

Path Computation Element (PCE) Communication Protocol (PCEP)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Abstract

This document specifies the Path Computation Element (PCE) Communication Protocol (PCEP) for communications between a Path Computation Client (PCC) and a PCE, or between two PCEs. Such interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering. PCEP is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

Table of Contents

1. Introduction	5
1.1. Requirements Language	5
2. Terminology	5
3. Assumptions	6
4. Architectural Protocol Overview (Model)	7
4.1. Problem	7
4.2. Architectural Protocol Overview	7
4.2.1. Initialization Phase	8
4.2.2. Session Keepalive	9
4.2.3. Path Computation Request Sent by a PCC to a PCE	10
4.2.4. Path Computation Reply Sent by The PCE to a PCC	11
4.2.5. Notification	12
4.2.6. Error	14
4.2.7. Termination of the PCEP Session	14
4.2.8. Intermittent versus Permanent PCEP Session	15
5. Transport Protocol	15
6. PCEP Messages	15
6.1. Common Header	16
6.2. Open Message	16
6.3. Keepalive Message	18
6.4. Path Computation Request (PCReq) Message	19
6.5. Path Computation Reply (PCRep) Message	20
6.6. Notification (PCNtf) Message	21
6.7. Error (PCErr) Message	22
6.8. Close Message	23
6.9. Reception of Unknown Messages	23
7. Object Formats	23
7.1. PCEP TLV Format	24
7.2. Common Object Header	24
7.3. OPEN Object	25
7.4. RP Object	27
7.4.1. Object Definition	27
7.4.2. Handling of the RP Object	30
7.5. NO-PATH Object	31
7.6. END-POINTS Object	34
7.7. BANDWIDTH Object	35
7.8. METRIC Object	36
7.9. Explicit Route Object	39
7.10. Reported Route Object	39
7.11. LSPA Object	40
7.12. Include Route Object	42
7.13. SVEC Object	42
7.13.1. Notion of Dependent and Synchronized Path Computation Requests	42
7.13.2. SVEC Object	44
7.13.3. Handling of the SVEC Object	45

7.14.	NOTIFICATION Object	46
7.15.	PCEP-ERROR Object	49
7.16.	LOAD-BALANCING Object	54
7.17.	CLOSE Object	55
8.	Manageability Considerations	56
8.1.	Control of Function and Policy	56
8.2.	Information and Data Models	57
8.3.	Liveness Detection and Monitoring	57
8.4.	Verifying Correct Operation	58
8.5.	Requirements on Other Protocols and Functional Components	58
8.6.	Impact on Network Operation	58
9.	IANA Considerations	59
9.1.	TCP Port	59
9.2.	PCEP Messages	59
9.3.	PCEP Object	59
9.4.	PCEP Message Common Header	61
9.5.	Open Object Flag Field	61
9.6.	RP Object	61
9.7.	NO-PATH Object Flag Field	62
9.8.	METRIC Object	63
9.9.	LSPA Object Flag Field	63
9.10.	SVEC Object Flag Field	64
9.11.	NOTIFICATION Object	64
9.12.	PCEP-ERROR Object	65
9.13.	LOAD-BALANCING Object Flag Field	67
9.14.	CLOSE Object	67
9.15.	PCEP TLV Type Indicators	68
9.16.	NO-PATH-VECTOR TLV	68
10.	Security Considerations	69
10.1.	Vulnerability	69
10.2.	TCP Security Techniques	70
10.3.	PCEP Authentication and Integrity	70
10.4.	PCEP Privacy	71
10.5.	Key Configuration and Exchange	71
10.6.	Access Policy	73
10.7.	Protection against Denial-of-Service Attacks	73
10.7.1.	Protection against TCP DoS Attacks	73
10.7.2.	Request Input Shaping/Policing	74
11.	Acknowledgments	75
12.	References	75
12.1.	Normative References	75
12.2.	Informative References	76
Appendix A.	PCEP Finite State Machine (FSM)	79
Appendix B.	PCEP Variables	85
Appendix C.	Contributors	86

1. Introduction

[RFC4655] describes the motivations and architecture for a Path Computation Element (PCE) based model for the computation of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering Label Switched Paths (TE LSPs). The model allows for the separation of PCE from Path Computation Client (PCC), and allows for the cooperation between PCEs. This necessitates a communication protocol between PCC and PCE, and between PCEs. [RFC4657] states the generic requirements for such a protocol including that the same protocol be used between PCC and PCE, and between PCEs. Additional application-specific requirements (for scenarios such as inter-area, inter-AS, etc.) are not included in [RFC4657], but there is a requirement that any solution protocol must be easily extensible to handle other requirements as they are introduced in application-specific requirements documents. Examples of such application-specific requirements are [RFC4927], [RFC5376], and [INTER-LAYER].

This document specifies the Path Computation Element Protocol (PCEP) for communications between a PCC and a PCE, or between two PCEs, in compliance with [RFC4657]. Such interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of MPLS and GMPLS Traffic Engineering.

PCEP is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Terminology

The following terminology is used in this document.

AS: Autonomous System.

Explicit path: Full explicit path from start to destination; made of a list of strict hops where a hop may be an abstract node such as an AS.

IGP area: OSPF area or IS-IS level.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.