

Path Computation Element 프로토콜 (PCEP)의 설계 및 구현 - FSM과 인터페이스*

이원혁* · 강승애** · 김현철***

요 약

백본(코어) 네트워크에서 유연하고 안정적인 품질이 보장되는 서비스 제공에 대한 요구가 폭발적으로 증가하면서 이러한 요구를 수용하기 위해 제어평면의 프로토콜로 MPLS/GMPLS가 도입되었다. GMPLS 기반 제어평면에서 경로계산 및 상호연동 기능은 최적의 서비스 품질을 제공하기 위한 핵심기술 중의 하나이다. 이를 위해 IETF에서는 Path Computation Element (PCE) 구조를 제안하였다. PCE는 경로계산 전용의 네트워크 요소이며 경로계산을 요청하는 Path Computation Clients (PCC)와 경로계산을 수행하는 PCE 간의 통신은 PCE 프로토콜 (PCEP)을 이용한다. 본 논문은 PCE 기반 경로계산 구조에 대해서 살펴보고 이를 기반으로 PCEP 설계 및 구현을 위한 몇 가지 요소를 제시하였다. 우선 본 논문에서는 PCEP Finite State Machine (FSM)을 포함하여 각 상태에서 필요한 동작 요소를 정리하였다. 또한 본 논문에서는 PCEP에서 제공하는 통신 채널을 효과적으로 제어하기 위한 내부/외부 인터페이스를 더불어 제안하였다.

Design and Implementation of Path Computation Element Protocol (PCEP) – FSM and Interfaces

Wonhyuk Lee* · Seungae Kang** · Hyuncheol Kim***

ABSTRACT

The increasing demand for fast, flexible and guaranteed Quality of Service (QoS) in core networks has caused to deploy MultiProtocol Label Switching (MPLS) and Generalized MPLS (GMPLS) control plane. In GMPLS control plane, path computation and cooperation processes are one of the crucial element to maintain an acceptable level of service. The Internet Engineering Task Force (IETF) has proposed the Path Computation Element (PCE) architecture. The PCE is a dedicated network element devoted to path computation process and communications between Path Computation Clients (PCC) and PCEs is realized through the PCE Protocol (PCEP). This paper examines the PCE-based path computation architecture to include the design and implementation of PCEP. The functional modules including Finite State Machine (FSM) and related key design issues of each state are presented. In particular we also discuss internal/external protocol interfaces that efficiently control the communication channels.

Key words : PCE, PCEP, GMPLS, QoS, Path Computation

접수일(2013년 9월 4일), 수정일(1차: 2013년 9월 15일),
게재확정일(2013년 9월 25일)

★ 이 논문은 2013년도 남서울대학교 교내연구지원에 의해
연구되었음.

* 한국과학기술정보연구원 (KISTI)

** 남서울대학교 운동건강학과

*** 남서울대학교 컴퓨터학과

1. 서론

모바일 서비스로 대표되는 다양한 통합 서비스의 유연한 제공을 위해서 네트워크 또한 각각의 서비스를 위한 개별망에서 이용자의 다양한 통합 욕구를 실현시켜 주는 통합망의 형태로 발전하고 있으며, 관련한 통신망 제어기술 또한 급속히 발전하고 있다. 또한 고속 대용량의 백본용 전달망 능력이 메트로 그리고 가입자 망으로까지 그 영역을 확장하고 있는 상황에서 제어평면을 이용한 연결 제어는 중요한 이슈로 부상하고 있으며 네트워크 또한 다계층 통합망의 형태로 발전하고 있다 [1].

다계층 통합망 표준 제어 프로토콜인 Generalized Multi-Protocol Label Switching (GMPLS)는 Packet Switching Capable (PSC), Ethernet Switching Capable (L2SC) 또는 Lambda Switching Capable (LSC) 등과 같이 다른 스위칭 방식을 제공하는 장치라도 동시에 제어할 수 있다. GMPLS에서 트래픽 엔지니어링 (TE: Traffic Engineering)을 지원하기 위해 제약 사항 (Constraint) 기반의 경로 계산을 수행한다 [1] [2].

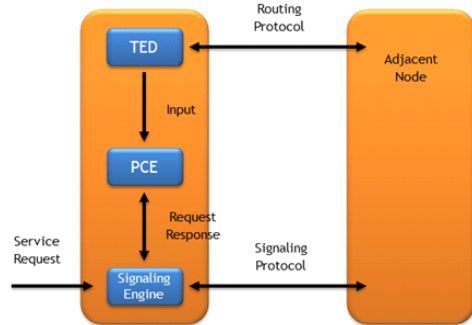
이처럼 다계층 정보를 기반으로 제약사항을 고려한 경로계산이 필요하며 이를 위해 제안된 방식이 하는 PCE (Path Computation Element) 기반 구조이다. 일반적으로 경로계산시 목적지까지의 모든 토폴로지를 알 수 없는 경우가 많고, 경로계산은 상당한 컴퓨팅 파워를 필요로 하기 때문에 경로계산을 수행하는 노드가 충분한 처리능력을 갖추지 못할 수도 있다. 따라서 PCE는 하나의 고성능 노드로 구현될 필요는 없으며 경로계산을 위해 분산 협업이 필수적이다. PCE 기반 네트워크 구조에서 경로계산을 요청하는 Path Computation Clients (PCC) 와 경로계산을 수행하는 PCE 간의 통신은 PCE 프로토콜 (PCEP)을 이용한다 [3] [4].

본 논문은 PCE 기반 네트워크에서 PCEP의 설계 및 구현 방안을 제안하였다. 아울러 본 논문에서는 PCEP를 구성하고 있는 기능 모듈과 핵심 설계 요소를 정리하였다. 특히 본 논문에서는 PCEP에서 제공하는 통신채널을 효과적으로 제어하기 위한 내부/외부 인터페이스 또한 제안하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 PCE 기반 네트워크 선행 기술의 조사 및 분석을 수행하였다. 3장에서는 PCEP 메시지 종류와 더불어 Finite State Machine (FSM)과 인터페이스들을 제안하였다. 마지막으로 4장에서는 본 논문의 결론에 대해서 기술하였다.

2. PCE 기반 네트워크 구조

PCE는 인그레스 노드에 장착되는 독립적인 모듈이나 특정 네트워크 서버에 포함된 독립적 개체가 될 수 있다. PCE의 주요 역할은 인그레스 노드 또는 라우터로부터 전달 받은 경로 계산 요청을 처리하는 것이다. 이러한 기능을 수행하는 PCE를 이용한 네트워크 구성 모델은 크게 Composite PCE, External PCE, Management-based PCE로 구분할 수 있다 [4][5].



(그림 1) Composite PCE

Composite PCE는 인그레스 노드나 라우터에서 구현될 수 있다. 또한 라우팅 프로토콜은 TE 정보를 교환하기 위해 사용되며 이러한 정보를 기반으로 TED가 구축된다. External PCE는 외부로 트래픽 전달에 참여하지 않는 특정 네트워크 서버 위에 구현될 수 있다. 이러한 구조에서 서비스 요청이 헤드엔드 노드에 전달되면 헤드엔드 노드는 시그널링을 시작하기 전에 경로계산을 하기 위해 external PCE에 경로계산 요청을 한다.

또한 네트워크 관리시스템 (NMS)는 목적지까지의 계산된 전체경로를 헤드엔드 노드에게 전달할 수 있

으며 이를 바탕으로 헤드엔드 노드는 시그널링을 수행할 수 있다. NMS에서의 경로는 PCE를 통해서 얻은 정보이거나 operator에 의해 제공된 정보일 수 있다. 이러한 형태의 구성을 Management 기반 PCE라 한다 [6][7].

3. PCEP

3.1 프로토콜 구조

PCEP는 request/response 형태의 프로토콜이며 대 표적 전달계층 프로토콜인 TCP를 기반으로 동작한다. 따라서 PCEP는 통신의 신뢰성과 순차적인 전송(in-order delivery)을 지원하며 보안과 관련된 모든 문제 역시 TCP에서 제공하는 기능을 사용한다. 또한 PCEP는 세션을 기반으로 하는 프로토콜이다. 즉 통신을 위해서 PCE와 PCC는 세션을 설정하여 서로의 파라미터를 협의하고 지원 능력(capabilities)을 학습한다. PCE와 PCC 사이에서 교환되는 모든 메시지들은 (표 1)과 같이 7가지 형태를 갖는다 [8][9][10].

(표 1) PCEP 메시지 종류

P C E P 메시지	내 용
Open	Open/Keepalive 메시지는 PCEP 세션의 개시(Initiate)와 유지(maintain)를 수행한다.
Keepalive	
PCReq	경로계산 요청을 위해 PCC로부터 PCE로 전달되는 PCEP 메시지
PCRep	경로계산 요청에 대한 응답으로 PCE로부터 PCC로 전달되는 PCEP 메시지. PCRep 메시지는 요청을 만족하는 하나 이상의 계산된 경로를 포함하고 있거나 만족하는 경로가 없는 경우에는 negative 응답을 보낸다.
PCNtf	PCEP notification 메시지는 PCC 에서 PCE, 또는 PCE 에서 PCC로 전송되며 특정 event 가 발생했음을 나타낸다.
PCErr	프로토콜 error 상황이 발생했을 때 전송되는 PCEP 메시지
Close	PCEP 세션을 종료하기 위해 사용되는 메시지

(표 2) PCEP FSM 변수

Variable	내 용
Connect	TCP 연결을 시작한 후에 시작되는 timer. 초단위로 동작하며 기본 값은 60초.
ConnectRetry	PCEP peer와 TCP 연결을 설정하기 위해 retry한 횟수
ConnectMaxRetry	PCEP peer와 TCP 연결을 설정하기 위해 retry할 수 있는 횟수의 최대값. Retry 횟수가 ConnectMaxRetry를 초과하는 경우에는 Idle 상태로 돌아간다.
OpenWait	TCP 연결설정이 완료된 이후에 Open 메시지를 수신하기 위해 대기할 수 있는 시간. OpenWait timer가 expire 되면 리소스를 release 하고 Idle 상태로 돌아간다.
KeepWait	Open 메시지를 송신한 후에 sender는 반드시 KeepWait 타이머를 구동시키는데, 만일 KeepWait timer가 expire 되었을 때까지 Keepalive 메시지나 PCErr 메시지를 수신하지 못했다면 PCErr 메시지를 전송하고 TCP 연결을 release 해야 한다.
OpenRetry	용납할 수 없는 PCEP session 값을 갖는 Open 메시지를 시스템에서 수신한 횟수

(표 3) PCEP FSM 타이머 변수

Variable	내 용
Keepalive timer	PCEP 메시지들간의 최소 시간 간격. 권고값은 30초.
DeadTimer	DeadTimer가 expire 되면 아무런 메시지를 PCEP peer로부터 받지 못했음을 나타내고 세션이 down 된 것으로 간주.
SyncTimer	SyncTimer의 목적은 misbehaving PCC 와 같은 이유로 여러 개의 path computation request 중에 하나를 잃어 버렸을 경우에 storage 낭비를 방지하기 위해서이다.

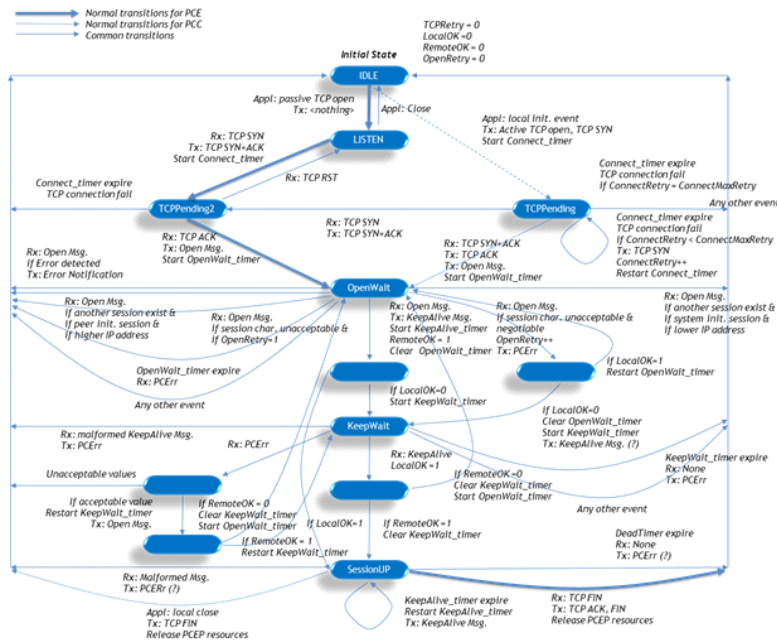
3.2 프로토콜 동작

PCE와 PCC 간의 통신을 위해 하나의 TCP 포트(Port 번호 4189)가 미리 할당되며 이 포트를 이용하여 PCE와 PCC 간의 TCP 연결이 설정될 수 있다. PCEP 초기화 단계는 ① PCE와 PCC TCP 연결 (3-way handshake)설정, ② TCP 연결을 이용한 PCEP 세션을 설정으로 이루어진다.

세션이 설정되고 난 이후에 PCE 또는 PCC는 peer

가 여전히 동작중인지에 대한 정보는 TCP가 제공하는 특정 기능을 통해 얻어질 수도 있지만, 적절한 시간 내에 상황을 파악하기는 어렵다. 이를 위해 PCEP에서는 Keepalive 타이머, DeadTimer, 그리고 Keepalive 메시지에 기초한 비대칭적인(asymmetrical) Keepalive 매커니즘을 제공한다. 즉 정해진 시간동안 Keepalive 메시지를 받지 못하는 경우 세션이 실패하였을

그리고 내부 처리 내용에 대해서 기술하고 있다. RFC 5440에서는 오직 5개의 상태만을 규정하고 있지만 본 논문에서는 TCP 구현에서 볼 수 있듯이 각각의 상태에서 다음 상태로 넘어가기 전에 필요한 변수들의 확인과 같은 분기제어 기능 수행을 위해 몇 개의 상태가 추가되었다.



(그림 1) PCC를 위한 PCEP FSM

을 감지하게 된다.

PCC나 PCE에서는 또한 DeadTimer를 구동시킨다. DeadTimer는 세션을 통하여 메시지를 받을 때마다 재시작 된다. 만일 세션의 한쪽 peer가 Deadtimer가 경과될 때까지 아무런 메시지도 받지 못한다면, 세션은 죽은 것으로 간주된다.

3.3 FSM 변수

PCEP 구현에서는 (표 2), (표 3)과 같이 몇 개의 변수들을 정의하고 있다.

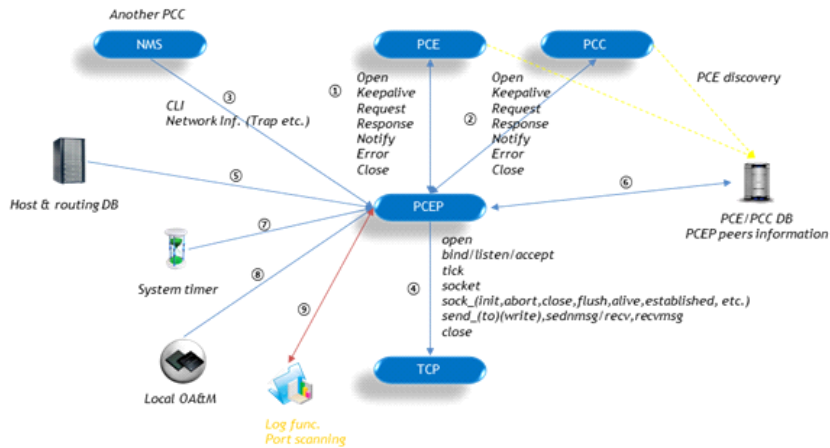
3.4 PCEP 상태와 동작

(그림 1)은 PCEP 구현에서 FSM과 관련된 메시지,

(1) idle 상태

초기화 이벤트를 기다리고 있는 상태이다. 초기화 이벤트는 사용자에게 의해 시작되거나 이벤트에 의해서 자동으로 시작된다. Idle 상태에서는 PCEP 리소스가 할당되지만 아무런 PCEP 메시지를 받지 않은 상태이다. 따라서 등록된 TCP 포트를 listens 하게 된다. Idle 상태에서 TCPRetry, LocalOK, RemoteOK, OpenRetry와 같은 변수들을 초기화 한다. PCEP 세션을 설정하라는 로컬 이벤트를 받게 되면 다음과 같은 일을 수행한다.

PCEP peer와 TCP 연결을 초기화
Connect timer를 시작
TCPPending 상태로 이전



(그림 2) PCEP 외부 인터페이스

만일 TCP 연결 설정이 성공적으로 이루어지게 되면 시스템은 다음과 같은 일을 수행한다.

- Open 메시지 전송
- OpenWait timer 시작
- OpenWait 상태로 이전

TCPPending 상태

만일 TCP 연결설정이 성공적으로 이루어지게 되면 다음과 같은 일을 수행한다.

- Open message 전송
- OpenWait timer 시작
- OpenWait 상태로 이전

만일 TCP 연결 설정이 실패하거나 Connect timer가 expires 된 경우 다음 일을 수행한다.

만일 ConnectRetry = ConnectMaxRetry 이라면, Idle 상태로 이전

만일 ConnectRetry < ConnectMaxRetry 이라면, system은 다음과 같이 동작한다.

- PCEP peer와 TCP 연결을 시작
- ConnectRetry variable 값을 증가
- Connect timer를 Restarts
- TCPPending 상태 에 그대로 머무름

OpenWait 상태

OpenWait 상태에서는 PCEP peer의 Open 메시지

를 기다리고 있게 된다. 만일 OpenWait timer가 expire 되기 전에 PCEP peer로부터 Open 메시지를 수신하면, 우선 OpenWait 또는 KeepWait 상태에 있는 모든 세션을 검사한다. 만일 동일한 peer와 (동일한 IP address) 이미 다른 세션이 존재한다면 collision-resolution 절차를 수행한다.

KeepWait 상태

Keepwait 상태에서는 Open 메시지에 대한 응답인 Keepalive 메시지를 peer로부터 기다리고 있게 된다. 또한 Open 메시지에서 제안했던 특징에 대해 협상이 되지 않아 unacceptable PCEP session characteristics를 의미하는 PCErr 메시지를 수신할 수 있다. 만일 KeepWait timer가 expire 되기 전에 Keepalive 메시지를 수신하였다면 LocalOK=1로 한다.

UP 상태

만일 Keepalive timer가 expires 되면 Keepalive timer를 재시작하고 Keepalive 메시지를 전송한다. 만일 DeadTimer가 expire 되기 전에 어떠한 PCEP 메시지도 받지 못했다면 세션을 종료하고 TCP 연결을 close 하고, PCEP 리소스를 releases 하고 Idle 상태로 돌아간다. 만일 TCP 연결이 실패하면, TCP 연결을 close 하고, PCEP 리소스를 releases 한다. 이후 Idle 상태로 돌아간다.

3.5 PCEP 상태와 동작

(표 4)와 (그림 2)는 PCEP의 인터페이스와 내용을 나타내고 있다.

(표 4) PCEP 인터페이스와 내용

번호	Interface	Interface 내용
①	PCEP-P CE	PCE에게 PCEP 서비스를 제공하는 인터페이스. PCE는 동시에 PCC가 될 수 있다. 대표적으로 다음과 같은 API 제공. ① Open ② Keepalive ③ Request ④ Response ⑤ Notify ⑥ Error ⑦ Close
②	PCEP-P CC	PCC에게 PCEP 서비스를 제공하는 인터페이스. 대표적으로 다음과 같은 API 제공. ① Open ② Keepalive ③ Request ④ Response ⑤ Notify ⑥ Error ⑦ Close
③	PCEP-N MS	NMS에게 PCEP 서비스를 제공하는 인터페이스. NMS는 configuration이나 CLI를 통해서 PCEP에 인터페이스 가능. NMS는 독립적으로 PCE에게 접속할 수 있음.
④	PCEP-T CP	PCC, PCE에게 PCEP 서비스를 제공하기 위해서 하위 TCP 프로토콜 기능을 이용하는 TCP 접속 인터페이스. 기존 Socket 인터페이스를 모두 사용 ① open ② bind/listen/accept ③ tick ④ socket ⑤ sock_(init,abort,close,flush,alive,established, etc.) ⑥ send_(to)(write),sednmsg/recv,recvmmsg) ⑦ close
⑤	PCEP-T ED	protocol error condition 검사를 위해 필요한 정보를 access 하는 인터페이스 System의 TED 정보, Host 및 라우팅 DB 접속 인터페이스를 포함한다.
⑥	PCEP-P CE DB	protocol error condition 검사를 위해 필요한 PCC 및 PCE 호스트 정보를 access 하는 인터페이스. PCE의 dynamic discovery 방법에 관해서는

		관여하지 않으며 시스템은 PCE/PCE host 및 관련된 정보를 PCEP-PCE DB interface를 통해서 PCEP에게 알려주어야 한다.
⑦	PCEP-T imer	필요한 각종 timer를 구동하기 위한 인터페이스. 모든 timer는 초단위로 동작하지만 PCEP implementation 및 기타 상/하위 인터페이스를 제공하기 위해 msec 단위의 타이머 인터페이스를 사용
⑧	PCEP-O A&M	protocol error condition 검사를 위해 필요한 각종 alarm 및 system 정보를 주고 받는 인터페이스. PCEP-TED/PCE DB 인터페이스와 함께 protocol 검사 및 FSM 운용에 사용.
⑨	PCEP-L OG	디버깅이나 각종 정보 저장에 위한 인터페이스. 파일을 기본 저장장치로 사용한다.

4. 결 론

최근 다양한 형태로 발생하는 응용 계층에서의 통합 특성은 고속 IP 장비와 광전송 장비에도 그 특성이 신속하게 반영되고 있으며, 진보된 액세스 기술의 광대역화는 네트워크의 중심이 되는 고속 IP 장비 및 광전송 장비의 고용량화를 이끌고 있다. 이처럼 다양한 환경에서의 PCE 시스템 설계/개발 과정에서 획득한 핵심기술을 분석하여 다계층 광네트워크 제어 플랫폼에 탑재를 하고 이를 통하여 PCC/PCE 메커니즘에 대한 기능 규격 등을 제시하기 위한 연구가 필요하다.

본 논문에서는 PCE 기반 경로계산 구조에 대해서 살펴보고 이를 기반으로 PCEP 설계 및 구현 방안을 제시하였다. 이를 위해 본 논문에서는 PCEP를 구성하고 있는 PCEP의 각종 메시지와 변수들의 용도를 정리하였고, TCP 구현에서 볼 수 있듯이 상태기반 프로토콜 구현을 위해 각각의 상태에서 필요한 동작의 기술과 다음 상태로 넘어가기 전에 필요한 분기제어 기능 수행을 위해 RFC5440에서 제안하고 있는 상태보다 몇 개의 상태가 추가 제안하였다.

마지막으로 본 논문에서는 PCEP에서 제공하는 통신 채널을 효과적으로 제어하기 위한 내부/외부 인터페이스를 더불어 제안하였다. PCEP 구현을 위한 FSM 확장과 인터페이스 제안과 관련된 논문은 아직 없는 상태이다.

참고문헌

- [1] K. Kompella and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)," RFC 4206, Oct. 2005.
- [2] A. Bonerjee, J. Drake, J. P. Lang, and B. Turner, "Generalized Multiprotocol Label Switching: an Overview of Routing and Management Enhancements," IEEE Communications Magazine, Vol. 39, No. 1, pp. 144-150, Jan. 2001.
- [3] K. Shimoto, E. Oki, D. Shimazaki, and T. Miyamura, "Multilayer Traffic Engineering Experiments in MPLS/GMPLS Networks," IEEE BROADNETS, Sep. 2007.
- [4] Richard Douville, Jean-Louis Le Roux, Stefano Secci, "A Service Plane over the PCE Architecture for Automatic Multidomain Connection-Oriented Services", IEEE Communications Magazine, Vol. 46, No. 6, pp. 94-102, Jun. 2008.
- [5] H. Matsuura, N. Morita, T. Murakami, and K. Takami, "Hierarchically Distributed PCE for GMPLS Multilayered Networks," IEEE Globecom, Nov. 2005.
- [6] Cristel Pelsser, Olivier Bonaventure, "Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs," Lecture Notes in Computer Science, Vol. 3976, pp. 209-220, 2006.
- [7] Eiji Oki, Ichiro Inoue, Kohei Shimoto, "Path Computation Element (PCE)-based Traffic Engineering in MPLS and GMPLS networks," IEEE Communications Magazine, 2008.
- [8] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," IETF RFC 4655, Aug. 2006.
- [9] J. Ash and J. L. Le Roux, "PCE Communication Protocol Generic Requirements," IETF RFC 4657, Sep. 2006.
- [10] J. P. Vasseur, J. L. Roux, A. Ayyangar, E. Oki, A. Atlas, and A. Dolganow, "Path Computation Element (PCE) Communication Protocol (PCEP)," IETF RFC 5440, Mar. 2009.

[저자 소개]



이원혁 (Wonhyuk Lee)

2001년 2월 성균관대학교 학사
 2003년 2월 성균관대학교 석사
 2010년 8월 성균관대학교 박사
 2003년 3월 ~ 현재
 한국과학기술정보연구원
 선임연구원

email : livezone@kisti.re.kr



강승애 (Seuungae Kang)

1995년 2월 이화여자대학교 학사
 1997년 8월 이화여자대학교 석사
 2006년 8월 이화여자대학교 박사
 2006년 9월 ~ 현재 남서울대학교
 운동건강학과 교수

email : sahome@nsu.ac.kr



김현철 (Hyuncheol Kim)

1990년 2월 성균관대학교 학사
 1992년 2월 성균관대학교 석사
 2005년 8월 성균관대학교 박사
 2006년 9월 ~ 현재 남서울대학교
 컴퓨터학과 교수

email : hckim@nsu.ac.kr