(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0065070
(43) 공개일자 2018년06월18일

(51) 국제특허분류(Int. Cl.)
H04L 12/24 (2006.01) *H04L 12/26* (2006.01)
(52) CPC특허분류
H04L 41/0654 (2013.01)
H04L 41/0695 (2013.01)
(21) 출원번호 10-2016-0165201
(22) 출원일자 2016년12월06일
심사청구일자 2016년12월06일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
백상헌
서울특별시 서초구 고무래로 35, 109동 1004호(반포동, 반포리체아파트)
서동은
서울특별시 마포구 도화2길 30 501호 (도화동,대원빌리지)
백호성
서울특별시 성북구 길음로 118, 410동 901호(길음동 대림e편한세상아파트)
(74) 대리인
김등용, 김홍석

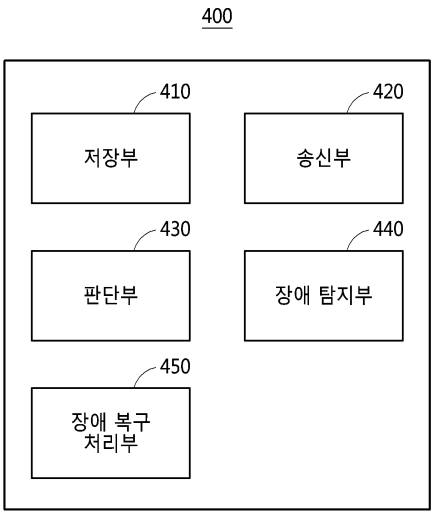
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 **서비스 기능 체인의 고가용성을 위한 분산적인 서비스 기능 장애 복구를 위한 방법 및 시스템**

(57) 요 약

장애 복구 방법이 개시된다. 상기 장애 복구 방법은 장애 복구 시스템에 의해 수행되고, 상기 장애 복구 시스템에 포함되는 송신부가 서비스 기능 체인 상의 다음 서비스 기능을 제공하는 서비스 노드로 핑(ping) 메시지를 일정 주기로 송신하는 단계, 상기 장애 복구 시스템에 포함되는 판단부가 상기 핑 메시지에 대한 응답의 수신 여부를 판단하는 단계, 및 상기 장애 복구 시스템에 포함되는 장애 탐지부가, 상기 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신되지 않은 경우 상기 다음 서비스 기능의 장애를 탐지하는 단계를 포함한다.

대 표 도 - 도4



(52) CPC특허분류
H04L 43/10 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711028339
부처명 미래창조과학부
연구관리전문기관 정보통신산업진흥원
연구사업명 방송통신산업기술개발
연구과제명 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발
기 여 율 1/2
주관기관 광주과학기술원
연구기간 2015.06.01 ~ 2016.05.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711035256
부처명 미래창조과학부
연구관리전문기관 정보통신산업진흥원
연구사업명 정보통신기술인력양성
연구과제명 초고속 무선 네트워크 기반 u-Office 서비스 연구개발
기 여 율 1/2
주관기관 중앙대학교 산학협력단
연구기간 2016.01.01 ~ 2016.12.31

명 세 서

청구범위

청구항 1

장애 복구 시스템에 의해 수행되는 장애 복구 방법에 있어서,

상기 장애 복구 시스템에 포함되는 송신부가 서비스 기능 체인 상의 다음 서비스 기능을 제공하는 서비스 노드로 핑(ping) 메시지를 일정 주기로 송신하는 단계;

상기 장애 복구 시스템에 포함되는 판단부가 상기 핑 메시지에 대한 응답의 수신 여부를 판단하는 단계; 및

상기 장애 복구 시스템에 포함되는 장애 탐지부가, 상기 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신되지 않은 경우 상기 다음 서비스 기능의 장애를 탐지하는 단계

를 포함하는 장애 복구 방법.

청구항 2

제1항에 있어서,

상기 장애 복구 시스템에 포함되는 장애 복구 처리부가, 상기 장애가 탐지된 경우 상기 다음 서비스 기능에 대한 백업 인스턴스를 활성화시키는 단계; 및

상기 장애 복구 처리부가, 상기 장애 복구 시스템으로 유입된 패킷들이 상기 장애 복구 시스템이 제공하는 서비스 기능에 의해 처리된 후, 상기 백업 인스턴스가 활성화된 상기 다음 서비스 기능에 전달되어 처리될 수 있도록 내부 네트워크를 업데이트 하는 단계

를 더 포함하는 장애 복구 방법.

청구항 3

제2항에 있어서,

상기 장애 복구 시스템이 제공하는 서비스 기능으로 유입되는 패킷들은 상기 서비스 기능 및 상기 다음 서비스 기능에서 처리 후 네트워크 서비스 헤더(Network Service Header, NSH)의 서비스 인덱스(Service Index, SI)가 일정치씩 감소되는 것을 특징으로 하는 장애 복구 방법.

청구항 4

제3항에 있어서,

상기 서비스 인덱스는 2씩 감소되는 것을 특징으로 하는 장애 복구 방법.

청구항 5

i+1번째 서비스 기능에 대한 백업 인스턴스를 유지하는 저장부;

i+1번째 서비스 노드로 핑 메시지를 일정 주기로 송신하는 송신부;

상기 핑 메시지에 대한 응답의 수신 여부를 판단하는 판단부; 및

상기 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신하지 않는 경우 i+1번째 서비스 기능의 장애

를 탐지하는 장애 탐지부

를 포함하는 장애 복구 시스템.

청구항 6

제5항에 있어서,

상기 백업 인스턴스를 활성화하여 장애를 복구하는 장애 복구 처리부

를 더 포함하는 장애 복구 시스템.

청구항 7

제6항에 있어서,

상기 장애 복구 처리부는,

i+1번째 서비스 기능의 장애가 탐지되면, 상기 i+1번째 서비스 기능에 대한 상기 백업 인스턴스를 활성화시키고,

상기 장애 복구 시스템이 제공하는 i번째 서비스 기능으로 유입된 패킷들이 상기 i번째 서비스 기능에 의해 처리된 후, 상기 백업 인스턴스가 활성화된 상기 i+1번째 서비스 기능에 전달되어 처리될 수 있도록 내부 네트워크를 업데이트 하는 장애 복구 시스템.

청구항 8

제7항에 있어서,

상기 장애 복구 처리부는,

상기 i번째 서비스 기능 및 상기 i+1번째 서비스 기능에서 처리 후 네트워크 서비스 헤더의 서비스 인덱스가 일정치씩 감소되도록 i번째 서비스 기능으로 유입되는 패킷들을 처리하는 것을 특징으로 하는 장애 복구 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 서비스 기능 체인 상의 서비스 기능 장애 발생 시, 이에 대처하기 위한 분산적인 서비스 기능 장애 복구 방법에 관한 것이다.

배 경 기 술

[0003] 네트워크 기능 가상화 기술은 Firewall, NAT(Network Address Translation), IDS(Intrusion Detection System) 등의 네트워크 서비스 기능을 제공하는 하드웨어 장비를 소프트웨어화하여 상용 서버 상에 구현하는 기술로, 비용 감소 및 유연성 향상 등의 장점들로 인해 네트워크 사업자들로부터 많은 관심을 받고 있다. 서비스 기능 체이닝 기술은 이러한 서비스 기능들을 정책에 따라 하나의 논리적인 서비스 순서 리스트 (즉, 서비스 기능 체인)으로 구성한 뒤 네트워크에 인입된 트래픽을 순차적으로 전달 및 처리하는 기술이다. 한편, 체인 상의 서비스 기능에 장애 발생 시, 해당 서비스를 받아야 하는 트래픽은 높은 지연시간을 경험할 수 있고, 이를 방지하기 위해 서비스 기능 장애에 빠르게 대처하기 위한 장애 극복 방법이 필요하다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 미합중국 특허출원 제14/572,335호 "Methods, systems, and computer readable storage devices for managing faults in a virtual machine network"

(특허문헌 0002) 미합중국 특허출원 제14/572,716호 "System, method, and computer program for preserving service continuity in a network function virtualization (NFV) based communication network"

발명의 내용

해결하려는 과제

[0006] 본 발명은 서비스 기능 체인 상의 서비스 기능에 대한 분산적인 장애 극복 방법을 통해 서비스 기능 장애 극복 시간을 줄이는 것을 목적으로 한다.

[0007] 본 발명은 각 서비스 기능에 DFA(Distributed Fail-over Agent)를 위치시켜 체인 상의 다음 서비스 기능에 대한 장애 탐지 및 장애 극복 메커니즘을 수행하도록 하고, 이를 통해 중앙 집중적인 SFC 컨트롤러를 통해서만 이루어질 수 있는 체인 업데이트 과정 없이 분산적으로 장애를 극복하여 서비스 기능 장애 극복시간을 줄이는 것을 목적으로 한다.

과제의 해결 수단

[0009] 본 발명의 일 실시 예에 의한 장애 복구 방법은 장애 복구 시스템에 의해 수행되고, 상기 장애 복구 시스템에 포함되는 송신부가 서비스 기능 체인 상의 다음 서비스 기능을 제공하는 서비스 노드로 핑(ping) 메시지를 일정 주기로 송신하는 단계, 상기 장애 복구 시스템에 포함되는 판단부가 상기 핑 메시지에 대한 응답의 수신 여부를 판단하는 단계, 및 상기 장애 복구 시스템에 포함되는 장애 탐지부가, 상기 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신되지 않은 경우 상기 다음 서비스 기능의 장애를 탐지하는 단계를 포함한다.

[0010] 본 발명의 일 실시 예에 의한 장애 복구 시스템은 i+1번째 서비스 기능에 대한 백업 인스턴스를 유지하는 저장부, i+1번째 서비스 노드로 핑 메시지를 일정 주기로 송신하는 송신부, 상기 핑 메시지에 대한 응답의 수신 여부를 판단하는 판단부, 및 상기 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신하지 않는 경우 i+1번째 서비스 기능의 장애를 탐지하는 장애 탐지부를 포함한다.

발명의 효과

[0012] 일실시예에 따르면, 서비스 기능 체인 상의 서비스 기능에 대한 분산적인 장애 극복 방법을 통해 서비스 기능 장애 극복 시간을 줄일 수 있다.

[0013] 일실시예에 따르면, 각 서비스 기능에 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)를 위치시켜 체인 상의 다음 서비스 기능에 대한 장애 탐지 및 장애 극복 메커니즘을 수행할 수 있다.

[0014] 일실시예에 따르면, 중앙 집중적인 서비스 기능 체이닝 컨트롤러를 통해서만 이루어질 수 있는 체인 업데이트 과정 없이 분산적으로 장애를 극복하여 서비스 기능 장애 극복시간을 줄일 수 있다.

도면의 간단한 설명

[0016] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 제어 평면과 데이터 평면으로 구성된 서비스 기능 체이닝(Service Function Chainning, SFC) 구조를 나타낸다.

도 2는 서비스 기능 체이닝의 컨트롤러를 통한 서비스 기능 장애 극복 과정을 나타낸다.

도 3은 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)를 통한 서비스 기능 장애 극복 과정을 나타낸다.

도 4는 일실시예에 따른 장애 복구 시스템을 설명하는 도면이다.

도 5는 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)의 분산적인 서비스 기능 장애 극복을 위한, 일실시예에 따른 장애 복구 프로세서의 동작 방법에 대한 플로우 차트이다.

도 6은 컨트롤러와 서비스 기능간의 지연시간의 범위를 변화시키면서 평균 장애극복 시간의 변화를 살펴본 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시 예들에 대해서 특정한 구조적 또는 기능적 설명은 단지 본 발명의 개념에 따른 실시 예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시 예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시 예들에 한정되지 않는다.

[0018] 본 발명의 개념에 따른 실시 예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시 예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시 예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0019] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0020] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0021] 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0022] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0023] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다.

[0025] 도 1은 제어 평면과 데이터 평면으로 구성된 서비스 기능 체이닝(Service Function Chaining, SFC) 구조(100)를 나타낸다.

[0026] 제어 평면의 컨트롤러는 SFC 테이블을 유지한다. 테이블 엔트리는 체인에 대한 Service Path Index(SPI), 정책, 서비스 기능들의 순서로 구성될 수 있다. 예를 들어, chain C의 경우, SPI가 P로 정의되어 있다. 체인이 부여된 패킷이 지나야 하는 실제 네트워크 상의 경로를 Service Function Path(SFP)라 하는데, SPI(Service Path Index)는 실제 네트워크 상의 경로에 대한 식별자이다. 또한 테이블 엔트리에는 해당 체인을 목적지 포트번호가

80번인 트래픽에 대해서만 적용하도록 하는 정책이 명시되어있다.

[0027] 다음으로는 Service Function(SF) 인스턴스들(또는 서비스 기능들)의 순서로 구성된 실제 네트워크 상의 경로가 SF 1 - SF 2 - SF 3 - SF 4의 순서로 명시되어 있다.

[0028] 본 명세서에서 사용되는 SF 1, SF 2 등 SF i의 표현은 1번째 서비스 기능, 2번째 서비스 기능, i번째 서비스 기능으로 해석될 수 있다.

[0029] 한편, 괄호안의 정보는 해당 서비스 기능이 연결되어 있는 서비스 기능 포워더(Service Function Forwarder, SFF)를 의미한다. 서비스 기능 포워더는 전달받은 패킷들을 정의된 체인에 따라 서비스 기능(또는 서비스 노드) 또는 서비스 기능 포워더에게 전달하는 역할을 수행한다.

[0030] 한편, 컨트롤러는 서비스 기능 인스턴스들(또는 서비스 기능들)을 데이터 평면상의 각 Service Node(SN)에 설치한다. SN은 CPU, 메모리, 저장 공간을 제공하는 물리적인 또는 가상의 요소로, NFV(Network Function Virtualization) 기술을 통해 VM(Virtual Machine) 형태로 서비스 기능 인스턴스(또는 서비스 기능)를 호스팅할 수 있는 기능을 제공한다. 컨트롤러는 새로운 체인 구성에 대한 요청을 받으면, SFC 테이블에 새로운 엔트리를 입력한다. 엔트리가 입력되면, 컨트롤러는 정책에 따라 특정 트래픽이 실제 네트워크 상의 경로에 정의된 순서에 따라 서비스를 받을 수 있도록, 서비스 분류기(Service Classifier, SC) 및 서비스 기능 포워더의 SFC 테이블을 업데이트한다.

[0031] 또한, 컨트롤러는 주기적인 핑 메시지를 통해 각 서비스 기능들에 대한 장애 탐지를 수행할 수 있다. 즉, 미리 정의된 문턱 값인 T_o의 시간 동안 핑 메시지에 대한 응답이 없을 시, 해당 서비스 기능에 장애가 발생하였다고 판단할 수 있다.

[0032] 만약, 특정 서비스 기능 인스턴스에 대한 장애가 탐지되면, 컨트롤러는 해당 서비스 기능에 대한 서비스 기능 인스턴스를 장애가 발생한 서비스 기능 인스턴스와 같은 서비스 기능 포워더와 연결된 서비스 기능 인스턴스들(또는 서비스 노드들) 중 자신과의 통신 지연시간이 최소인 서비스 기능 인스턴스(또는 서비스 노드)와 같은 위치에 설치하고, 새로운 서비스 기능 인스턴스(또는 새로운 서비스 기능이 설치된 서비스 노드)로 트래픽이 지날 수 있도록 새로운 실제 네트워크 상의 경로를 구성할 수 있다.

[0033] 데이터 평면의 인그레스 노드(ingress node)는 SFC 도메인으로 유입되는 패킷들에 대해 컨트롤러로부터 전달받은 네트워크 정책에 따라 네트워크 서비스 헤더(Network Service Header, NSH)를 부여하여 해당 패킷들이 실제 네트워크 상의 경로에 정의된 순서에 따라 서비스를 받을 수 있도록 한다. NSH에는 SPI 및 서비스 인덱스(Service Index, SI) 정보가 포함되어 있는데, SI는 각 서비스 기능 인스턴스에 의해 1씩 감소되어 해당 패킷이 현재 실제 네트워크 상의 경로 상의 어떤 순서까지 처리되었는지를 나타낸다. 예를 들어, 인그레스 노드는 정책에 따라 목적지 포트번호가 80번인 패킷에 대해 SPI=P 및 SI=255를 부여한 것을 확인할 수 있다.

[0034] 인그레스 노드는 NSH 부여 후, 패킷을 실제 네트워크 상의 경로에 명시된 첫 번째 서비스 기능 인스턴스에 연결된 서비스 기능 포워더로 전달한다. 서비스 기능 포워더는 SPI, SI 그리고 Next Hop(NH)로 구성된 서비스 기능 포워더 테이블을 통해 해당 패킷을 전달한다. SC로부터 전달받은 패킷의 SPI, SI가 각각 P, 255이므로 이에 해당하는 NH인 SF 1로 패킷을 전달한다.

[0035] SF 1은 패킷을 처리 후, SI값을 1만큼 감소시킨 뒤, 다시 서비스 기능 포워더 1에게 전달한다. 이러한 과정을 반복하여 해당 패킷은 서비스 기능 포워더 테이블 상에서 1 - 2 - 3 - 4 - 5의 순서로 매칭되며 순서대로 서비스 기능 인스턴스들을 지나게 된다.

[0036] 앞서 언급하였듯, 서비스 기능 포워더 테이블은 컨트롤러 상에서 새로운 실제 네트워크 상의 경로 구성 시, 컨트롤러에 의해 업데이트 된다. 인그레스 노드는 SFC 도메인 내에서 처리가 완료된 패킷들을 SFC 도메인 외부로 전달하는 역할을 수행한다.

[0037] 본 발명에서는 실제 네트워크 상의 경로에 포함된 모든 서비스 기능 인스턴스들이 위치한 SN들에 분산 페일오버 에이전트(실시 예에 따라 분산 페일오버부로 명명될 수도 있음)가 위치한다. 실제 네트워크 상의 경로에 포함된 모든 SF들이 같은 서비스 기능 포워더에 연결되어 있다고 가정할 수 있다. 또한 실제 네트워크 상의 경로를 SF 1 - SF 2 - SF 3 - ... - SF N로 나태내고, i번째 서비스 기능에 위치한 분산 페일오버 에이전트를 i번째 분산 페일오버 에이전트라고 하면, i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능이 i번째 서비스 기능과 같은 서비스 기능 포워더에 연결되어 있을 시, i+1번째 서비스 기능에 대한 장애 탐지 및 복구를 수행할 수 있다.

[0038] 먼저 i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능에 대한 장애 탐지를 수행할 수 있다. 즉, i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능(또는 i+1번째 서비스 노드)에게 주기적인 핑 메시지를 보내고, 미리 정의된 문턱값인 T_o의 시간 동안 응답이 없을 시, i+1번째 서비스 기능(또는 i+1번째 서비스 노드)에 장애가 발생하였다고 판단할 수 있다.

[0039] 한편, i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능에 대한 백업 인스턴스를 유지한다. i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능의 장애 판단 시, i+1번째 서비스 기능에 대한 백업 인스턴스를 활성화시키고, i번째 서비스 기능으로 유입된 패킷들이 i번째 서비스 기능에 의해 처리된 후, 새로 활성화된 i+1번째 서비스 기능에 전달되어 처리될 수 있도록 내부 네트워크를 업데이트 한다.

[0040] 예를 들어, 도면 1에서 SF 2에 장애 발생 시, 분산 페일오버 에이전트 1은 비활성화되어 있던 SF 2를 활성화 시키고, 내부 네트워크를 업데이트 한다. 이후 SF 1로 유입되는 패킷들은 SF 1에서 처리된 후, 같은 SN 1에 위치한 SF 2에서 처리된다.

[0041] 이로 인해 SF 1로 유입되는 패킷들은 SF 1 및 SF 2에서 처리가 되어 NSH의 SI 값이 2씩 감소된 뒤, 서비스 기능 포워더 1로 전달된다. 전달받은 패킷은 SI 값이 253이므로, 장애가 발생한 SF 2 인스턴스로 전달되지 않고, 바로 SF 3 인스턴스로 전달된다. 따라서 컨트롤러에 의한 실제 네트워크 상의 경로 생성 및 서비스 기능 포워더 테이블 업데이트 과정을 거치지 않고도 장애를 극복할 수 있게 된다.

[0042] 한편, SF 1에 장애가 발생하였을 경우 혹은 i+1번째 서비스 기능이 i번째 서비스 기능과 서로 다른 서비스 기능 포워더에 연결되어 있을 때 i+1번째 서비스 기능에 장애가 발생하였을 경우, 해당 장애들을 탐지/복구할 분산 페일오버 에이전트가 없으므로, 컨트롤러가 앞서 언급한 장애 탐지/복구 메커니즘을 통해 실제 네트워크 상의 경로 생성 및 서비스 기능 포워더 테이블 업데이트 과정을 수행하여 장애를 극복할 수 있다.

[0044] 도 2는 SFC 컨트롤러를 통한 SF 장애 극복 과정을 나타내는 실시예(200)이다.

[0045] 도 2는 i번째 서비스 기능에 장애 발생 시, SFC 컨트롤러에 의한 장애 복구 과정 및 이 때의 장애 복구 시간 t_i^C을 나타낸다. 먼저 컨트롤러는 주기적으로 i번째 서비스 기능의 상태를 확인하며, T_o의 시간 동안 응답이 없을 시 i번째 서비스 기능에 장애가 발생하였다고 판단한다. 따라서 장애를 탐지하기까지는 T_o의 시간이 소요된다. i번째 서비스 기능은 SN i에 설치되어있다고 하고, l_i를 SFC 컨트롤러와 i번째 서비스 기능(또는 SN i) 간의 네트워크 지연시간으로 가정할 수 있다. 또한, SN i와 같은 서비스 기능 포워더에 연결된 SN들 중 컨트롤러와의 지연시간이 가장 낮은 SN을 j라 하면, SFC 컨트롤러는 SN j에게 i번째 서비스 기능에 대한 설치 요청을 보내고, 이 때 걸리는 시간은 l_i이다. 그 후, 설치 요청을 받은 SN j가 i번째 서비스 기능을 설치하게 되며, 이 때 걸리는 시간을 T_p라 한다. i번째 서비스 기능의 설치가 완료되면, SN j는 해당 작업에 대한 ACK을 보내고, 컨트롤러는 ACK을 받은 이후 서비스 기능 포워더 테이블을 업데이트한다. $2*l_{SFF}$를 컨트롤러와 서비스 기능 포워더 간의 네트워크 지연시간이라 하면, 업데이트 요청 후 서비스 기능 포워더를 받을 때까지 걸리는 시간은 $2*l_{SFF}$가 된다. 결과적으로 t_i^C 는 [수학식 1]과 같이 계산된다.

[0047] [수학식 1]

[0048] $$t_i^C = T_o + 2l_j + T_p + 2l_{SFF}, \quad \text{for} \, 1 \le i \le N$$

[0050] 이때, T_o는 장애 탐지 타임아웃 값으로 해석될 수 있고, T_p는 SF 설치에 걸리는 시간으로 해석될 수 있으며, l_i는 i번째 서비스 기능과 SFC controller 사이의 네트워크 지연시간으로 해석될 수 있고, l_{SFF}는 서비스 기능 포워더와 SFC controller 사이의 네트워크 지연시간으로 해석될 수 있으며, t_i^C는 centralized fail-over의 경우, i번째 서비스 기능에 대한 장애 복구시간으로 해석될 수 있고, t_{avg}^C는 centralized fail-over의 경우, 평균 장애 복구시간으로 해석될 수 있으며, t_i^D는 distributed fail-over의 경우, i번째 서비스 기능에 대한 장애 복구시간으로 해석될 수 있으며, t_{avg}^D는 distributed fail-over의 경우, 평균 장애 복구시간으로 해석될 수 있다.

[0051] 이 때, $j = \arg_x(\min l_x)$이다. SFC 컨트롤러를 통한 평균 장애 극복 시간 t_{avg}^C는 [수학식 2]과 같이 계산된다.

[0053] [수학식 2]

[0054] $$t_{avg}^C = \sum_{i=1}^{N} t_i^C / N$$

[0056] 이때, t_i^C는 centralized fail-over의 경우, i번째 서비스 기능에 대한 장애 복구시간으로 해석될 수 있다.

[0058] 도 3은 분산 페일오버 에이전트를 통한 서비스 기능 장애 극복 과정을 나타낸다.

[0059] 즉, 도 3은 i가 2 이상일 때(즉, i가 실제 네트워크 상의 경로의 첫 번째 SF가 아닐 때) i번째 서비스 기능에 장애 발생 시, 분산 페일오버 에이전트를 통한 장애 극복 과정 및 이 때의 장애 복구 시간 t_i^D를 타나낸다.

[0060] i번째 분산 페일오버 에이전트-1은 주기적으로 i번째 서비스 기능의 상태를 확인하며, T_o의 시간 동안 응답이 없을 시 i번째 서비스 기능에 장애가 발생하였다고 판단한다. 따라서 장애를 탐지하기까지는 T_o의 시간이 소요된다. i-1번째 분산 페일오버 에이전트는 장애를 탐지하면 SN i-1에 위치한 back-up i번째 서비스 기능을 활성화시키고, 내부 네트워크를 업데이트 시켜 i번째 서비스 기능이 SN i-1에에 제공될 수 있도록 한다.

[0061] 이 때 걸리는 시간은 i번째 서비스 기능을 설치하는 데에 걸리는 시간 T_p와 동일하다고 가정한다. i번째 서비스 기능에 대한 복구 시간은 [수학식 3]과 같다.

[0063] [수학식 3]

[0064] $$t_i^D = \begin{cases} T_o + T_p, & \text{for } 2 \le i \le N \\ t_i^C, & \text{for } i = 1 \end{cases}$$

[0066] T_o는 장애 탐지 타임아웃 값으로 해석될 수 있고, T_p는 서비스 기능 설치에 걸리는 시간으로 해석될 수 있으며, t_i^C는 centralized fail-over의 경우, i번째 서비스 기능에 대한 장애 복구시간으로 해석될 수 있다.

[0067] 한편, i가 1일 경우, 장애 복구는 컨트롤러에 의해 수행되므로, 장애 복구 시간은 t_i^C이 된다. 분산 페일오버 에이전트를 통한 평균 장애 극복시간 t_{avg}^D는 [수학식 4]와 같이 계산된다.

[0069] [수학식 4]

[0070] $$t_{avg}^D = \sum_{i=1}^{N} t_i^D / N$$

[0072] 도 4는 일실시예에 따른 장애 복구 시스템을 설명하는 도면이다.

[0073] 도 4에서는 서비스 기능(SF)에 장애 발생 시 분산적으로 장애 탐지/복구를 수행하도록 하는 구조/기법 등이 제안된다.

[0074] 구체적으로, 본 발명에 따르면, SN에 분산 페일오버 에이전트(Distributed Fail-over Agent) 및 Back-up SF를 위치시킬 수 있다. 또한, 분산 페일오버 에이전트는 분산적인 장애 탐지 및 Back-up SF를 통한 분산적인 장애 극복을 수행할 수 있다. 본 발명에 따르면, 기존의 SFC 컨트롤러 기반의 장애 복구 기법 대비 짧은 장애 극복시간을 갖고, SFC 컨트롤러를 통하지 않고도 서비스 기능의 장애 복구가 가능하다. 또한, 실시 예에 따라 도 4에 도시된 장애 복구 시스템(400)은 서비스 노드를 의미할 수도 있다.

[0075] 이를 위해, 일 실시 예에 따른 장애 복구 시스템(400)은 저장부(410), 송신부(420), 판단부(430), 및 장애 탐지부(440)를 포함할 수 있다. 또한, 장애 복구 시스템(400)은 장애 복구 장치로 명명될 수도 있다.

[0076] 일 실시 예에 따른 저장부(410)는 i+1번째 서비스 기능(SF i+1)에 대한 백업 인스턴스(back-up instance)를 유지할 수 있다. 한편, 송신부(420)는 i+1번째 서비스 기능(또는 i+1번째 서비스 노드)에 일정 주기로 핑(ping) 메시지를 송신할 수 있다. 판단부(430)에서는 송신된 핑(ping) 메시지의 수신 여부(또는 상기 핑 메시지에 대한 응답의 수신 여부)를 판단할 수 있다.

[0077] 일 실시 예에 따른 장애 탐지부(440)는 송신된 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신하지 않는 경우 i+1번째 서비스 기능(Service Function, SF) 또는 i+1번째 서비스 노드의 장애를 탐지할 수 있다.

[0078] 이때, 다른 일 실시 예에 따른 장애 복구 시스템(400)은 백업 인스턴스(또는 백업 서비스 기능)를 활성화하여 장애를 복구하는 장애 복구 처리부(450)를 더 포함할 수 있다. 특히, 장애 복구 처리부(45)는 장애가 탐지되면, i+1번째 서비스 기능(Service Function, SF)에 대한 백업 인스턴스(back-up instance)를 활성화시킬 수 있다. 즉, i번째 서비스 기능(Service Function, SF)으로 유입된 패킷들이 i번째 서비스 기능(Service Function, SF)에 의해 처리된 후, 백업 인스턴스(back-up instance)가 활성화된 상기 i+1번째 서비스 기능(Service

Function, SF)에 전달되어 처리될 수 있도록 내부 네트워크를 업데이트할 수 있다.

[0079] 또한, 장애 복구 처리부(450)는 i번째 서비스 기능(Service Function, SF) 및 i+1번째 서비스 기능(Service Function, SF)에서 처리 후 네트워크 서비스 헤더(Network Service Header, NSH)의 서비스 인덱스(Service Index, SI)가 일정치씩 감소되도록 i번째 서비스 기능(또는 i번째 서비스 노드)으로 유입되는 패킷들을 처리할 수 있다.

[0080] 일 실시 예에 따른 장애 복구 처리부(450)는 i+1번째 서비스 기능(Service Function, SF)에서 장애가 탐지된 경우, 컨트롤러를 통해 실제 네트워크 상의 경로(Service Function Path, SFP)를 생성하고, 생성된 실제 네트워크 상의 경로(Service Function Path, SFP)에 상응하는 서비스 기능 포워더(Service Function Forwarder, SFF) 테이블을 업데이트하며, 업데이트된 서비스 기능 포워더(Service Function Forwarder, SFF) 테이블을 이용하여 장애를 처리할 수 있다.

[0082] 도 5는 분산 페일오버 에이전트의 분산적인 SF 장애 극복을 위한, 일실시예에 따른 장애 복구 프로세서의 동작 방법에 대한 플로우 차트이다.

[0083] 서비스 기능(Service Function, SF)에 위치하는 적어도 하나 이상의 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)를 유지할 수 있다.

[0084] 먼저, 일예에 따른 장애 복구 프로세서의 동작 방법은 i번째 분산 페일오버 에이전트는 i+1번째 서비스 기능에게 핑 메시지를 전송한다(단계 510). 즉, 적어도 하나 이상의 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA) 중에서, i번째 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)에서 i+1번째 서비스 기능(Service Function, SF)에 일정 주기로 핑(ping) 메시지를 송신할 수 있다.

[0085] 일 실시 예에 따른 장애 복구 프로세서의 동작 방법은 송신된 핑(ping) 메시지의 수신 여부를 판단할 수 있다(단계 520).

[0086] 다음으로, 일 실시 예에 따른 장애 복구 프로세서의 동작 방법은 To 시간 내에 응답이 수신되면, 다시 핑 메시지를 전송한다(단계 510). 만일 To 시간 내에 응답이 없을 시, i+1번째 서비스 기능의 back-up SF를 활성화시킨다(단계 530). 또한 i번째 서비스 기능에서 처리된 패킷들이 i+1번째 서비스 기능으로 전달되도록 내부 네트워크를 업데이트한다(단계 540).

[0087] 즉, 장애 복구 프로세서의 동작 방법은 송신된 핑 메시지에 대한 응답이 미리 정의된 문턱값 시간 내에 회신하지 않는 경우 i+1번째 서비스 기능(Service Function, SF)의 장애를 탐지할 수 있다. 또한, 장애가 탐지되는 경우, 상기 i번째 분산 페일오버 에이전트(Distributed Fail-over Agent, DFA)에서 상기 i+1번째 서비스 기능(Service Function, SF)에 대한 백업 인스턴스(back-up instance)를 활성화시킬 수 있다. 뿐만 아니라, i번째 서비스 기능(Service Function, SF)으로 유입된 패킷들이 상기 i번째 서비스 기능(Service Function, SF)에 의해 처리된 후, 백업 인스턴스(back-up instance)가 활성화된 i+1번째 서비스 기능(Service Function, SF)에 전달되어 처리될 수 있도록 내부 네트워크를 업데이트 할 수 있다.

[0088] 장애 복구 프로세서의 동작 방법은 i번째 서비스 기능(Service Function, SF)으로 유입되는 패킷들에 대해 상기 i번째 서비스 기능(Service Function, SF) 및 상기 i+1번째 서비스 기능(Service Function, SF)에서 처리 후 네트워크 서비스 헤더(Network Service Header, NSH)의 서비스 인덱스(Service Index, SI)가 일정치씩 감소되도록 할 수 있다.

[0089] 일 실시 예에 따른 장애 복구 프로세서의 동작 방법은 유입된 패킷의 서비스 인덱스(Service Index, SI)가 특정 값 이상인 경우, 상기 i+1번째 서비스 기능(Service Function, SF)을 스킵하고, i+2번째 서비스 기능(Service Function, SF)으로 상기 유입된 패킷을 전달할 수 있다.

[0090] 또한, 장애 복구 프로세서의 동작 방법은 상기 i+1번째 서비스 기능(Service Function, SF)에서 장애가 탐지된 경우, 컨트롤러를 통해 실제 네트워크 상의 경로(Service Function Path, SFP)를 생성하고, 상기 생성된 실제 네트워크 상의 경로(Service Function Path, SFP)에 상응하는 서비스 기능 포워더(Service Function Forwarder, SFF) 테이블을 업데이트하며, 및 상기 업데이트된 서비스 기능 포워더(Service Function Forwarder, SFF) 테이블을 이용하여 상기 장애를 처리할 수 있다.

[0092] 도 6은 컨트롤러와 SF간의 지연시간의 범위를 변화시키면서 평균 장애극복 시간의 변화를 살펴본 그래프(600)이다.

[0093] 즉, 도 6의 그래프(600)에서는 본 발명의 성능 평가를 위해 시뮬레이션을 진행하고 기존의 방법들과 비교를 하였다. 시뮬레이션 결과를 나타내는 도면에서 도면부호 620에 해당하는 Distributed Fail-over의 결과는 본 발명, 도면부호 610에 해당하는 Centralized Fail-over의 결과는 SFC 컨트롤러가 모든 SF들의 장애 탐지/복구를 수행하는 기술에 해당한다.

[0094] 그래프(600)는 컨트롤러와 SF들간의 지연시간의 범위를 변화시키면서 평균 장애 극복 시간의 변화를 나타낸다. 그래프(600)에서 볼 수 있듯이 본 발명의 지연시간이 더욱 낮은 평균 장애 극복시간을 달성하는 것을 확인할 수 있다. 이는 본 발명이 분산 페일오버 에이전트를 통한 분산적인 서비스 기능 장애 극복 방법을 통해 컨트롤러를 통한 서비스 기능 장애 극복을 최소화하였기 때문이다.

[0095] 결국, 본 발명을 이용하는 경우, 서비스 기능 체인 상의 서비스 기능에 대한 분산적인 장애 극복 방법을 통해 서비스 기능 장애 극복 시간을 줄일 수 있다. 또한, 각 서비스 기능에 분산 페일오버 에이전트(Distributed Fail-over Agent)를 위치시켜 체인 상의 다음 서비스 기능에 대한 장애 탐지 및 장애 극복 메커니즘을 수행할 수 있고, 중앙 집중적인 SFC 컨트롤러를 통해서만 이루어질 수 있는 체인 업데이트 과정 없이 분산적으로 장애를 극복하여 서비스 기능 장애 극복시간을 줄일 수 있다.

[0097] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시 예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

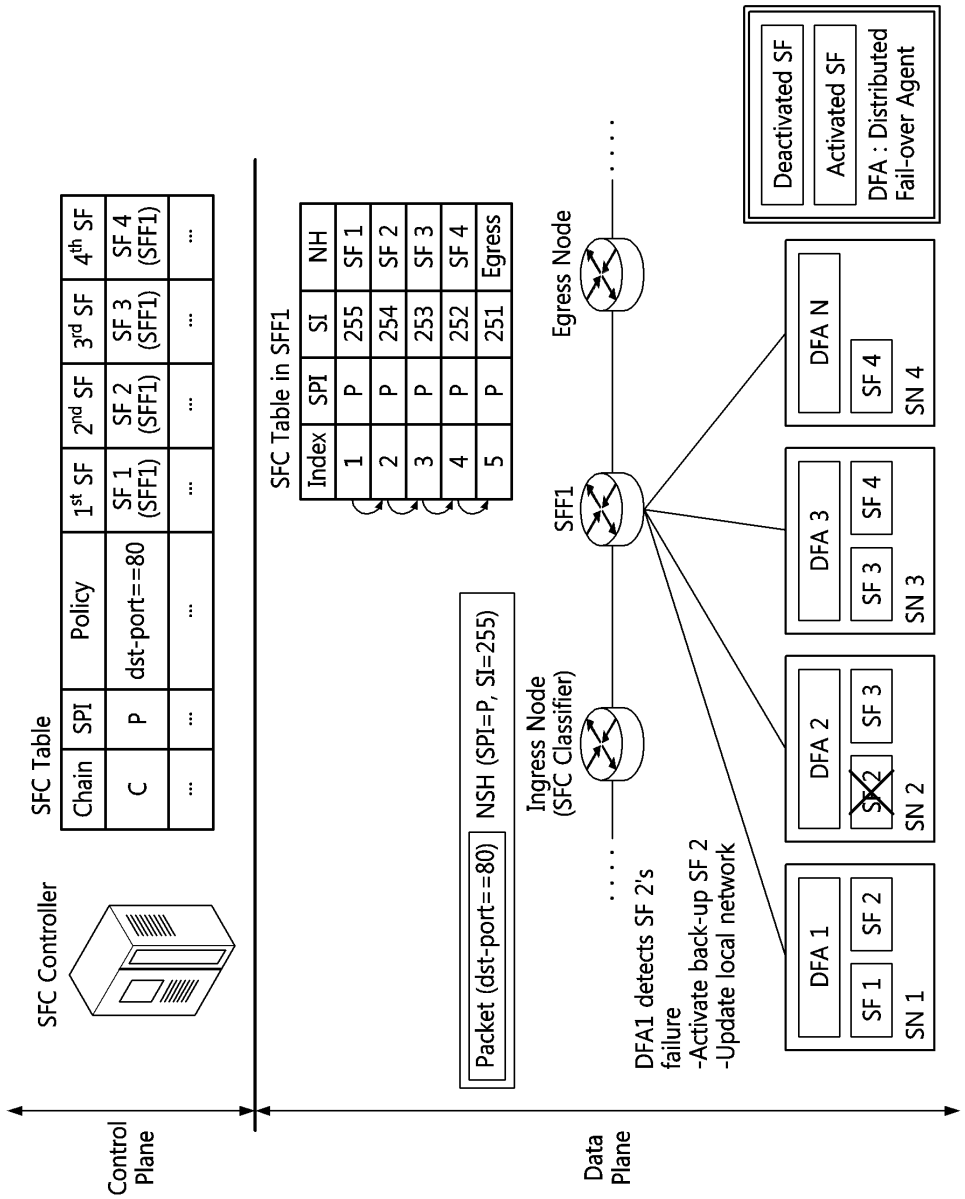
[0098] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0099] 실시 예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시 예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시 예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

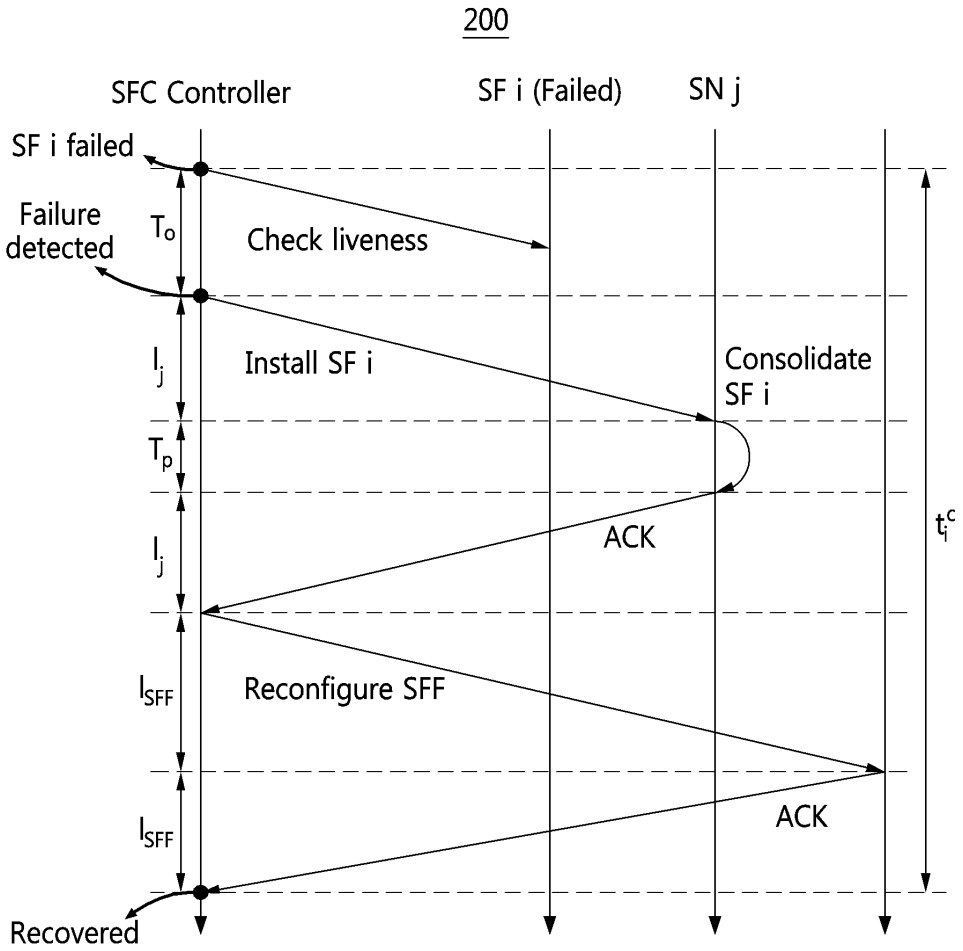
[0101] 본 발명은 도면에 도시된 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

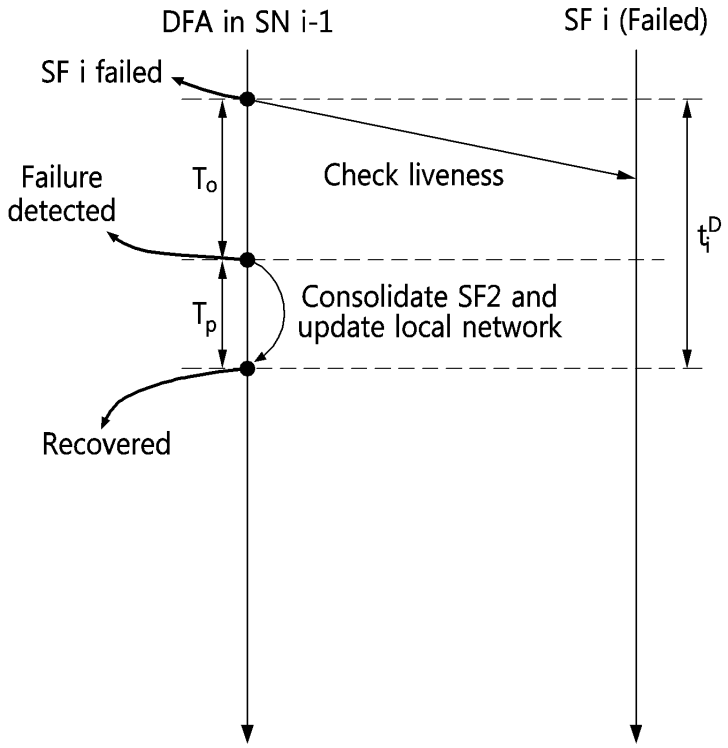
도면1



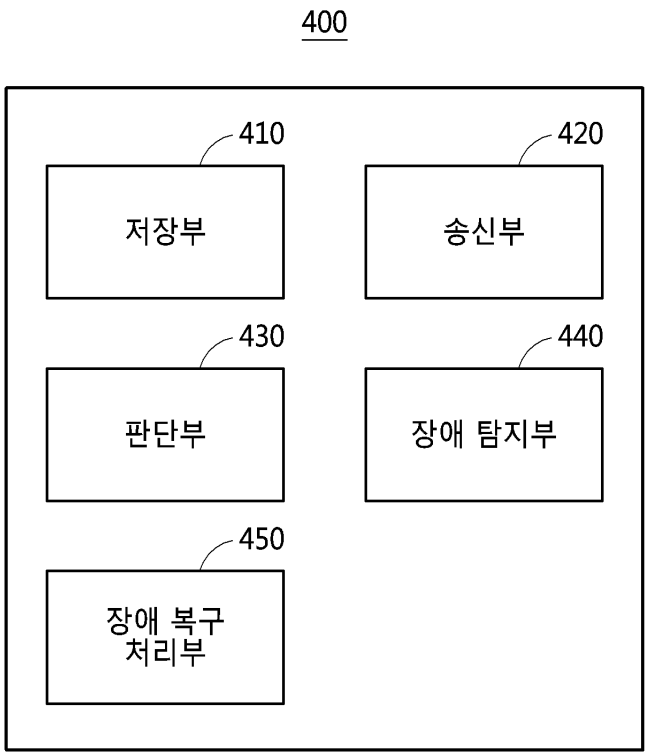
도면2


200
SFC Controller
SF i (Failed)
SN j
SF i failed
Failure detected
To
Check liveness
lj
Install SF i
Consolidate SF i
Tp
lj
ACK
tic
lSFF
Reconfigure SFF
lSFF
ACK
Recovered

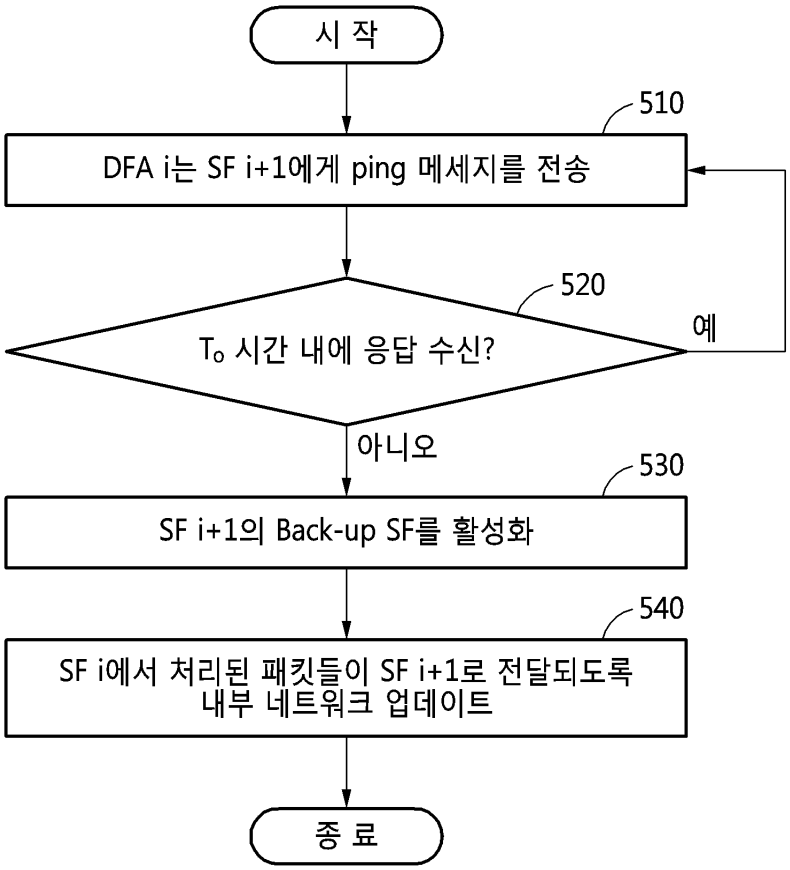
도면3


DFA in SN i-1
SF i (Failed)
SF i failed
Failure detected
To
Check liveness
tiD
Tp
Consolidate SF2 and update local network
Recovered

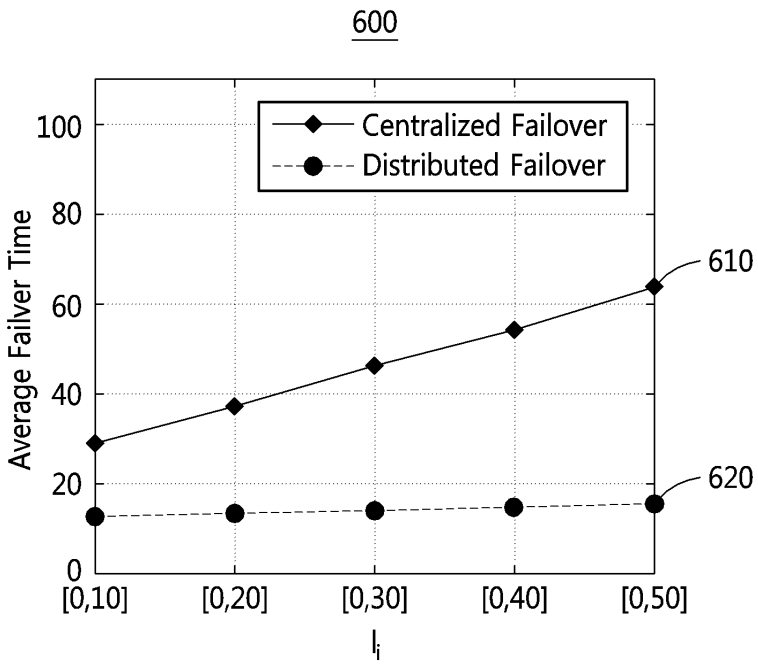
도면4


400
410
저장부
420
송신부
430
판단부
440
장애 탐지부
450
장애 복구 처리부

도면5


시 작
510
DFA i는 SF i+1에게 ping 메세지를 전송
520
T_0 시간 내에 응답 수신?
예
아니오
530
SF i+1의 Back-up SF를 활성화
540
SF i에서 처리된 패킷들이 SF i+1로 전달되도록
내부 네트워크 업데이트
종 료

도면6


600
Centralized Failover
Distributed Failover
Average Failver Time
100
80
60
40
20
0
610
620
[0,10]
[0,20]
[0,30]
[0,40]
[0,50]
l_i