 기존에 보관소에 보관되고, 특정한 클라이언트 장치 및 TCP 연결 처리 장치 간의 연결에 사용되는 인스턴스의 제거 여부를 결정할 수도 있다.
- [0058] 구체적으로 예를 들면, 먼저 타이머가 구동을 개시하고, 미리 정의된 바에 따라서 소정의 시간(사용자 또는 설계자 등에 의해 정해질 수 있음)이 경과된 이후에 만료된다(431).
- [0059] 타이머가 만료되면(실시예에 따라 타이머가 만료될 때마다), 적어도 하나의 특정 클라이언트 장치 및 TCP 연결 처리 장치 간에 추가적이거나 반복적인 연결이 존재하였는지 여부가 판단될 수 있다(432).
- [0060] 만약 타이머가 만료될 때까지 적어도 하나의 특정 클라이언트 장치 및 TCP 연결 처리 장치 사이에 추가적 또는 반복적 연결이 더 이상 존재하지 않았다면(432의 아니오), 추가적 또는 반복적 연결이 부재는 계수되고, 계수 결과(즉, 동작 기간 동안에 양 장치 사이에 연결이 없었던 타이머의 동작 회수)가 사용자나 설계자 등에 의해 미리 정의된 기준 회수(예를 들어, 2회)를 초과하는지 여부가 판단될 수 있다(433).
- [0061] 추가적 또는 반복적 연결이 부재한 회수가 미리 정의된 기준 회수를 초과하였다면(433의 예), 특정 클라이언트 장치 및 TCP 연결 처리 장치의 TCP 연결 수립을 위해 이용되었던 인스턴스(즉, SYNACK 패킷 및 리퀘스트 소켓)를 보관소에서 제거한다(434). 이에 따라 불필요한 인스턴스를 제거하여 보관소를 최적으로 관리할 수 있게 된다.
- [0062] 반대로 추가적 또는 반복적 연결이 부재한 회수가 미리 정의된 기준 회수를 초과하지 않았다면(433의 아니오), 특정 클라이언트 장치 및 TCP 연결 처리 장치의 TCP 연결 수립을 위해 이용되었던 인스턴스는 계속 보관소에 유지된다.

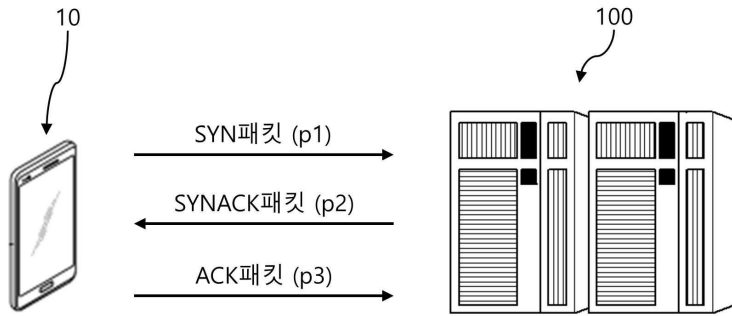
- [0063] 한편, 만약 타이머가 동작하는 동안 적어도 하나의 특정 클라이언트 장치 및 TCP 연결 처리 장치가 서로 연결된 바 있다면(432의 예), 상술한 기준 회수 초과 판단(433)이나 인스턴스 제거 동작(434)는 수행되지 않는다.
- [0064] 상술한 과정은 계속해서 반복될 수 있다(435). 따라서, 타이머는 계속해서, 주기적으로 또는 임의적으로 동작 개시 및 만료를 수행하고(431), 이에 따라 특정 클라이언트 장치 및 TCP 연결 처리 장치 사이의 반복 연결 여부가 판단되고(432), 이에 따라 인스턴스의 제거 여부가 판단될 수 있다(433, 434).
- [0065] 상술한 실시예에 따른 전송 제어 프로토콜 연결 처리 방법은, 컴퓨터 장치에 의해 구동될 수 있는 프로그램의 형태로 구현될 수 있다. 프로그램은, 명령어, 라이브러리, 데이터 파일 및/또는 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있으며, 기계어 코드나 고급 언어 코드를 이용하여 설계 및 제작된 것일 수 있다. 프로그램은 상술한 방법을 구현하기 위하여 특별히 설계된 것일 수도 있고, 컴퓨터 소프트웨어 분야에서 통상의 기술자에게 기 공지되어 사용 가능한 각종 함수나 정의를 이용하여 구현된 것일 수도 있다. 또한, 여기서, 컴퓨터 장치는, 프로그램의 기능을 실현 가능하게 하는 프로세서나 메모리 등을 포함하여 구현된 것일 수 있으며, 필요에 따라 통신 장치를 더 포함할 수도 있다. 상술한 전송 제어 프로토콜 연결 처리 방법을 구현하기 위한 프로그램은, 컴퓨터 등의 장치에 의해 판독 가능한 기록 매체에 기록될 수 있다. 컴퓨터에 의해 판독 가능한 기록 매체는, 예를 들어, 롬, 램, SD카드 또는 플래시 메모리(일레로 솔리드 스테이트 드라이브(SSD) 등)와 같은 반도체 저장 매체나, 하드 디스크 또는 플로피 디스크 등과 같은 자기 디스크 저장 매체나, 콤팩트 디스크 또는 디브이디 등과 같은 광 기록 매체나, 또는 플롭티컬 디스크 등과 같은 자기-광 기록 매체 등과 같이 컴퓨터 등의 장치의 호출에 따라 실행되는 하나 이상의 프로그램을 일시적 또는 비일시적으로 저장 가능한 적어도 한 종류의 물리적 저장 매체를 포함할 수 있다.
- [0066] 이상 전송 제어 프로토콜 연결 처리 장치 및 전송 제어 프로토콜 연결 처리 방법의 여러 실시예에 대해 설명하였으나, 전송 제어 프로토콜 연결 처리 장치 또는 전송 제어 프로토콜 연결 처리 방법은 오직 상술한 실시예에 한정되는 것은 아니다. 해당 기술 분야에서 통상의 지식을 가진 자가 상술한 실시예를 기초로 수정 및 변형하여 구현할 수 있는 다른 다양한 장치나 방법 역시 상술한 전송 제어 프로토콜 연결 처리 장치 또는 전송 제어 프로토콜 연결 처리 방법의 일 실시예가 될 수 있다. 예를 들어, 설명된 방법(들)이 설명된 바와 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소(들)가 설명된 바와 다른 형태로 결합, 연결 또는 조합되거나 다른 구성 요소 또는 균등물 등에 의하여 대체 또는 치환되더라도, 상술한 전송 제어 프로토콜 연결 처리 장치 및/또는 전송 제어 프로토콜 연결 처리 방법의 일 실시예가 될 수 있다.

### 부호의 설명

- [0067] 10: 클라이언트 장치    100: 전송 제어 프로토콜 연결 장치
- 102: 리슨 소켓    120: 보관소
- 130: 처리부    191: SYNACK 패킷
- 192: 리퀘스트 소켓

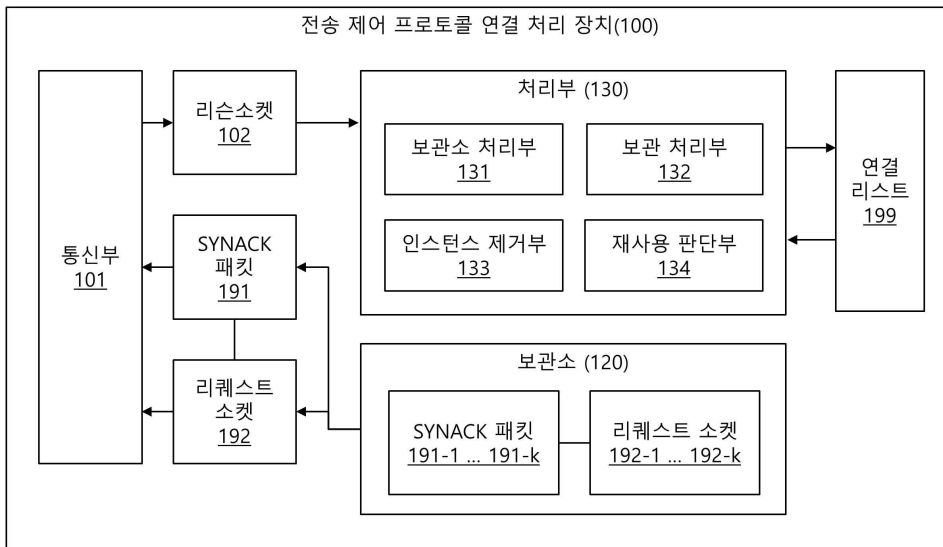
도면

도면1

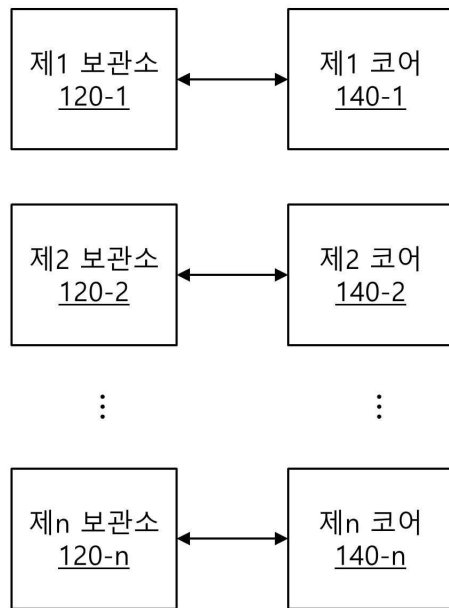


1

도면2

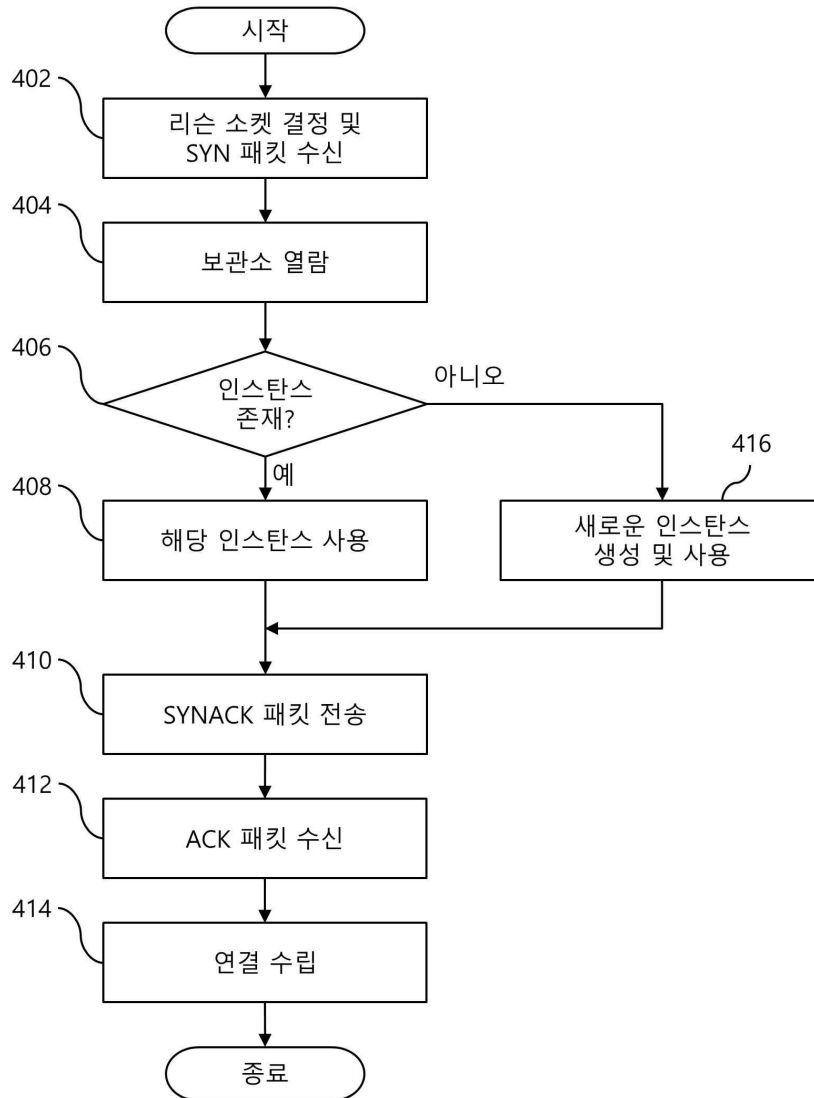


도면3

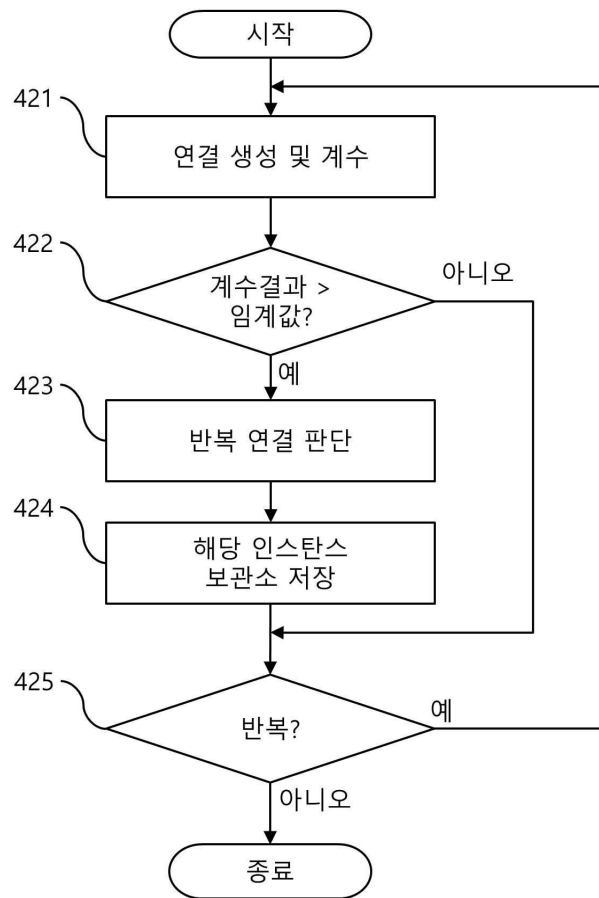


100

도면4



도면5



도면6

