

OSPF Version 2

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.

Abstract

This memo documents version 2 of the OSPF protocol. OSPF is a link-state routing protocol. It is designed to be run internal to a single Autonomous System. Each OSPF router maintains an identical database describing the Autonomous System's topology. From this database, a routing table is calculated by constructing a shortest-path tree.

OSPF recalculates routes quickly in the face of topological changes, utilizing a minimum of routing protocol traffic. OSPF provides support for equal-cost multipath. An area routing capability is provided, enabling an additional level of routing protection and a reduction in routing protocol traffic. In addition, all OSPF routing protocol exchanges are authenticated.

The differences between this memo and [RFC 1583](#) are explained in [Appendix G](#). All differences are backward-compatible in nature. Implementations of this memo and of [RFC 1583](#) will interoperate.

Please send comments to ospf@gated.cornell.edu.

Table of Contents

1	Introduction	5
1.1	Protocol Overview	5
1.2	Definitions of commonly used terms	6
1.3	Brief history of link-state routing technology	9
1.4	Organization of this document	10
1.5	Acknowledgments	11
2	The link-state database: organization and calculations	11
2.1	Representation of routers and networks	11

2.1.1	Representation of non-broadcast networks	13
2.1.2	An example link-state database	14
2.2	The shortest-path tree	18
2.3	Use of external routing information	20
2.4	Equal-cost multipath	22
3	Splitting the AS into Areas	22
3.1	The backbone of the Autonomous System	23
3.2	Inter-area routing	23
3.3	Classification of routers	24
3.4	A sample area configuration	25
3.5	IP subnetting support	31
3.6	Supporting stub areas	32
3.7	Partitions of areas	33
4	Functional Summary	34
4.1	Inter-area routing	35
4.2	AS external routes	35
4.3	Routing protocol packets	35
4.4	Basic implementation requirements	38
4.5	Optional OSPF capabilities	39
5	Protocol data structures	40
6	The Area Data Structure	42
7	Bringing Up Adjacencies	44
7.1	The Hello Protocol	44
7.2	The Synchronization of Databases	45
7.3	The Designated Router	46
7.4	The Backup Designated Router	47
7.5	The graph of adjacencies	48
8	Protocol Packet Processing	49
8.1	Sending protocol packets	49
8.2	Receiving protocol packets	51
9	The Interface Data Structure	54
9.1	Interface states	57
9.2	Events causing interface state changes	59
9.3	The Interface state machine	61
9.4	Electing the Designated Router	64
9.5	Sending Hello packets	66
9.5.1	Sending Hello packets on NBMA networks	67
10	The Neighbor Data Structure	68
10.1	Neighbor states	70
10.2	Events causing neighbor state changes	75
10.3	The Neighbor state machine	76
10.4	Whether to come adjacent	82
10.5	Receiving Hello Packets	83
10.6	Receiving Database Description Packets	85
10.7	Receiving Link State Request Packets	88
10.8	Sending Database Description Packets	89
10.9	Sending Link State Request Packets	90
10.10	An Example	91

11	The Routing Table Structure	93
11.1	Routing table lookup	96
11.2	Sample routing table, without areas	97
11.3	Sample routing table, with areas	97
12	Link State Advertisements (LSAs)	100
12.1	The LSA Header	100
12.1.1	LS age	101
12.1.2	Options	101
12.1.3	LS type	102
12.1.4	Link State ID	102
12.1.5	Advertising Router	104
12.1.6	LS sequence number	104
12.1.7	LS checksum	105
12.2	The link state database	105
12.3	Representation of TOS	106
12.4	Originating LSAs	107
12.4.1	Router-LSAs	110
12.4.1.1	Describing point-to-point interfaces	112
12.4.1.2	Describing broadcast and NBMA interfaces	113
12.4.1.3	Describing virtual links	113
12.4.1.4	Describing Point-to-MultiPoint interfaces	114
12.4.1.5	Examples of router-LSAs	114
12.4.2	Network-LSAs	116
12.4.2.1	Examples of network-LSAs	116
12.4.3	Summary-LSAs	117
12.4.3.1	Originating summary-LSAs into stub areas	119
12.4.3.2	Examples of summary-LSAs	119
12.4.4	AS-external-LSAs	120
12.4.4.1	Examples of AS-external-LSAs	121
13	The Flooding Procedure	122
13.1	Determining which LSA is newer	126
13.2	Installing LSAs in the database	127
13.3	Next step in the flooding procedure	128
13.4	Receiving self-originated LSAs	130
13.5	Sending Link State Acknowledgment packets	131
13.6	Retransmitting LSAs	133
13.7	Receiving link state acknowledgments	134
14	Aging The Link State Database	134
14.1	Premature aging of LSAs	135
15	Virtual Links	135
16	Calculation of the routing table	137
16.1	Calculating the shortest-path tree for an area	138
16.1.1	The next hop calculation	144
16.2	Calculating the inter-area routes	145
16.3	Examining transit areas' summary-LSAs	146
16.4	Calculating AS external routes	149
16.4.1	External path preferences	151
16.5	Incremental updates -- summary-LSAs	151

16.6	Incremental updates -- AS-external-LSAs	152
16.7	Events generated as a result of routing table changes	153
16.8	Equal-cost multipath	154
	Footnotes	155
	References	158
A	OSPF data formats	160
A.1	Encapsulation of OSPF packets	160
A.2	The Options field	162
A.3	OSPF Packet Formats	163
A.3.1	The OSPF packet header	164
A.3.2	The Hello packet	166
A.3.3	The Database Description packet	168
A.3.4	The Link State Request packet	170
A.3.5	The Link State Update packet	171
A.3.6	The Link State Acknowledgment packet	172
A.4	LSA formats	173
A.4.1	The LSA header	174
A.4.2	Router-LSAs	176
A.4.3	Network-LSAs	179
A.4.4	Summary-LSAs	180
A.4.5	AS-external-LSAs	182
B	Architectural Constants	184
C	Configurable Constants	186
C.1	Global parameters	186
C.2	Area parameters	187
C.3	Router interface parameters	188
C.4	Virtual link parameters	190
C.5	NBMA network parameters	191
C.6	Point-to-MultiPoint network parameters	191
C.7	Host route parameters	192
D	Authentication	193
D.1	Null authentication	193
D.2	Simple password authentication	193
D.3	Cryptographic authentication	194
D.4	Message generation	196
D.4.1	Generating Null authentication	196
D.4.2	Generating Simple password authentication	197
D.4.3	Generating Cryptographic authentication	197
D.5	Message verification	198
D.5.1	Verifying Null authentication	199
D.5.2	Verifying Simple password authentication	199
D.5.3	Verifying Cryptographic authentication	199
E	An algorithm for assigning Link State IDs	201
F	Multiple interfaces to the same network/subnet	203
G	Differences from RFC 1583	204
G.1	Enhancements to OSPF authentication	204
G.2	Addition of Point-to-MultiPoint interface	204
G.3	Support for overlapping area ranges	205

G.4	A modification to the flooding algorithm	206
G.5	Introduction of the MinLSArrival constant	206
G.6	Optionally advertising point-to-point links as subnets	207
G.7	Advertising same external route from multiple areas ..	207
G.8	Retransmission of initial Database Description packets	209
G.9	Detecting interface MTU mismatches	209
G.10	Deleting the TOS routing option	209
	Security Considerations	210
	Author's Address	211

1. Introduction

This document is a specification of the Open Shortest Path First (OSPF) TCP/IP internet routing protocol. OSPF is classified as an Interior Gateway Protocol (IGP). This means that it distributes routing information between routers belonging to a single Autonomous System. The OSPF protocol is based on link-state or SPF technology. This is a departure from the Bellman-Ford base used by traditional TCP/IP internet routing protocols.

The OSPF protocol was developed by the OSPF working group of the Internet Engineering Task Force. It has been designed expressly for the TCP/IP internet environment, including explicit support for CIDR and the tagging of externally-derived routing information. OSPF also provides for the authentication of routing updates, and utilizes IP multicast when sending/receiving the updates. In addition, much work has been done to produce a protocol that responds quickly to topology changes, yet involves small amounts of routing protocol traffic.

1.1. Protocol overview

OSPF routes IP packets based solely on the destination IP address found in the IP packet header. IP packets are routed "as is" -- they are not encapsulated in any further protocol headers as they transit the Autonomous System. OSPF is a dynamic routing protocol. It quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence. This period of convergence is short and involves a minimum of routing traffic.

In a link-state routing protocol, each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each participating router has an identical database. Each individual piece of this database is a particular router's local state (e.g., the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the Autonomous System by flooding.

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.