(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년03월03일
(11) 등록번호 10-1018575
(24) 등록일자 2011년02월22일

(51) Int. Cl.
H04L 12/56 (2006.01) *H04L 12/28* (2006.01)
(21) 출원번호 10-2007-7008944
(22) 출원일자(국제출원일자) 2005년10월19일
심사청구일자 2008년12월26일
(85) 번역문제출일자 2007년04월19일
(65) 공개번호 10-2007-0085272
(43) 공개일자 2007년08월27일
(86) 국제출원번호 PCT/US2005/037941
(87) 국제공개번호 WO 2006/045057
국제공개일자 2006년04월27일
(30) 우선권주장
10/969,376 2004년10월19일 미국(US)
(56) 선행기술조사문헌
International Conference on Computer Design, Gregory L. Frazier et al., 1989 oct, THE DESIGN AND IMPLEMENTATION OF A MULTI-QUEUE BUFFER FOR VLSI COMMUNICATION SWITCHES
전체 청구항 수 : 총 28 항

(73) 특허권자
엔비디아 코포레이션
미국 캘리포니아 95050 산타 클라라 산 토마스 익스프레스웨이 2701
(72) 발명자
미나미 존 시게토
미국 96817 하와이주 호놀룰루 누아누 애비뉴 1212 넘버1206
우예시로 로빈 와이
미국 96734 하와이주 카일루아 켈레위나 스트리트 1234
(뒷면에 계속)
(74) 대리인
양영준, 백만기

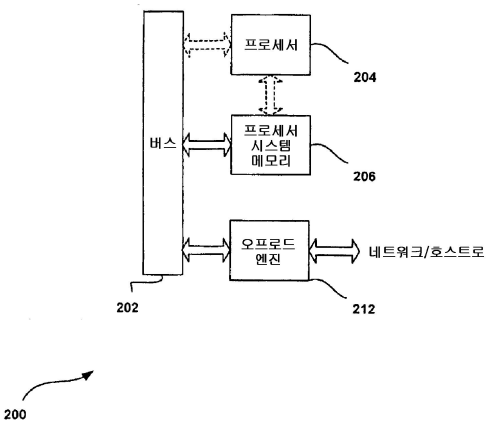
심사관 : 이성영

(54) RX FIFO 버퍼를 사용하여 고속 네트워크애플리케이션에서 RX 패킷을 프로세싱하는 시스템 및방법

(57) 요 약

네트워크를 통해 수신되는 패킷을 프로세싱하는 시스템 및 방법을 제공한다. 일반적인 사용에서, 데이터 패킷 및 제어 패킷은 네트워크를 통해 수신된다. 또한 데이터 패킷은 제어 패킷과 병렬로 프로세싱된다.

대 표 도 - 도2



(72) 발명자

오오이 티엔 이

미국 96815 하와이주 호놀룰루 알라 모아나 블러바드 1920넘버604

존슨 마이클 워드

미국 94550 캘리포니아주 리버모어 노팅엄 서클 482

카누리 므루둘라

미국 95050 캘리포니아주 산타 클라라 린덴 드라이브 872

특허청구의 범위

청구항 1

네트워크를 통해 수신되는 패킷들을 프로세싱하는 방법으로서,

데이터 패킷들 및 제어 패킷들을 네트워크를 통해 수신하는 단계; 및

상기 제어 패킷들의 프로세싱과 병렬로 상기 데이터 패킷들을 프로세싱하는 단계를 포함하고,

상기 데이터 패킷들에 태그 정보가 프리펜딩되고(prepended),

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 데이터 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 데이터 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 2

제1항에 있어서,

상기 제어 패킷들은 제1 프로세싱 경로를 이용하여 프로세싱되고,

상기 데이터 패킷들은 상기 제1 프로세싱 경로와는 별도의 제2 프로세싱 경로를 이용하여 프로세싱되는, 패킷 프로세싱 방법.

청구항 3

제1항에 있어서,

상기 데이터 패킷들 및 상기 제어 패킷들을 프로세싱하는 단계는 상기 데이터 패킷들 및 상기 제어 패킷들과 연관된 정확한 소켓 제어 블록을 식별하기 위해 탐색표(look-up table) 및 캐시를 병렬로 이용하는 단계를 포함하는, 패킷 프로세싱 방법.

청구항 4

제1항에 있어서,

상기 데이터 패킷들을 프로세싱하는 단계는 상기 패킷들과 연관되는 정확한 소켓 제어 블록을 식별하기 위해 실질적인 이중 로직(substantially duplicate logic)을 병렬로 이용하는 단계를 포함하는, 패킷 프로세싱 방법.

청구항 5

제1항에 있어서,

상기 데이터 패킷들 및 상기 제어 패킷들과 연관된 제어 블록들은 후속하는 데이터 패킷들 및 제어 패킷들의 프로세싱과 병렬로 업데이트되는, 패킷 프로세싱 방법.

청구항 6

삭제

청구항 7

제1항에 있어서,

상기 태그 정보는 수신기(RX) 선입선출(FIFO) 버퍼에 버퍼링되는 동안에 상기 데이터 패킷들에 프리펜딩되는, 패킷 프로세싱 방법.

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

제1항에 있어서,

상기 태그 정보는 상기 대응 패킷의 타입을 포함하는, 패킷 프로세싱 방법.

청구항 17

제1항에 있어서,

상기 태그 정보는 상기 대응 패킷과 연관되는 상태 정보를 포함하는, 패킷 프로세싱 방법.

청구항 18

제1항에 있어서,

상기 태그 정보는 상기 대응 패킷과 연관되는 제어 정보를 포함하는, 패킷 프로세싱 방법.

청구항 19

제1항에 있어서,

상기 데이터 패킷들은 수신기(RX) 선입선출(FIFO) 버퍼를 이용해 버퍼링되는, 패킷 프로세싱 방법.

청구항 20

제1항에 있어서,

상기 데이터 패킷들 및 상기 제어 패킷들을 프로세싱하는 단계는 상기 데이터 패킷들 및 상기 제어 패킷들과 연관된 정확한 소켓 제어 블록을 식별하기 위해 탐색 모드 동안 캐시를 이용하는 단계를 포함하고, 상기 정확한 소켓 제어 블록이 상기 캐시에 있지 않은 경우, 상기 정확한 소켓 제어 블록은 메인 메모리로부터 회수되고(retrieved), 상기 탐색 모드에서 상기 캐시에 저장되지 않는, 패킷 프로세싱 방법.

청구항 21

데이터 패킷들 및 제어 패킷들을 네트워크를 통해 수신하기 위해 버스와 통신하고, 상기 제어 패킷들과 병렬로 상기 데이터 패킷들을 프로세싱하는 오프로드 엔진을 포함하는 서브 시스템으로서,

상기 데이터 패킷들에 태그 정보가 프리펜딩되고(prepended),

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 데이터 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 데이터 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하도록(strips) 동작가능한, 서브 시스템.

청구항 22

버스;

상기 버스와 통신하는 메모리;

상기 버스와 통신하는 프로세서; 및

데이터 패킷들 및 제어 패킷들을 네트워크를 통해 수신하기 위해 버스와 통신하고, 상기 제어 패킷들과 병렬로 상기 데이터 패킷들을 프로세싱하는 오프로드 엔진을 포함하는 시스템으로서,

상기 데이터 패킷들에 태그 정보가 프리펜딩되고(prepended),

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 데이터 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 데이터 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하도록(strips) 동작가능한, 시스템.

청구항 23

제22항에 있어서,

상기 시스템은 범용 컴퓨터를 포함하는, 시스템.

청구항 24

제22항에 있어서,

상기 시스템은 게임 콘솔을 포함하는, 시스템.

청구항 25

제22항에 있어서,

상기 프로세서는 중앙 프로세싱 유닛, 그래픽 처리 유닛, 칩셋에 포함되는 복수의 집적 회로 중 하나를 포함하는 그룹으로부터 선택되는, 시스템.

청구항 26

제22항에 있어서,

상기 시스템은 회로 보드를 포함하는, 시스템.

청구항 27

네트워크를 통해 수신된 패킷들을 프로세싱하는 방법으로서,

네트워크를 통해 패킷들을 수신하는 단계; 및

정확한 소켓 제어 블록의 식별을 가속화하기 위해 탐색표 및 캐시를 병렬로 이용하여 상기 패킷들과 연관된 상기 정확한 소켓 제어 블록을 식별하는 단계를 포함하고,

상기 패킷들과 연관된 제어 블록들은 후속하는 패킷들의 프로세싱과 병렬로 업데이트되고,

상기 패킷들에 태그 정보가 프리펜딩되고,

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 패킷들 중 하나로부터 패킷 버퍼

헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 28

네트워크를 통해 수신된 패킷들을 프로세싱하는 방법으로서,

네트워크를 통해 패킷들을 수신하는 단계; 및

정확한 소켓 제어 블록의 식별을 가속화하기 위해 실질적인 이중 로직을 병렬로 이용하여 상기 패킷들과 연관된 상기 정확한 소켓 제어 블록을 식별하는 단계를 포함하고,

상기 패킷들에 태그 정보가 프리펜딩되고,

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 패킷들 중 하나로부터 패킷 버퍼 헤더를 파싱(parsing)하고 제거(stripping)하는, 패킷 프로세싱 방법.

청구항 29

네트워크를 통해 수신된 데이터 패킷들을 프로세싱하는 방법으로서,

네트워크를 통해 데이터 패킷들을 수신하는 단계; 및

상기 데이터 패킷들을 프로세싱하는 단계를 포함하고,

상기 데이터 패킷들이 수신기(RX) 선입선출(FIFO) 버퍼에 저장된 이후, 상기 데이터 패킷들에 태그 정보가 프리펜딩되고,

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 데이터 패킷들 중 하나가 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 데이터 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 30

네트워크를 통해 수신된 패킷들을 프로세싱하는 방법으로서,

네트워크를 통해 제1 패킷을 수신하는 단계;

네트워크를 통해 상기 제1 패킷을 프로세싱하는 단계;

상기 제1 패킷과 연관된 제어 블록을 업데이트하는 단계; 및

상기 제1 패킷과 연관된 상기 제어 블록의 상기 업데이트가 적어도 개시된 이후 상기 업데이트가 끝나기 전에, 제2 패킷의 프로세싱을 적어도 개시하는 단계를 포함하고,

상기 제1 패킷에 태그 정보가 프리펜딩되고,

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 제1 패킷이 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 제1 패킷으로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 31

네트워크를 통해 수신된 패킷들을 프로세싱하는 방법으로서,

전단 모듈을 이용해 수신 패킷들을 프로세싱하는 단계; 및

후단 모듈을 이용해 수신 패킷들을 프로세싱하는 단계를 포함하고,

수신기(RX) 선입선출(FIFO) 버퍼가 상기 전단 모듈과 상기 후단 모듈 사이에 연결되어 상기 모듈들 사이에 경계(boundary)를 제공하고,

상기 수신 패킷들과 연관된 제어 블록들은 후속하는 수신 패킷들의 프로세싱과 병렬로 업데이트되고,

상기 수신 패킷들에 태그 정보가 프리펜딩되고,

상기 태그 정보는 대응 패킷과 연관되는 소켓 처리를 포함하고,

상기 수신 패킷들 중 하나가 완전히 버퍼링되고 나면 후단 모듈이 상기 수신 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 32

네트워크를 통해 수신되는 패킷들을 프로세싱하는 방법으로서,

데이터 패킷들 및 제어 패킷들을 네트워크를 통해 수신하는 단계; 및

상기 제어 패킷들의 프로세싱과 병렬로 상기 데이터 패킷들을 프로세싱하는 단계를 포함하고,

상기 데이터 패킷들은 수신기(RX) 선입선출(FIFO) 버퍼를 이용해 버퍼링되고,

상기 데이터 패킷들 중 하나가 상기 수신기(RX) 선입선출(FIFO) 버퍼를 이용해 완전히 버퍼링되고 나면 수신기(RX) 후단 모듈이 상기 데이터 패킷들 중 하나로부터 패킷 버퍼 헤더를 분석하고(parse) 제거하는(strips), 패킷 프로세싱 방법.

청구항 33

제32항에 있어서,

상기 데이터 패킷들 중 하나는 상기 패킷 버퍼 헤더가 상기 데이터 패킷들 중 하나로부터 분석되고(parsed) 제거된(stripped) 이후 재정렬되는, 패킷 프로세싱 방법.

청구항 34

제4항에 있어서,

상기 패킷들과 연관된 상기 정확한 소켓 제어 블록은 연결 시도 또는 연결을 추적하는데 이용되는 정보를 포함하는, 패킷 프로세싱 방법.

청구항 35

제4항에 있어서,

상기 데이터 패킷들에 상기 태그 정보를 프리펜딩한 결과로서, 상기 패킷들과 연관된 상기 정확한 소켓 제어 블록이 식별되는 동안에, 상기 패킷들이 저장되는, 패킷 프로세싱 방법.

청구항 36

제1항에 있어서,

상기 데이터 패킷들을 프로세싱하는 단계는, 상기 패킷들과 연관된 정확한 소켓 제어 블록을 식별하기 위해 실질적인 이중 로직, 탐색표와 소켓 제어 블록(CB) 캐시를 포함하는 다중 데이터 구조를 병렬로 이용하는 단계를 포함하는, 패킷 프로세싱 방법.

청구항 37

제36항에 있어서,

상기 패킷과 연관된 상기 정확한 소켓 제어 블록은, 상기 패킷과 연관된, 매칭되는 소켓 제어 블록(CB) 해시가 상기 소켓 제어 블록(CB) 캐시 내 존재하는지를 결정하기 위해 상기 소켓 제어 블록(CB) 캐시의 해시 연관 테이블에 문의함으로써 식별되는, 패킷 프로세싱 방법.

명 세 서

[0001] 본 발명의 기술분야

[0002] 본 발명은 네트워크 통신에 관한 것으로, 더 상세하게는 수신 (RX) 패킷의 프로세싱에 관한 것이다.

[0003] 발명의 배경

[0004] 전송 오프로드 엔진 (transport offload engine; TOE) 은 스루풋을 최적화하고, 프로세서 사용률을 줄이기 위해 고속 시스템에서 널리 사용되고 있는 기술을 포함한다. TOE 컴포넌트는 종종 네트워크 인터페이스 카드 (NIC), 호스트 버스 어댑터 (HBA), 머더보드와 같은 다양한 인쇄 회로 기판 중 하나에 또는 원하는 임의의 다른 오프로딩 콘텍스트에 포함된다.

[0005] 최근 몇 년간, 시스템에서의 통신 속도는 프로세서의 속도보다 더 신속하게 증가하였다. 이것은 입출력 (I/O) 병목을 생성하였다. 통상적으로, I/O 가 아닌 연산용으로 주로 설계된 프로세서는 네트워크를 통해 흐르는 데이터를 따라잡을 수 없다. 결과적으로, 데이터 흐름은 네트워크 속도보다 더 느린 레이트로 프로세싱된다. TOE 기술은 프로세서 및/또는 I/O 서브시스템으로부터의 부담을 제거함으로써 (즉, 오프로딩함으로써) 이 문제를 해결한다.

[0006] 종종 TOE 로 오프로딩되는 프로세싱의 하나의 타입은 전송 제어 프로토콜 (TCP) 패킷 프로세싱을 포함한다. TCP 는 인터넷 프로토콜 (IP) 과 함께 사용되는 규정 세트 (프로토콜) 이며, 인터넷을 통해 컴퓨터 사이에서 메시지 단위 형태로 데이터를 전송한다. IP 는 데이터의 실제 전달을 처리하는 것을 담당하고, TCP 는 인터넷을 통해 효율적으로 라우팅하기 위해 메시지가 분할된 각각의 패킷을 추적하는 것을 담당한다.

[0007] 고속 네트워크 상에서 TCP 패킷을 처리하는 것은 많은 프로세싱을 요구한다. 패킷은 아웃-오브-시퀀스로 (out of sequence) 도달할 수도 있으며, 따라서 데이터가 인-시퀀스로(in sequence) 애플리케이션에 전달되려면, 그 패킷은 저장되어야 한다. 또한, 수신 패킷의 프로세싱은 네트워크 접속의 라인 레이트 (line rate) 를 따라잡을 수 있어야만 한다.

[0008] 따라서, TCP 및 다른 패킷 프로세싱을 가속하는 기술이 필요하다.

[0009] 발명의 요약

[0010] 네트워크를 통해 수신되는 패킷을 프로세싱하는 시스템 및 방법을 제공한다. 일반적인 사용에서, 데이터 패킷 및 제어 패킷은 네트워크를 통해 수신된다. 또한 데이터 패킷은 제어 패킷과 병렬로 프로세싱된다.

[0011] 일 실시형태에 있어서, 제어 패킷은 제 1 프로세싱 경로를 사용하여 프로세싱될 수도 있으며, 데이터 패킷은 제 1 프로세싱 경로와는 별도의 제 2 프로세싱 경로를 사용하여 프로세싱될 수도 있다.

[0012] 다른 실시형태에 있어서, 패킷의 프로세싱은 패킷과 연관되는 정확한 소켓 제어 블록을 식별하기 위해 탐색표 (look-up table) 와 캐시를 병렬로 사용하는 것을 포함할 수도 있다. 유사하게, 데이터 패킷의 프로세싱은 패킷과 연관되는 정확한 소켓 제어 블록을 식별하기 위해 실질적인 이중 로직 (duplicate logic) 을 병렬로 사용하는 것을 포함할 수도 있다.

[0013] 또 다른 실시형태에 있어서, 패킷과 연관되는 제어 블록은 후속 패킷의 프로세싱과 동시에 업데이트될 수도 있다.

[0014] 추가적인 옵션으로서, 태그 정보가 데이터 패킷에 프리펜딩될 (prepended) 수도 있다. 이런 태그 정보는 수신기 (RX) 선입선출 (FIFO) 버퍼에서 버퍼링되는 동안에 데이터 패킷에 더 프리펜딩될 수도 있다. 선택적으로, 태그 정보는 대응 패킷의 타입, 대응 패킷에 연관되는 소켓 처리, 대응 패킷과 연관되는 상태 정보, 및/또는 대응 패킷과 연관되는 제어 정보를 포함할 수도 있다.

[0015] 일반적인 사용에서, 데이터 패킷은 RX FIFO 버퍼를 사용하여 버퍼링될 수도 있다.

[0016] 도면의 간단한 설명

[0017] 도 1 은 일 실시형태에 따른 네트워크 시스템을 나타낸다.

[0018] 도 2 는 일 실시형태가 구현될 수도 있는 아키텍처를 나타낸다.

[0019] 도 3 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 아키텍처의 특정 실시예를 나타낸다.

[0020] 도 4 는 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 전단 (front-end) 모듈을 나타낸다.

[0021] 도 5 는 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 수신기 후단 (back-end) 모듈을 나타낸다.

[0022] 도 6 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 전단 방법을 나타낸다.

[0023] 도 7 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 후단 방법을 나타낸다.

[0024] 도 8 은 일 실시형태에 따른, 수신 제어 패킷을 프로세싱하는 예시적인 제어 패킷 프로세싱 방법을 나타낸다.

[0025] 발명의 상세한 설명

[0026] 도 1 은 일 실시형태에 따른 네트워크 시스템 (100) 을 나타낸다. 도시된 바와 같이, 네트워크 (102) 가 제공된다. 이 네트워크 시스템 (100) 의 콘텍스트에서, 네트워크 (102) 는 로컬 영역 네트워크 (LAN), 및 인터넷 등과 같은 광역 네트워크 (WAN) 를 포함하지만 이에 제한되지 않고, 임의의 형태를 취할 수도 있다.

[0027] 네트워크 (102) 를 통해 통신할 수 있는 로컬 호스트 (104) 및 원격 호스트 (106) 가 네트워크 (102) 에 연결된다. 이 설명의 콘텍스트에서, 이런 호스트 (104, 106) 는 웹서버, 저장 디바이스 또는 서버, 데스크탑 컴퓨터, 랩탑 컴퓨터, 휴대용 컴퓨터, 프린터 또는 임의의 다른 타입의 하드웨어/소프트웨어를 포함할 수도 있다. 임의의 다른 예시되지 않은 디바이스 뿐 아니라 전술한 컴포넌트 각각은 하나 이상의 네트워크를 경유하여 상호 접속될 수도 있다.

[0028] 도 2 는 일 실시형태가 구현될 수도 있는 예시적인 아키텍처 (200) 를 나타낸다. 일 실시형태에 있어서, 아키텍처 (200) 는 도 1 의 호스트 (104, 106) 중 하나를 나타낼 수도 있다. 그러나, 물론, 아키텍처 (200) 는 임의의 원하는 콘텍스트에서 구현될 수도 있다.

[0029] 예를 들어, 아키텍처 (200) 는 일반적인 컴퓨터 시스템, 회로 기판 시스템, 오락 전용 게임 콘솔 시스템, 셋-탑 박스, 라우터, 네트워크 시스템, 저장 시스템, 애플리케이션-특정 시스템, 또는 네트워크 (102) 와 연관되는 임의의 다른 원하는 시스템에서 구현될 수도 있다.

[0030] 도시된 바와 같이, 아키텍처 (200) 는 버스 (202) 를 통해 연결되는 복수의 컴포넌트를 포함한다. 데이터를 프로세싱하는 하나 이상의 프로세서 (204) 가 포함된다. 프로세서 (204) 는 임의의 형태를 취할 수도 있으며, 일 실시형태에 있어서 중앙 프로세싱 유닛 (CPU), 그래픽 모듈, 칩셋 (즉, 관련 기능을 수행하는 유닛으로서 동작하도록 설계되거나 판매되는 집적 회로 그룹 등), 이들의 조합물, 또는 임의의 다른 해당 집적 회로의 형태를 취할 수도 있다. 그래픽 모듈의 예에서, 이런 집적 회로는 변환 모듈, 발광 모듈, 및 래스터화 모듈을 포함할 수도 있다. 각각의 전술한 모듈은 그래픽 프로세싱 유닛 (GPU) 을 형성하기 위해 단일 반도체 플랫폼 상에 위치될 수도 있다.

[0031] 데이터를 저장하기 위해 프로세서 (204) 와 통신하면서 상주하는 프로세서 시스템 메모리 (206) 가 더 포함된다. 이런 프로세서 시스템 메모리 (206) 는 온-보드 또는 오프-보드 랜덤 액세스 메모리(RAM), 하드 디스크 드라이브, 착탈형 저장 드라이브(즉, 플로피 디스크 드라이브, 자기 테이프 드라이브, 컴팩트 디스크 드라이브 등), 및/또는 임의의 다른 타입의 원하는 데이터 저장 가능 메모리의 형태를 취할 수도 있다.

[0032] 일반적인 사용에서, 프로그램 또는 제어 로직 알고리즘은 프로세서 시스템 메모리 (206) 에 선택적으로 저장될 수도 있다. 이런 프로그램은, 실행되는 경우, 아키텍처 (200) 가 다양한 기능을 수행하게 할 수 있다. 물론, 아키텍처 (200) 는 간단히 하드 와이어드 게이트-레벨 회로에서 직접 구현될 수도 있다.

[0033] 또한, 프로세서 (204) 및 네트워크 (예를 들어, 도 1 의 네트워크 (102) 참조) 와 통신하는 전송 오프로드 엔진 (212) 이 도시되어 있다. 일 실시형태에 있어서, 전송 오프로드 엔진 (212) 은 버스 (202) 를 통해 프로세서 (204) 와의 통신 상태를 유지할 수도 있다. 그러나, 물론, 전송 오프로드 엔진 (212) 은 통신을 제공하는 임의의 메커니즘을 통해 프로세서 (204) 와의 통신 상태를 유지할 수도 있다. 전송 오프로드 엔진 (212) 은 전송 (즉, TCP/IP) 오프로드 엔진 (TOE), 시스템, 또는 네트워크에서 전송되는 데이터를 관리할 수 있는 임의의 집적 회로를 포함할 수도 있다.

[0034] 전술한 컴포넌트 간의 통신을 제공하는 단일 버스 (202) 가 도시되어 있지만, 임의의 수의 버스 (또는 다른 통신 메커니즘) 가 컴포넌트 간의 통신을 제공하도록 사용될 수도 있다. 단지 예로써, 추가적인 버스가 프로세서 (204) 와 프로세서 시스템 메모리 (206) 간의 통신을 제공하도록 사용될 수도 있다. 또한, 일 실시형태에 있어서, 도 2 에 도시된 임의의 2 개 이상의 컴포넌트가 단일 집적 회로 상에 집적될 수도 있다.

[0035] 동작하는 동안에, 전송 오프로드 엔진 (212), 프로세서 (204) 및/또는 소프트웨어는 네트워크 (즉, 예를 들어, 도 1 의 네트워크 (102) 등을 참조) 를 통해 수신되는 패킷을 프로세싱하도록 동작한다. 일 실시형태에 의하면, 데이터 패킷 및 제어 패킷이 네트워크를 통해 수신된다. 또한, 데이터 패킷은 제어 패킷의 프로세싱과 병렬로 프로세싱된다.

[0036] 이 설명의 콘텍스트에서, "데이터 패킷" 은 데이터를 통신하는데 사용되는 임의의 패킷을 지칭할 수도 있으며, "제어 패킷" 은 네트워크 통신을 통한 임의의 양태의 제어를 나타내는 임의의 패킷을 지칭할 수도 있다. 또한, "병렬" 은 임의의 양태의 데이터 패킷 및 제어 패킷을 적어도 부분적으로 동시에 프로세싱하는 프로세싱을 지칭할 수도 있다. 이 때문에, 수신 패킷 프로세싱이 향상된다.

[0037] 또 다른 실시형태에 있어서, 패킷의 프로세싱은 패킷과 연관되는 정확한 소켓 제어 블록 (CB) 을 식별하기 위해, 실질적인 이중 로직 및/또는 다중 데이터 구조 (즉, 탐색표 및 캐시 등) 를 병렬로 사용하는 단계를 포함할 수도 있다. 옵션으로서, 이런 탐색 모드 동안에 (즉, 이런 CB 가 식별되는 경우에), 정확한 소켓 제어 블록이 캐시 내에 존재하지 않는다면, 정확한 소켓 제어 블록은 메인 메모리로부터 취출될 수도 있으며, 취출된 정확한 소켓 제어 블록은 탐색 모드에서 캐시 내에 저장되지 않는다. 이 설명의 콘텍스트에서, CB 는 접속 시도 및/또는 접속을 추적하는데 사용될 수 있는 임의의 정보를 포함할 수도 있다.

[0038] CB 를 최초로 정확히 식별한 데이터 구조의 식별 결과를 사용함으로써, 전술한 식별 프로세스는 가속된다. 또한, 실질적인 이중 로직을 사용함으로써, CB 식별 프로세스는 다수의 패킷에 관해 동시에 수행될 수도 있으며, 따라서, 추가 가속을 제공한다.

[0039] 또 다른 옵션으로서, 태그 정보가 패킷 (즉, 데이터 패킷 및/또는 제어 패킷 등) 에 프리펜딩될 수도 있다. 이런 태그 정보는, 수신기 (RX) 선입선출 (FIFO) 버퍼에서 버퍼링되는 동안에 패킷에, 더 프리펜딩될 수도 있다. 태그 정보를 프리펜딩함으로써 수신 패킷은 전술한 CB 식별이 수행되는 동안에 저장될 수도 있다. 또한, 옵션으로서, 태그 정보에 관해 별도의 버퍼가 필요없을 수도 있다.

[0040] 또 다른 실시형태에 있어서, 추가 패킷이 수신됨에 따라, CB 는 업데이트될 수도 있다. 따라서, 일반적인 사용에서, 제 1 패킷과 연관되는 CB 가 업데이트될 수도 있다. 또한, 적어도 제 1 패킷과 연관되는 CB 의 업데이트를 시작한 후 및 업데이트를 완료하기 전에, 제 2 패킷의 프로세싱이 시작될 수도 있으며, 따라서, 성능이 향상된다. 이 때문에, 패킷 프로세싱은 CB 업데이트와 병렬로 수행될 수도 있으며, 따라서, 다른 프로세스에서는 병목을 생성하는 프로세스 중 하나에서 병목 확률을 감소시킬 수 있다.

[0041] 더 많은 선택적 특징 및 예시적인 구현 세부 설명이 상술한 실시형태에 관해 개시될 것이다. 다음의 세부 설명은 단지 예시적인 목적으로 개시된 것이고, 임의의 방법으로 제한하는 것으로써 해석되어야 한다.

[0042] 도 3 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 아키텍처 (300) 를 나타낸다. 옵션으로서, 아키텍처 (300) 는 도 2 의 예시적인 아키텍처 (200) 의 콘텍스트에서 구현될 수도 있다. 그러나, 물론, 아키텍처 (300) 는 임의의 원하는 콘텍스트에서 구현될 수도 있다.

[0043] 도시된 바와 같이, 복수의 네트워크 (102) 로부터의 데이터 패킷 및 제어 패킷 모두를 수신하는 복수의 미디어 액세스 제어 (MAC) RX 버퍼 (302) 가 제공된다. RX 버퍼 (302) 는 반드시 복수의 네트워크 (102) 를 위한 것은 아니다. 일 실시형태는 단일 네트워크를 위한 하나의 버퍼를 포함할 수도 있다. 이런 MAC RX 버퍼 (302) 는 인터넷 프로토콜 (IP) RX (310) 및 예외 처리기 (exception handler; 314) 모두에 차례로 연결되는 이더넷 RX (306) 에 연결된다. 모든 IP 패킷은 IP RX (310) 로 전송되고, 다른 모든 패킷은 예외 처리기 (314) 로 전송된다.

[0044] IP RX (310) 내에서 패킷 IP 헤더가 분석된다. 그 후, 모든 인터넷 프로토콜 보안 (IPSEC) 프로토콜 패킷은 IPSEC RX 처리기 (308) 로 전송되어, IPSEC RX 버퍼 (304) 에 저장되고, 모든 TCP 또는 UDT 패킷은 TCP RX 전단 모듈 (316) 로 전송된다. 다른 모든 패킷은 예외 처리기 (314) 로 전송된다. 멀티플렉서 (312) 는 IP RX (310) 로부터의 정규 TCP/UDP 패킷과 IPSEC RX 처리기 (308) 로부터의 처리된 IPSEC 패킷 중에서 선택한다.

[0045] 일반적인 사용에서, TCP RX 전단 모듈 (316) 은 인커밍하는 TCP 패킷을 분석하여, 수신 패킷이 TCP 데이터를 포함하는지 또는 TCP 제어 패킷인지를 결정한다. 그 후, 각각의 패킷 타입에 관한 별도의 프로세싱 경로를 제공한다. 이 때문에, 상기 개시한 방법에서 패킷 프로세싱이 향상된다. 또한, 패킷 프로세싱은 패킷과 연관되는 정확한 소켓 CB 를 식별하기 위해 실질적인 이중 로직 및/또는 다중 데이터 구조 (즉, 탐색표 (CB 탐색표 (318) 참조) 및 CB 캐시 (CB 데이터 캐시 (324) 참조) 등) 를 병렬로 사용하는 것을 포함할 수도 있다.

[0046] 그러나, 여전히 상기 개시한 이유 때문에, 데이터 패킷은 MAC RX FIFO 버퍼 (322) 에 저장되며, 동시에 패킷과 연관되는 대응 CB 를 탐색한다. 데이터 패킷이 저장되고 CB 탐색이 완료된 후에, 태그 정보는 MAC RX FIFO 버퍼 (322) 내의 데이터 패킷에 프리펜딩될 수도 있다. TCP RX 전단 모듈 (316) 의 또 다른 양태에서, 제 1

패킷과 연관되는 CB 는 업데이트될 수도 있다. 적어도 제 1 패킷과 연관되는 CB 의 업데이트를 시작한 후 및 업데이트를 완료하기 전에, 제 2 패킷의 프로세싱이 시작될 수도 있으며, 따라서, 성능이 향상된다.

[0047] 이런 TCP RX 전단 모듈 (316) 기능 및 선택적인 구현 세부 설명에 관한 더 많은 정보는 도 4 및 부가 설명을 참조하는 동안에 더 상세히 개시될 것이다.

[0048] 도 3 을 계속 참조하면, TCP RX 전단 모듈 (316) 은 예외 처리기 (314) 에 연결된다. 임의의 로직 예외를 발생시킴으로써 식별되는 모든 패킷은 예외 처리기 (314) 로 전송된다. 예를 들어, 수신 필터 설정과 매칭되거나, 또는 지원되지 않는 옵션을 포함하는 패킷은 예외 처리기 (314) 로 전송될 수도 있다. 상술한 바와 같이, 정규 데이터 패킷은 RX 버퍼 제어기 (320) 를 통해 MAC RX FIFO 버퍼 (322) 에서 저장된다. 이 RX 버퍼 제어기 (320) 는 예외 처리기 (314) 또는 TCP RX 전단 모듈 (316) 로부터의 패킷을 저장하라는 요청을 수락하고, 2 개의 소스 사이에서 조정한다. 이런 상호 작용에 관한 더 많은 정보는 이후 더 상세히 개시될 것이다.

[0049] RX 버퍼 제어기 (320) 는 RX 후단 모듈 (326) 에 차례로 연결되는 추가 MAC RX FIFO 버퍼 (322) 에 연결된다. 따라서, MAC RX FIFO 버퍼 (322) 는 전단 모듈과 후단 모듈 사이에 연결되어 그들 간에 경계를 제공하고 그들을 분리한다.

[0050] 도 3 에 도시된 바와 같이, TCP RX 전단 모듈 (316) 은 패킷이 MAC RX FIFO 버퍼 (322) 에서 서비스하기에 이용가능하다는 것을 RX 후단 모듈 (326) 에게 나타낸다. 곧 명백하게 될 것이지만, RX 후단 모듈 (326) 은 인터넷 소형 컴퓨터 시스템 인터페이스 (iSCSI) 프로토콜 또는 임의의 다른 원하는 프로토콜 (즉, RDMA (원격 데이터 메모리 액세스) 등) 과 같은 애플리케이션 레벨 프로세싱을 처리한다. 이런 RX 후단 모듈 (326) 기능 및 선택적인 구현 세부 설명에 관한 더 많은 정보는 도 5 및 부가 설명을 참조하는 동안에 더 상세히 개시될 것이다.

[0051] 최종적으로, RX 후단 모듈 (326) 은 직접 메모리 액세스 (DMA) 로직 (330) 뿐 아니라 스캐터-게더 리스트 (scatter-gather list: SGL)(344) (및/또는, 가능하게는 메모리 디스크립터 리스트(memory descriptor list: MDL)), 및 익명 버퍼 리스트 (328) 모두를 사용하여, 호스트 메모리 (즉, 예를 들어, 도 2 의 프로세서 시스템 메모리 (206, 106) 참조) 에 수신 패킷을 저장한다. 이 설명의 콘텍스트에서, SGL 은 인커밍하는 데이터가 최종적으로 저장되는 메모리 내의 다양한 영역을 설명하기 위해 제공되는 임의의 데이터 리스트의 대상을 포함할 수도 있다.

[0052] 도 4 는 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 TCP RX 전단 모듈 (316) 을 나타낸다. 옵션으로서, TCP RX 전단 모듈 (316) 은 도 3 의 예시적인 아키텍처 (300) 의 콘텍스트에서 구현될 수도 있다. 그러나, 물론, TCP RX 전단 모듈 (316) 은 임의의 원하는 콘텍스트에서 구현될 수도 있다.

[0053] 도 4 에 도시된 바와 같이, TCP RX 전단 모듈 (316) 은 IP 계층으로부터 (즉, 예를 들어, 도 3 의 IP RX (310) 등을 통해) 데이터를 수신하고, 패킷을 프로세싱하거나 예외로서 패킷을 취급한다. 이를 달성하기 위해, TCP RX 분석기 (414) 및 소켓 위치기 (locator) 모듈 (402) 이 제공된다. 옵션으로서, 다중의 TCP RX 분석기 (414) 및 소켓 위치기 모듈 (402) 이 제공될 수도 있다. 이와 관련하여, 여기에 개시된 임의의 로직 모듈은 프로세싱을 향상하도록 실질적으로 이중 또는 삼중으로 제공할 수도 있다.

[0054] 일반적인 사용에서, TCP RX 분석기 (414) 는 수신 TCP 및 사용자 데이터그램 프로토콜 (UDT) 패킷의 분석을 담당한다. 옵션으로서, 모든 UDT 패킷은 예외로서 (즉, 예를 들어, 도 3 의 예외 처리기 모듈 (314) 등을 통해) 전송될 수도 있거나, TCP 데이터 패킷과 유사한 방법으로 프로세싱될 수도 있다. 추가 옵션으로서, UDT 검사합계 (checksum) 가 유효할 수도 있고, UDT 검사합계가 불량하다면, 패킷은 처리중단될 수도 있다.

[0055] TCP 패킷에 관해, 모든 데이터 패킷은 RX FIFO 버퍼 (322) 에 저장되고, 모든 제어 패킷은 제어 패킷 큐 (404) 로 전송된다. 이 결정은 패킷 길이 뿐 아니라 TCP 헤더 내의 FLAG 비트를 조사함으로써 달성될 수도 있다. 따라서, 제어 패킷은 제 1 프로세싱 경로를 사용하여 프로세싱될 수도 있고, 제 1 프로세싱 경로와는 별도의 제 2 프로세싱 경로를 사용하여 프로세싱될 수도 있다.

[0056] 패킷이 데이터 패킷이라면, 소켓 해시 (hash) 가 소켓 위치기 모듈 (402) 에 의해 연산된다. 배경에 의해, 각각의 데이터 패킷은 IP 주소의 쌍 및 TCP 또는 UDT 포트의 쌍 모두에 연관되어 있다. 해시는 이런 IP 주소 및 포트에 기초하여 (즉, "4 튜플의 소켓(socket 4-tuple)" 을 사용함으로써) 생성될 수도 있다.

[0057] 그 후, 이 해시는 CB 탐색표 (318) 에 인덱싱하는데 사용될 수도 있다. 샘플 CB 탐색표 (318) 는 표 1 에

도시된 것이다.

[0058] [표 1]

[0059] 해시 1 / (메모리에서 소켓 CB 1 에 대한 주소 1)

[0060] 해시 2 / (메모리에서 소켓 CB 2 에 대한 주소 2)

[0061] 해시 3 / (메모리에서 소켓 CB 3 에 대한 주소 3)

[0062] CB 주소는 메모리에서 적절한 CB 의 위치를 식별하는데 사용될 수도 있고, CB 와 연관되는 소켓 처리 식별기로도 사용된다. 통상적으로 CB 는 소켓 상태 등과 같은 다른 정보와 함께 이런 소켓 처리를 포함한다. 생성되는 동일한 해시값을 포함하는 다음 CB 를 지시하는 필드가 CB 구조 내에 있다. 이 방법으로 충돌하는 해시값을 갖는 소켓을 해결할 수 있다.

[0063] 그 후, 현재 패킷과 연관되는 실제 소켓과 소켓 처리의 비교에 기초하여, CB 와 연관되는 소켓이 정확한 소켓인지 여부를 결정한다. 예를 들어, 패킷에서의 4-튜플이 소켓 CB 내의 파라미터와 매칭된다면, 정확한 CB 가 구해진다. 매칭되지 않으면, 다음에 링크되는 소켓 처리는 CB 로부터 판독되고, 그 후 그 소켓은 패치된다.

[0064] 다음에 링크되는 소켓 처리를 취출하는 이 프로세스는 정확한 소켓 CB 가 발견되거나, 수신 패킷과 연관될 수 있는 CB 가 존재하지 않다고 결정될 때까지 계속한다. 옵션으로서, CB 탐색표 (318) 사이즈는 해시 충돌 횟수를 감소시키기 위해 지원되는 소켓의 최대 수의 2 배일 수도 있고, 외부 메모리에 위치할 수도 있다.

[0065] CB 탐색표 (318) 의 이런 사용과 병렬로, CB 데이터 캐시 (324) 에서 탐색이 수행된다. CB 데이터 캐시 (324) 는 가장 최근에 사용된 "n" 소켓 CB (즉, 32 정도) 를 포함한다. CB 데이터 캐시 (324) 는 CB 데이터 캐시 (324) 에 존재하는 각각의 CB 엔트리에 관해 생성된 해시를 표시하는 해시 연관표를 더 포함한다. 그 후, 소켓 위치기 모듈 (402) 은 매칭되는 CB 해시가 CB 데이터 캐시 (324) 에 존재하는지 여부를 결정하도록 CB 데이터 캐시 (324) 에 문의할 수 있다. 이 때문에, 가능한 CB 매칭이 CB 데이터 캐시 (324) 내에 있는지 여부를 소정의 시간 (즉, 해시를 생성하는 클럭) 내에 결정할 수 있다.

[0066] 최초로 정확히 CB 를 식별하는 데이터 구조 (즉, CB 탐색표 (318) 또는 CB 데이터 캐시 (324)) 의 식별 결과를 사용함으로써, 전술한 식별 프로세스는 가속된다. 예를 들어, CB 가 CB 데이터 캐시 (324) 에 존재하지 않는 경우에, 정확한 소켓을 발견하는데 요구되는 메모리 판독의 최대 수는 수학식 1 에 의해 주어질 수도 있다.

수학식 1

[0067] 클럭수 = 1 + p(n),

[0068] · 여기서, 제 1 판독은 CB 탐색표 (318) 탐색에 의한 것이고,

[0069] · n 은 특정 해시에 관한 CB 해시 충돌 횟수이며,

[0070] · p 는 CB 엔트리로부터 소켓 포트 및 IP 주소를 판독하는데 요구되는 클럭 사이클 수이다.

[0071] CB 가 CB 데이터 캐시 (324) 에 존재한다면, CB 를 발견하는데 요구되는 클럭의 최대 수는 수학식 2 에 의해 주어진다.

수학식 2

[0072] 클럭의 수 = m,

[0073] · 여기서, m 은 특정 해시를 갖는 CB 데이터 캐시 (324) 내의 CB 의 수를 지칭한다.

[0074] 캐시 버스 폭은 이상적으로 사이징되기 때문에 (즉, 128 비트 등), 파라미터가 CB 데이터 캐시 (324) 로부터 신속하게 판독될 수 있다.

[0075] 추가 특징으로서, 소켓 위치기 모듈 (402) 이 CB 탐색표 (318) 에 의해 참조되는 CB 를 탐색하는 경우에, CB 데이터 캐시 (324) 는 모드를 통한 특별 판독을 허용할 수도 있다. 이 특별 판독 모드에서, 요청된 CB 는 CB 데이터 캐시 (324) 에 위치하는지 여부를 확인하기 위해 최초로 검사된다. CB 데이터 캐시에 존재한다면, 콘텐츠는 즉시 리턴될 수 있다. 그러나, CB 탐색표 (318) 에 위치하지 않는다면, 메인 CB 메모리로부터 판독되나, 이 모드에서 CB 는 CB 탐색표 (318) 로 전송되지 않는다. 이것은 이 포인트에서의 탐색 로직이 여전히 수신 패킷과 연관되는 CB 를 여전히 찾고 있기 때문이다. 정확한 CB 가 일단 위치하면, CB 와 연관되

는 처리는 CB 데이터 캐시 (324) 를 통해 CB 를 판독하는 TCP RX 상태 제어기 (412) 로 전달된다. 동시에, CB 는 메인 CB 메모리로부터 취출되어, CB 데이터 캐시 (324) 에 위치한다.

[0076] 데이터 패킷은 CB 탐색이 수행되는 것과 동시에 RX FIFO 버퍼 (322) 에 저장된다. 이 방법으로 소정의 사이즈 (즉, 80 바이트 등) 보다 더 큰 데이터 패킷에 관해, 적절한 CB 를 찾는데, 평균적으로 최소의 시간이 걸린다. 정확한 CB 를 위치시키는 것은 시간에 민감한 작업이기 때문에, 이런 로직 (즉, TCP RX 분석기 (414) 및/또는 소켓 위치기 모듈 (402) 등) 은 IP 계층으로부터 수신되는 다수의 패킷이 동시에 프로세싱될 수도 있도록, TCP RX 전단 모듈 (316) 에서 실질적으로 이중일 수도 있다 (또는 삼중으로 제공됨).

[0077] 이는 제 1 패킷이 여전히 프로세싱되는 동안에, 로직이 다음 패킷에 관한 CB 를 미리 보게 하거나 탐색을 시작하게 한다. 일단, CB 가 발견되고 콘텐츠가 패치되면, 패킷 프로세싱 (즉, 패킷에 관해 무엇을 할지를 결정) 은 수 클럭 사이클 내에서 행해진다.

[0078] 수신 프로세싱으로 돌아가서, 정확한 CB 가 CB 데이터 캐시 (324) 내에 존재하지 않으면, 메인 CB 메모리로부터 판독되는 동시에, CB 데이터 캐시 (324) 에 위치한다. CB 가 CB 데이터 캐시 (324) 에 이미 존재한다면, 직접 판독될 수 있다. 소켓 상태, 제어 비트, 및 (SEQ 및 ACK 수를 포함하는) 파라미터는 CB 를 사용하여 모두 획득될 수도 있다.

[0079] 일단, CB 파라미터 모두가 패치되었으면, 패킷의 프로세싱은 소켓 상태에 의존하여 다른 소정의 시간 (즉, 대략 2 클럭 사이클) 내에 완료된다. 따라서, 소정의 사이즈 (즉, 평균적으로 300 바이트) 보다 더 큰 데이터 패킷에 관한 프로세싱은 데이터가 RX FIFO 버퍼 (322) 에 저장되는 시간에 의해 완료될 수도 있다. 이것은 CB 가 메인 CB 메모리로부터 패치될 필요가 있다고 할지라도 적용될 수도 있다. CB 가 이미 CB 데이터 캐시 (324) 내에 존재한다면, 훨씬 더 작은 데이터 패킷은 패킷 프로세싱에 요구되는 최소 연장 시간으로 프로세싱될 수 있다.

[0080] 태그 정보는 RX FIFO 버퍼 (322) 내의 각각의 패킷에 프리펜딩된다. 이 태그 정보는 패킷 타입 (즉, 예외, TCP 데이터 패킷 등) 정보, 이와 관련되는 소켓 처리, 및/또는 다른 제어 정보 및 상태 정보를 포함할 수도 있다. 태그 정보는 전체 패킷이 수신되어 유효하다고 검증된 후에 채워질 수도 있다.

[0081] 이런 태그 정보는 RX FIFO 버퍼 (322) 에서 버퍼링되는 동안에, 패킷에 더 프리펜딩될 수도 있다. 이런 방법으로 태그 정보를 프리펜딩함으로써, 전술한 CB 식별이 수행되는 동안에 수신 패킷이 저장될 수도 있다. 또한, 옵션으로서, 태그 정보에 관한 별도의 버퍼가 필요없을 수도 있다.

[0082] TCP 분석기 (414) 의 또 다른 기능은 TCP 검사 합계를 유효하게 하는 것이다. 이것은 (데이터 패킷에 관한) RX FIFO 버퍼 (322) 또는 (TCP 제어 패킷에 관한) 제어 패킷 큐 (404) 에서 분석되어 저장되기 때문에 패킷을 스누핑함으로써 행해진다. 검사 합계 결과는 IP 계층에 의해 제공되는 의사 헤더 검사 합계와 결합되어, 최종 검사 합계를 생성한다. 이 검사는 IP 모듈로부터의 마지막 워드가 판독된 후에, 소정의 시간프레임 (즉, 3 클럭) 에 유효하다. 임의의 더 낮은 계층으로부터의 임의의 다른 에러에 의해 또는 TCP 검사합계 에러로부터 패킷이 불량하다고 간주되는 경우에, 패킷은 RX FIFO 버퍼 (322) 또는 제어 패킷 큐 (404) 로부터 드롭된다.

[0083] 도 4 를 계속 참조하면, TCP RX 상태 제어기 (412) 가 제공된다. TCP RX 상태 제어기 (412) 는 수신 TCP 데이터 패킷에 관한 액션 코스의 결정을 담당한다. 소켓 CB 내의 파라미터에 대해 패킷 파라미터를 검사한다. 이것은 패킷의 시퀀스 수, 소켓 상태 등을 검사하는 것을 포함한다.

[0084] 그러나, 패킷은 마치 양호할 것처럼 프로세싱된다. 그런 방법으로 전체 패킷이 수신된 후에는 모든 프로세싱이 이미 발생했고, 최소의 추가 연산이 필요하다. 패킷이 불량하다고 판명되면, CB 는 업데이트되지 않고, 패킷은 RX FIFO 버퍼 (322) 또는 제어 패킷 큐 (404) 로부터 드롭된다. 이 경우에는, 패킷이 도달하기 전임 포인트로, 각각의 버퍼에 관한 기록 포인터가 리셋된다.

[0085] 일단, 전체 데이터 패킷이 RX FIFO 버퍼 (322) 에 저장된다면, TCP RX 상태 제어기 (412) 는 TCP 송신기 모듈 (미도시) 을 통해 ACK 를 스케쥴링할 수도 있다. 4 개의 ACK 모드, 즉, 정규 즉시 ACK, 정규 지연 ACK, 호스트 모드 즉시 ACK, 호스트 모드 지연 ACK 가 지원될 수도 있다.

[0086] 정규 모드에서는 데이터 패킷이 수신되어 유효하다고 결정된 후에 ACK 또는 지연 ACK 를 즉시 요청하거나 큐잉하게 (queue) 한다. 2 개의 호스트 모드에서는 호스트가 RX DMA 를 통한 데이터 수신을 확인응답한 후에 ACK 를 단지 요청하거나 대기하게 한다.

[0087] 다시 TCP RX 분석기 (414) 로 돌아가서, 로직 블록은 순수 TCP 제어 패킷(즉, 임의의 데이터를 포함하지 않는 패킷) 을 데이터 패킷과 구별할 수도 있다. 이는 순수 제어 패킷이 통상적으로 더 짧은 패킷이며, 프로세싱에서 시간에 민감하지 않기 때문이다. 이들 TCP 제어 패킷은 TCP 헤더에 FLAG 비트 설정에 의해 탐지되어, 패킷의 전체 길이와 결합된다. 데이터가 패킷에 포함되지 않고 푸시 (PSH) 비트가 설정되어 있지 않다면, 그 패킷은 순수 제어 패킷이라고 간주된다.

[0088] 이들 패킷은 도 4 의 제어 패킷 큐 (404) 로 전송된다. 각각의 패킷에 관한 검사합계는 패킷이 전환될 때마다 연산되고, 불량 패킷은 폐기한다. 또한, 소켓 해시가 연산되어, 별도의 섹션에서 패킷에 프리펜딩된다. 그 후, 제어 패킷 처리기 (416) 는 큐로부터 패킷을 판독하고, 그 패킷을 프로세싱한다. 그 후, 표 2 의 다음 동작이 제어 패킷에 관해 수행된다.

[0089] [표 2]

[0090] 1. 소켓 해시는 CB 탐색표 (318) 에서 탐색된다. 동시에, 해시를 검사하여, CB 가 CB 데이터 캐시 (324) 내에 이미 존재하는지 여부를 확인한다.

[0091] 2. CB 가 이미 CB 데이터 캐시 (324) 내에 이미 존재한다고 가정하면, 응용가능 필드를 판독한다.

[0092] 3. CB 가 이미 CB 데이터 캐시 (324) 내에 존재하지 않는다면, CB 를 메인 CB 메모리로부터 판독하고, 데이터 캐시 (324) 에 위치시킨다.

[0093] 4. 그 후, 액션은 수신 패킷 타입 및 소켓의 현재 상태에 의해 결정된다. 이들 액션은 다음의,

[0094] a. TCP 송신기로부터의 응답 요청

[0095] b. 호스트로 상태 메시지 전송

[0096] c. 패킷 폐기

[0097] d. 예외로서 패킷 전송

[0098] 으로 제한되는 것은 아니지만, 이들 중 임의 것일 수 있다.

[0099] 5. 요구되는 액션을 결정한 후에, 이에 따라 CB 가 업데이트된다.

[0100] 소켓의 탐색 및 판독, 패킷 프로세싱, 및 CB 의 업데이트는 모두 파이프 라이닝된 동작이며, 이전의 제어 패킷이 여전히 프로세싱되는 동안에, 처리기가 다음 소켓을 발견하기 시작하게 한다.

[0101] 도 5 는 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 RX 후단 모듈 (326) 을 나타낸다. 옵션으로서, RX 후단 모듈 (326) 은 도 3 의 예시적인 아키텍처 (300) 의 콘텍스트에서 구현될 수도 있다. 그러나, 물론, RX 후단 모듈 (326) 은 임의의 원하는 콘텍스트에서 구현될 수도 있다.

[0102] 일단 패킷이 RX FIFO 버퍼 (322) 에 완전히 저장되어 패킷 버퍼 헤더가 가득 차면, RX 후단 모듈 (326) 은 그 패킷을 프로세싱하기 시작한다. RX 후단 모듈 (326) 은 데이터 할당 및 RX 버퍼 헤더 분석 모듈 (502) 을 사용하여 패킷 버퍼 헤더를 분석하고 제거함으로써 시작된다. 상술한 바와 같이, 이는 RX 후단 모듈 (326) 에 패킷 타입 (즉, 예외 또는 TCP 데이터 패킷), 패킷과 연관되는 CB 처리, 및 다른 상태 및 제어 정보를 알린다. 일 실시형태에 있어서, 헤더의 길이는 256 비트일 수도 있다.

[0103] 패킷 버퍼 헤더가 분석되고 제거된 후에, 패킷은 재정렬된다. 이 재정렬은 정규의 TCP (및 선택적으로 UDP) 데이터 패킷 때문에 필요할 수도 있으며, 따라서, 패킷 이더넷, IP, 및 TCP 헤더 또한 제거된다. 이들 헤더를 제거하는 것은 결과 데이터로 하여금 정렬된 비-FIFO 워드가 되게 할 수도 있고, 재정렬은 다음 로직 모듈이 패킷 데이터에 대해 동작하는 것을 더 간단하게 한다.

[0104] 재정렬 후, 패킷은 애플리케이션 특정 프로세싱 로직 (504) 을 통해 선택적으로 통과될 수도 있다. 이 로직은 iSCSI 프로토콜 또는 원격 직접 메모리 액세스 (RDMA) 기능을 구현하는 로직을 포함할 수도 있으나, 여기에 제한되지는 않는다. iSCSI 지원에 관해, 이 모듈은 iSCSI 순환 중복 검사 (CRC) 검증, iSCSI 프로토콜 데이터 유닛 (PDU) 헤더 분석, 및 고정 간격 마커 (FIM) 제거를 수행할 수도 있다.

[0105] (네트워크 스택에서 모든 계층으로부터의) 모든 예외 패킷은 예외 버퍼 리스트 (즉, 일시적인 버퍼, 홀딩 버퍼, 에디 버퍼 등) 에서 특정된 바와 같이 호스트 메모리로 전송된다. 예외 리스트 (즉, 예를 들어, 도 3 의 리

스트 (328) 참조) 는 호스트 드라이버에 의해 제공된다. 리스트는 버퍼를 사용함에 따라 호스트에 의해 계속적으로 향상될 수도 있다. 예외 버퍼 주소의 취출 및 예외 버퍼 리스트의 관리는 SGL 프로세싱 로직 (506) 에 의해 관리된다.

[0106] 일반적으로, 소켓에 도달하는 통상의 TCP 데이터는, 호스트 메모리에서 데이터가 저장되어야 하는 영역을 특정하기 위해 소켓 특정 SGL 을 사용한다. SGL (즉, 예를 들어, 도 3 의 SGL (334) 등 참조) 은 호스트 드라이버에 의해 제공된다. 리스트는 사용됨에 따라 호스트에 의해 계속적으로 향상될 수도 있다. 이 설명의 콘텍스트에서, SGL 은 인커밍하는 데이터가 최종적으로 저장되는 메모리 내의 다양한 영역을 설명하도록 제공되는 임의의 데이터 리스트 객체를 포함할 수도 있다. 데이터가 소켓에 수신되지만, 소켓에 연관되는 SGL 가 임의의 유효한 버퍼 주소를 포함하지 않는 경우에, 수신 데이터는 예외 버퍼 리스트를 사용하여 호스트 메모리로 전송된다. 또한, SGL 관리는 SGL 프로세싱 로직 (506) 에 의해 처리된다.

[0107] 정규 TCP 데이터 패킷에 관해, SGL 엔트리는 SGL 메모리 (334) 로부터 최초로 취출된다. 수신 데이터의 시퀀스 수는 데이터가 SGL 에서 위치해야 하는 영역을 표시한다. 이것은 심지어 아웃-오브-시퀀스 (OOS) 데이터의 정확한 정렬을 적절히 허용한다. 순서대로 수신된 데이터에 관해, SGL 로부터의 단일 판독은 데이터를 저장하는 장소의 호스트 주소를 획득하도록 요구된다.

[0108] 그 후, 데이터를 프로세서 시스템 메모리로 DMA 하라고 요청할 수 있다. DMA 요청이 주어지는 경우에, 데이터는 RX FIFO 버퍼 (322) 로부터 판독되어 호스트 DMA 인터페이스 (330) 로 전송되며 (도 3 참조), 여기서, 데이터는 특정 주소에서의 프로세서 시스템 메모리 (206) 로 DMA 된다. 다수의 SGL 은 핑퐁 동작 모드가 가능하도록 소켓마다 지원된다. 이것은 임의의 데이터가 익명 버퍼로 전송되게 하지 않고, 호스트 드라이버가 하나의 SGL 이 만료되자마자 다음 SGL 을 제공하게 한다.

[0109] 단일 데이터 패킷이 하나 이상의 SGL 엔트리를 점유하는 경우에, 패킷의 제 1 부분이 DMA 되는 동안, 다음 SGL 엔트리는 패치되어 프로세싱될 수도 있다. 이런 방법으로, 각각의 요청의 완료 후에 즉시 다음 DMA 를 요청한다.

[0110] 일단 DMA 가 완료되면, 데이터가 도달되었음을 호스트 드라이버에게 알리기 위해, 상태 메시지가 옵션으로 생성될 수도 있다. 동시에, 임의의 CB 파라미터는, TCP RX CB 업데이팅 및 상태 메시지 요청 모듈 (508) 을 사용하여 업데이트된다.

[0111] 패킷 헤더의 분석, SGL 의 패치, 데이터의 DMA, 및 CB의 업데이트는 모두 파이프라이닝된 동작일 수도 있다. 이는 심지어 이전 패킷에 관한 DMA 가 여전히 완료되어 있는 동안에도 RX 후단 모듈 (326) 이 다음 패킷 헤더를 프로세싱하기 시작하게 하며, 이는 데이터의 최대 스루풋을 보장한다. 또한, 여기에서 개시된 임의의 로직 모듈은 실질적으로 이중 또는 심지어 삼중으로 제공되어, 프로세싱을 향상시킬 수도 있다.

[0112] 도 6 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 전단 방법 (600) 을 나타낸다. 옵션으로서, 전단 방법 (600) 은 도 2 의 예시적인 아키텍처 (200) 또는 심지어 도 3 내지 도 5 의 예시적인 구조의 콘텍스트에서 수행될 수도 있다. 그러나, 물론, 전단 방법 (600) 은 임의의 원하는 콘텍스트에서 구현될 수도 있다. 또한, 다양한 기능은 예시적인 (즉, 상기 개시한 것과 같은) 컴포넌트에 기인될 수도 있지만, 다양한 기능이 임의의 원하는 엔티티에 의해 수행될 수도 있다는 것을 이해하는 것이 중요하다.

[0113] 도 6 은 RX 버퍼 (즉, 예를 들어, 도 3 의 RX FIFO 버퍼 (322) 참조) 까지의 수신 패킷에 관한 프로세싱 흐름을 도시한 것이고, 따라서 전단 프로세싱에 집중되어 있다.

[0114] 동작단계 602 에서, 인터넷 프로토콜 (IP) 계층은 수신 패킷이 이용가능한지 여부를 표시한다. 이런 표시에 응답하여, 판정단계 604 에서, RX 분석기 (즉, 예를 들어, 도 4 의 RX 분석기 (414) 참조) 가 이용가능한지 여부를 결정한다.

[0115] RX 분석기가 이용가능하지 않다고 결정된다면, 전단 방법 (600) 은 이용가능한 RX 분석기를 기다린다. 동작단계 606 을 주목한다. RX 분석기가 이용가능하다면, 동작단계 608 에서 수신 패킷을 이용가능한 RX 분석기로 전송한다.

[0116] 일단, RX 분석기가 이용가능하다면, 패킷 헤더 내에 포함되는 파라미터를 기초로 하여, 해시가 패킷에 관해 생성된다 (624). 그 후, RX 분석기는 TCP 헤더를 관찰하여, 패킷 타입을 결정하고 (판정단계 610 참조), 패킷 파라미터를 분석한다. 패킷이 순수 TCP 제어 패킷이라면 (즉, 패킷이 TCP 데이터를 포함하지 않는다면), 패킷은 제어 패킷 큐 (즉, 예를 들면, 도 4 의 제어 패킷 큐 (404) 참조) 로 전송된다 . 동작단계 612 를 참

조한다.

[0117] 패킷이 TCP 데이터를 포함하지 않는다면, CB 탐색이 동작단계 616 및 618 에서 시작된다. 이중 데이터 구조 (즉, 예를 들어, CB 탐색표 (318) 및 CB 데이터 캐시 (324) 등 참조) 를 통해 병렬로 탐색할 수도 있다. 어느 경로든 먼저 완료하면 탐색 프로세싱은 종료된다. 그러나, CB 데이터 캐시 탐색이 매칭되는 CB 엔트리를 발견하는데 실패한다면, 동작단계 620 에 표시된 바와 같이 로직은 CB 탐색표 탐색을 완료할 때까지 기다린다. 일 실시형태에 있어서, CB 탐색표가 CB 를 발견하는데 실패했지만, 그 CB 가 CB 데이터 캐시 내에서 발견된 경우는 제외된다.

[0118] CB 를 발견하는 것과 병렬로, TCP 데이터는 RX 버퍼 (즉, 예를 들어, RX FIFO 버퍼 (322) 등 참조) 에 저장된다. 동작단계 614 에 주목한다. 일단 모든 데이터가 기록되면, 동작단계 622 에서 태그 섹션은 RX FIFO 내의 데이터 섹션에 프리펜딩된다. 이 태그는 (하나가 발견되었다면) CB 엔트리로부터 획득된 일부 상태 정보 뿐 아니라 패킷에 관한 파라미터 (즉, 무슨 타입의 데이터인지) 를 포함한다. 태그가 RX FIFO 에 기록된 후에, 특정 분석기는 또 다른 수신 패킷을 자유롭게 수락한다.

[0119] 도 7 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 후단 방법 (700) 을 나타낸다. 옵션으로서, 후단 방법 (700) 은 도 2 의 예시적인 아키텍처 (200) 또는 심지어 도 3 내지 도 5 의 예시적인 구조의 콘텍스트에서 수행될 수도 있다. 그러나, 여전히 후단 방법 (700) 은 도 6 의 전단 방법 (600) 과 결합하여 수행될 수도 있다.

[0120] 그러나, 물론, 후단 방법 (700) 은 임의의 원하는 콘텍스트에서 구현될 수도 있다. 또한, 다양한 기능은 예시적인 (즉, 상기 개시한 것과 같은) 컴포넌트에 기인될 수도 있지만, 다양한 기능이 임의의 원하는 엔티티에 의해 수행될 수도 있다는 것을 이해하는 것이 중요하다.

[0121] 도 7 은 RX 버퍼 (즉, 예를 들어, 도 3 의 RX FIFO 버퍼 (322) 참조) 로부터의 수신 패킷에 관한 프로세싱 흐름을 도시한 것이고, 따라서 후단 프로세싱에 집중되어 있다.

[0122] 도 7 에서의 흐름은 데이터가 RX FIFO 버퍼 (즉, 예를 들면, RX FIFO 버퍼 (322) 등 참조) 의 출력에서 이용가능한 경우에 시작된다. 패킷이 이용가능하다면, 먼저, 동작단계 702 및 704 에서 RX FIFO 버퍼 엔트리에 관한 헤더 섹션을 판독한다. 이것은 다른 상태 정보 뿐 아니라 데이터 패킷 타입을 후단 로직에 표시한다.

[0123] 패킷 버퍼 헤더는 분석된 후에, 이더넷, IP, 및 (오프로딩된 연결에 대한 수신 패킷에 관한) TCP/UDP 헤더와 함께 제거되고, 데이터가 재정렬된다. 동작단계 706 를 참조한다.

[0124] 그 후, 데이터는 선택적 할당 특정 프로세싱 로직으로 선택적으로 통과된다. 동작단계 708 를 참조한다. 예를 들어, 이 로직에서는 iSCSI 및 RDMA 지원 프로세싱이 수행된다.

[0125] 패킷이 오프로딩된 연결에 속한다면 (즉, 패킷 파라미터와 매칭되는 CB 엔트리가 발견되었다면), 임의의 SGL 버퍼가 데이터에 관해 이용가능한지 여부를 확인하도록 검사한다. 판정단계 710 을 주목한다. 이용가능한 버퍼가 있다면, 데이터는 프로세서 시스템 메모리 내의 소켓 버퍼로 DMA 된다. 이것은, 데이터가 프로세서 시스템 메모리로 DMA 된 후에, 동작단계 712 에서 소켓의 SGL 로부터 호스트 버퍼 주소를 획득함으로써 달성된다. 동작단계 716 을 참조한다.

[0126] 소켓 버퍼가 이용가능하지 않다면, 데이터는 전역 (global) 리스트로부터의 예외 버퍼 주소를 사용하여 (또한, 프로세서 시스템 메모리에 위치한) 일반적인 예외 버퍼로 DMA 된다. 동작단계 714 를 참조한다. 또한, 그 후, 프로세싱되는 데이터가 존재한다는 것을 표시하는 통지가 호스트로 전송될 수도 있다.

[0127] (판정단계 718 마다) CB 가 패킷을 위해 사용되었다면, 그 CB 는 동작단계 722 에서 업데이트된다. 사용되지 않았다면, (동작단계 720 에 기재된 바와 같이) 추가 동작은 요구되지 않는다. 병렬로, 후단 로직은 RX FIFO 버퍼로부터의 다음 패킷을 프로세싱하기 시작할 수도 있다.

[0128] 도 8 은 일 실시형태에 따른, 수신 패킷을 프로세싱하는 예시적인 제어 패킷 프로세싱 방법 (800) 을 나타낸다. 옵션으로서, 프로세싱 방법 (800) 은 도 2 의 예시적인 아키텍처 (200) 또는 심지어 도 3 내지 도 5 의 예시적인 구조의 콘텍스트에서 수행될 수도 있다. 그러나, 여전히 프로세싱 방법 (800) 은 도 6 및 도 7 의 전단 방법 (600) 및 후단 방법 (700) 과 각각 결합하여 수행될 수도 있다.

[0129] 그러나, 물론, 프로세싱 방법 (800) 은 임의의 원하는 콘텍스트에서 구현될 수도 있다. 또한, 다양한 기능은 예시적인 (즉, 상기 개시한 것과 같은) 컴포넌트에 기인될 수도 있지만, 다양한 기능이 임의의 원하는 엔티

티에 의해 수행될 수도 있다는 것을 이해하는 것이 중요하다.

[0130] 도 8 의 프로세싱 방법 (800) 은 제어 패킷이 제어 패킷 큐 (즉, 예를 들어, 도 4 의 제어 패킷 큐 (404) 참조) 의 출력에서 이용가능한 경우에 시작된다. 동작단계 802 에 주목한다. 최초로 하는 일은 패킷 데이터 버퍼를 분석하는 것이다. 동작단계 804 를 주목한다. 이 헤더는, 제어 패킷과 연관되는 다른 상태 정보와 함께 생성된 패킷 해시를 포함한다. 다음으로, 매칭되는 CB 의 탐색이 취출된 해시값을 사용하여 시작된다. 데이터 패킷에 관한 전단 로직에 사용되는 도 6 의 전단 방법 (600) 과 유사하게, 이중 데이터 구조 (즉, 예를 들어, CB 탐색표 (318) 및 CB 데이터 캐시 (324) 등 참조) 에 병렬로 문의한다. 동작단계 806 및 808 을 참조한 다음에, 동작단계 810 에서 결과를 기다린다.

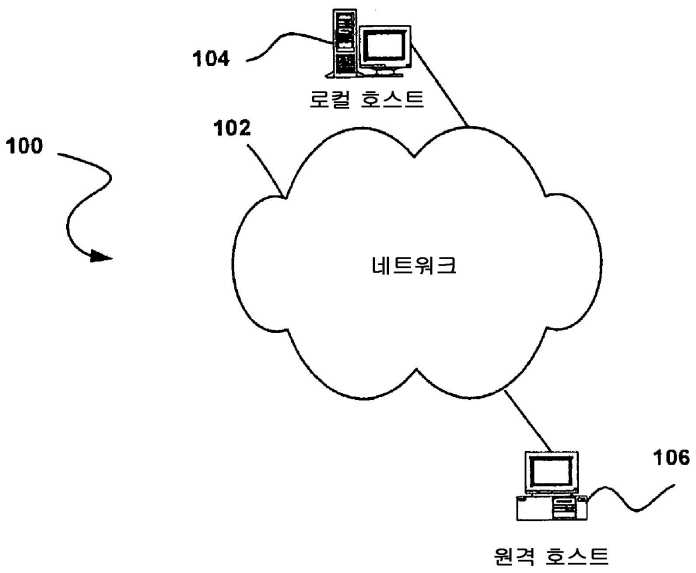
[0131] 판정단계 812 마다 수신 TCP 제어 패킷에 매칭되는 CB 가 발견되지 않았다면, 동작단계 816 에서, 패킷은 DMA 를 통해 호스트로 전송되도록 스케줄링되고, RX 제어 로직은 큐로부터의 다음 제어 패킷을 프로세싱하기 시작할 수 있다. 로직은, 계속되기 전에, 반드시 제어 패킷이 동작단계 824 에서의 호스트로 DMA 되도록 기다릴 필요는 없다.

[0132] 판정단계 812 마다 제어 패킷에 관해 매칭되는 CB 가 발견되었다면, 그 패킷은 동작단계 814 에서 프로세싱된다. 그 후, 소켓 CB 가 패킷 프로세싱의 결과로서 업데이트가 필요한지 여부를 확인하도록 검사한다. 판정단계 818 을 참조한다. CB 가 업데이팅이 필요하다면, 동작단계 820 및 822 에서 업데이트되도록 스케줄링된다. 그 후, 제어 패킷 로직은 큐로부터의 다음 제어 패킷을 프로세싱하기 시작할 수 있다. 동시에, 상술한 바와 같이, CB 는 현재의 제어 패킷에 관해 업데이트된다.

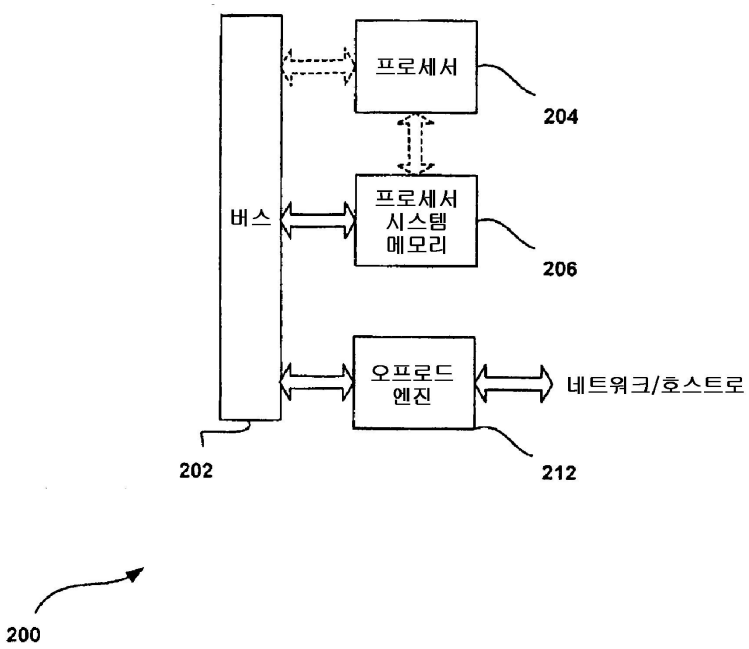
[0133] 다양한 실시형태가 상술되었지만, 이들은 제한이 아닌 단지 예로서 제공된 것임을 이해해야 한다. 따라서, 바람직한 실시형태의 폭과 범위는 상술한 임의의 예시적인 실시형태로 제한되어서는 안되고, 오직 다음의 특허청구 범위 및 그 균등물에 따라 정의되어야 한다.

도면

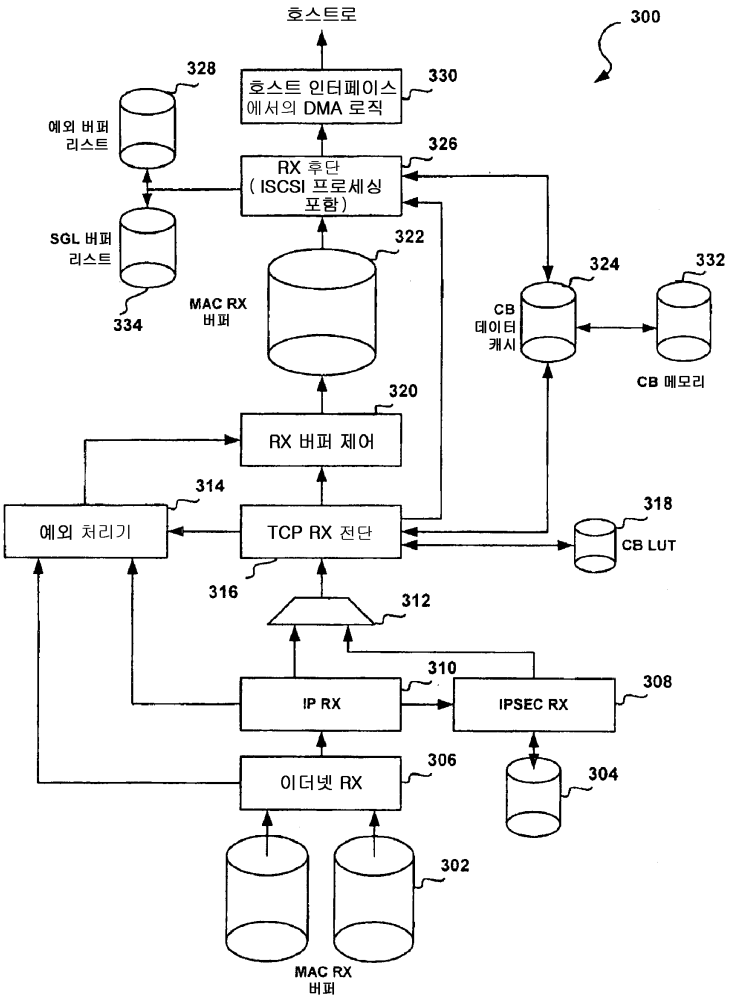
도면1



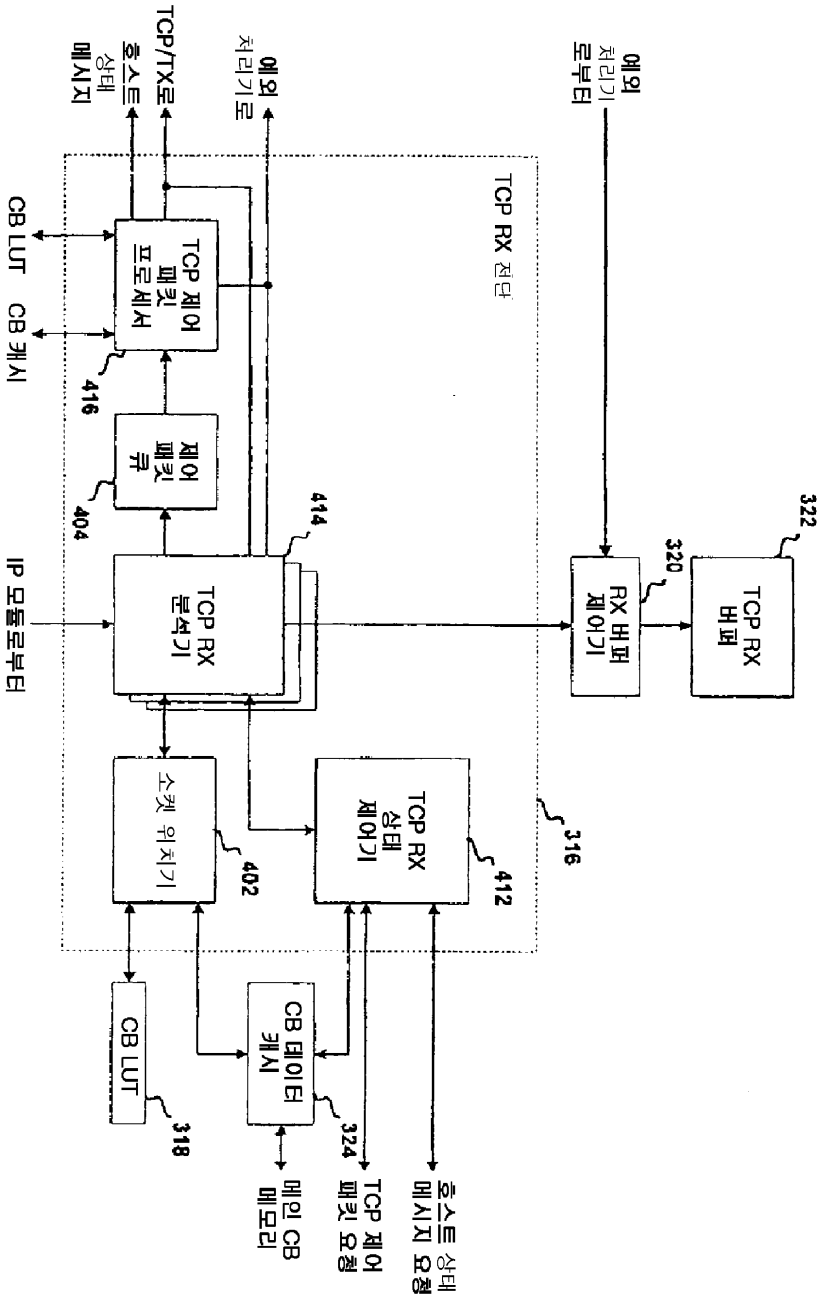
도면2



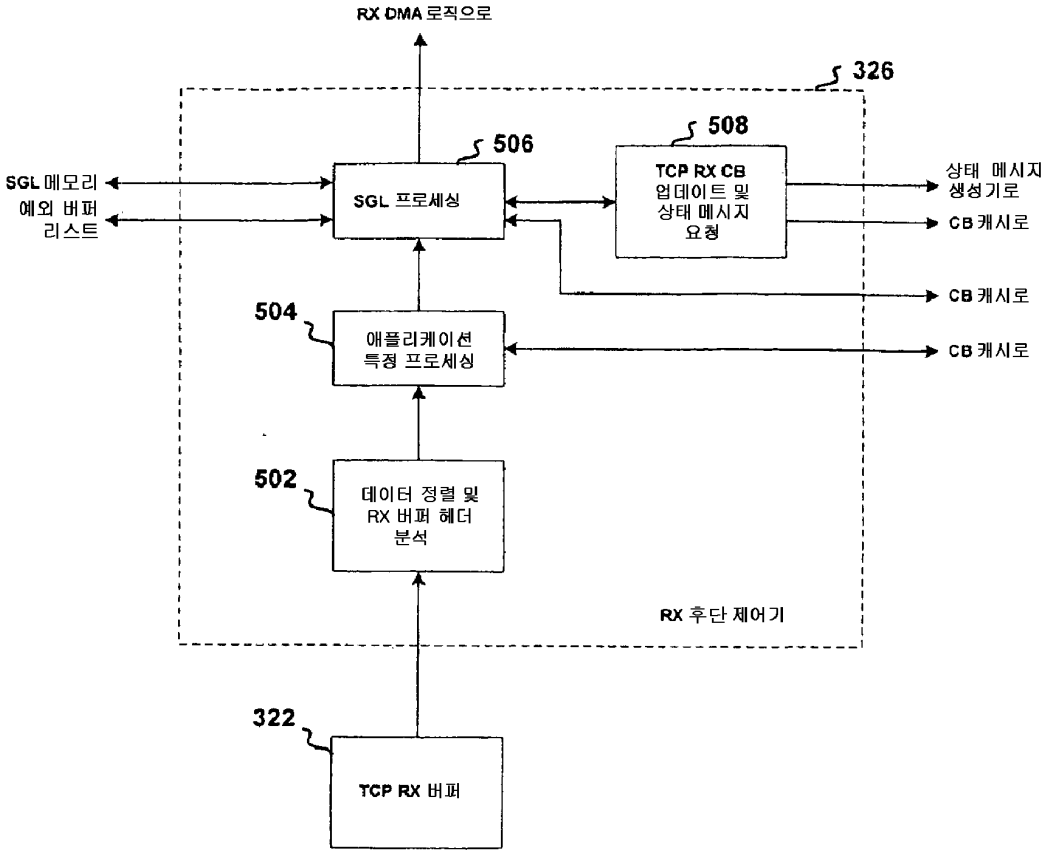
도면3


호스트로
300
328
예외 버퍼 리스트
호스트 인터페이스에서의 DMA 로직
330
326
RX 후단 (ISCSI 프로세싱 포함)
SGL 버퍼 리스트
334
322
MAC RX 버퍼
324
CB 데이터 캐시
332
CB 메모리
320
RX 버퍼 제어
314
예외 처리기
TCP RX 전단
318
CB LUT
316
312
310
IP RX
IPSEC RX
308
306
이더넷 RX
304
302
MAC RX 버퍼

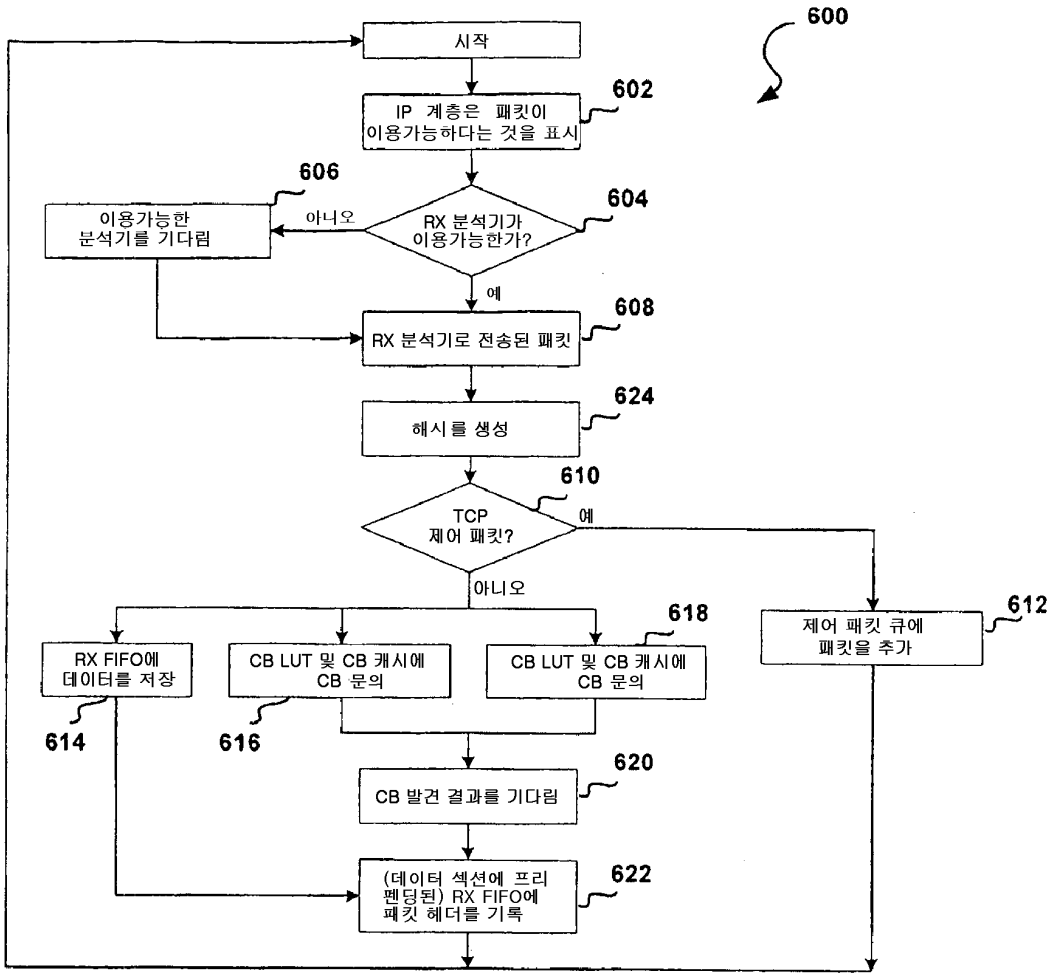
도면4


TCP RX 전단
316
예외 처리기로부터
322
TCP RX 버퍼
320
RX 버퍼 제어기
414
TCP RX 분석기
IP 모듈로부터
404
제어 패킷 큐
416
TCP 제어 패킷 프로세서
CB LUT
CB 캐시
TCP/TX로 호스트 상태 메시지
예외 처리기로
402
소켓 위치기
412
TCP RX 상태 제어기
318
CB LUT
324
CB 데이터 캐시
메인 CB 메모리
TCP 제어 패킷 요청
호스트 상태 메시지 요청

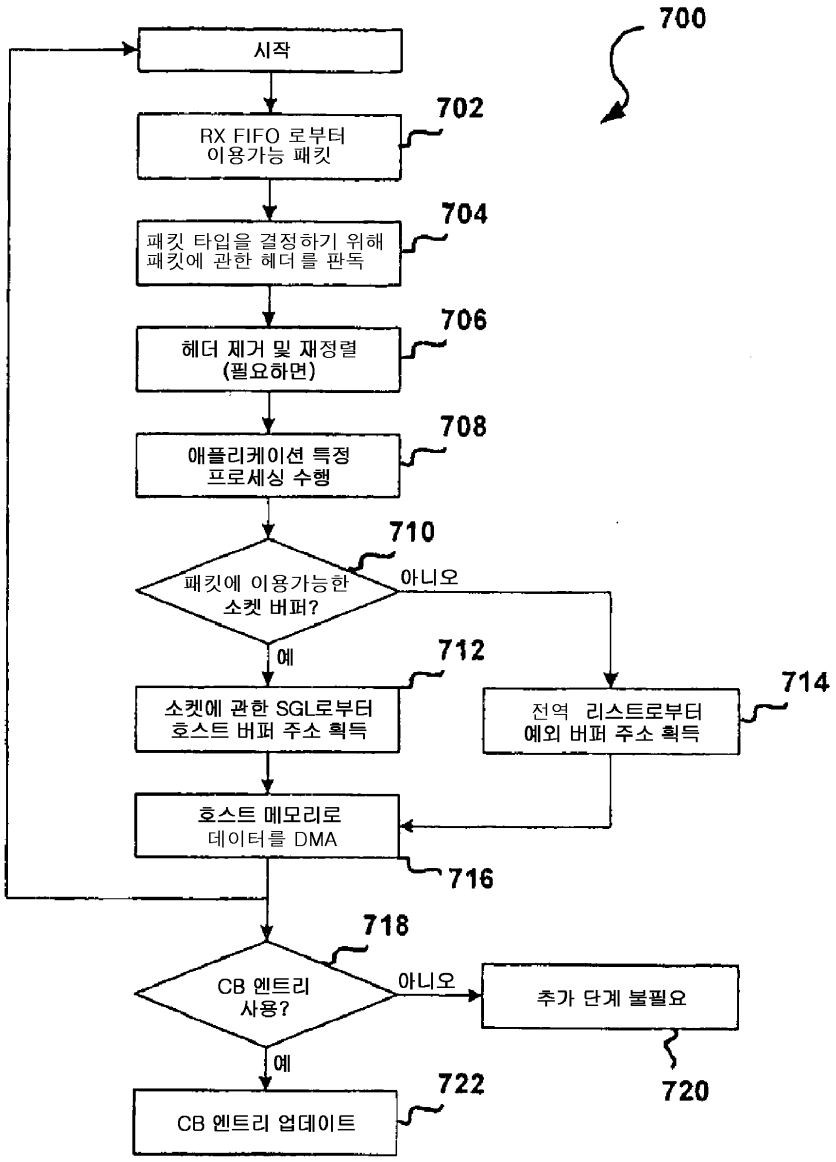
도면5


RX DMA 로직으로
326
506
508
SGL 메모리
예외 버퍼
리스트
SGL 프로세싱
TCP RX CB
업데이트 및
상태 메시지
요청
상태 메시지
생성기로
CB 캐시로
CB 캐시로
504
애플리케이션
특정 프로세싱
CB 캐시로
502
데이터 정렬 및
RX 버퍼 헤더
분석
RX 후단 제어기
322
TCP RX 버퍼

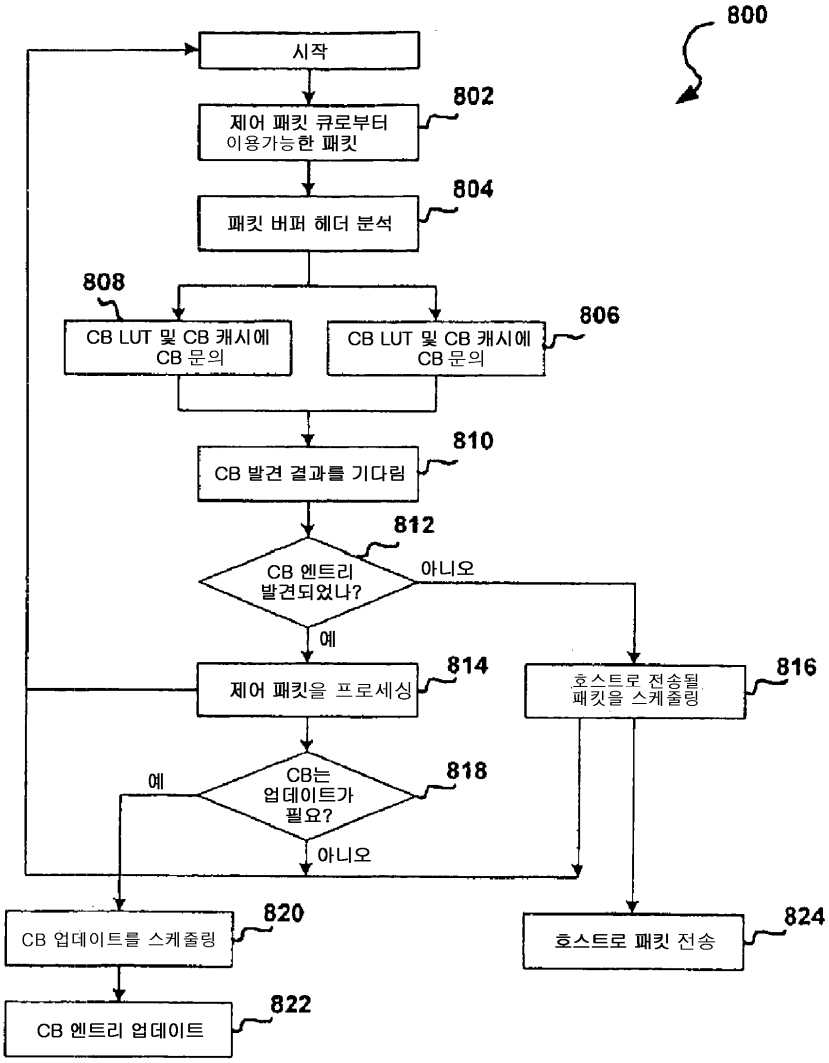
도면6


600
시작
602
IP 계층은 패킷이 이용가능하다는 것을 표시
604
RX 분석기가 이용가능한가?
아니오
606
이용가능한 분석기를 기다림
예
608
RX 분석기로 전송된 패킷
624
해시를 생성
610
TCP 제어 패킷?
예
아니오
612
제어 패킷 큐에 패킷을 추가
618
CB LUT 및 CB 캐시에 CB 문의
CB LUT 및 CB 캐시에 CB 문의
RX FIFO에 데이터를 저장
614
616
620
CB 발견 결과를 기다림
622
(데이터 섹션에 프리펜딩된) RX FIFO에 패킷 헤더를 기록

도면7


700
시작
RX FIFO 로부터
이용가능 패킷
702
패킷 타입을 결정하기 위해
패킷에 관한 헤더를 판독
704
헤더 제거 및 재정렬
(필요하면)
706
애플리케이션 특정
프로세싱 수행
708
패킷에 이용가능한
소켓 버퍼?
710
아니오
예
소켓에 관한 SGL로부터
호스트 버퍼 주소 획득
712
전역 리스트로부터
예외 버퍼 주소 획득
714
호스트 메모리로
데이터를 DMA
716
CB 엔트리
사용?
718
아니오
추가 단계 불필요
720
예
CB 엔트리 업데이트
722

도면8


800
시작
802
제어 패킷 큐로부터 이용가능한 패킷
804
패킷 버퍼 헤더 분석
808
CB LUT 및 CB 캐시에 CB 문의
806
CB LUT 및 CB 캐시에 CB 문의
810
CB 발견 결과를 기다림
812
CB 엔트리 발견되었나?
아니오
예
814
제어 패킷을 프로세싱
816
호스트로 전송될 패킷을 스케줄링
예
818
CB는 업데이트가 필요?
아니오
820
CB 업데이트를 스케줄링
824
호스트로 패킷 전송
822
CB 엔트리 업데이트