

Network Working Group
Request for Comments: 2328
STD: 54
Obsoletes: 2178
Category: Standards Track

J. Moy
Ascend Communications, Inc.
April 1998

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.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

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 2178 are explained in Appendix G. All differences are backward-compatible in nature.

Moy

Standards Track

[Page 1]

Implementations of this memo and of RFCs 2178, 1583, and 1247 will interoperate.

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

Table of Contents

1	Introduction	6
1.1	Protocol Overview	6
1.2	Definitions of commonly used terms	8
1.3	Brief history of link-state routing technology	11
1.4	Organization of this document	12
1.5	Acknowledgments	12
2	The link-state database: organization and calculations	13
2.1	Representation of routers and networks	13
2.1.1	Representation of non-broadcast networks	15
2.1.2	An example link-state database	18
2.2	The shortest-path tree	21
2.3	Use of external routing information	23
2.4	Equal-cost multipath	26
3	Splitting the AS into Areas	26
3.1	The backbone of the Autonomous System	27
3.2	Inter-area routing	27
3.3	Classification of routers	28
3.4	A sample area configuration	29
3.5	IP subnetting support	35
3.6	Supporting stub areas	37
3.7	Partitions of areas	38
4	Functional Summary	40
4.1	Inter-area routing	41
4.2	AS external routes	41
4.3	Routing protocol packets	42
4.4	Basic implementation requirements	43
4.5	Optional OSPF capabilities	46
5	Protocol data structures	47
6	The Area Data Structure	49
7	Bringing Up Adjacencies	52
7.1	The Hello Protocol	52
7.2	The Synchronization of Databases	53
7.3	The Designated Router	54
7.4	The Backup Designated Router	56
7.5	The graph of adjacencies	56

8	Protocol Packet Processing	58
8.1	Sending protocol packets	58
8.2	Receiving protocol packets	61
9	The Interface Data Structure	63
9.1	Interface states	67
9.2	Events causing interface state changes	70
9.3	The Interface state machine	72
9.4	Electing the Designated Router	75
9.5	Sending Hello packets	77
9.5.1	Sending Hello packets on NBMA networks	79
10	The Neighbor Data Structure	80
10.1	Neighbor states	83
10.2	Events causing neighbor state changes	87
10.3	The Neighbor state machine	89
10.4	Whether to become adjacent	95
10.5	Receiving Hello Packets	96
10.6	Receiving Database Description Packets	99
10.7	Receiving Link State Request Packets	102
10.8	Sending Database Description Packets	103
10.9	Sending Link State Request Packets	104
10.10	An Example	105
11	The Routing Table Structure	107
11.1	Routing table lookup	111
11.2	Sample routing table, without areas	111
11.3	Sample routing table, with areas	112
12	Link State Advertisements (LSAs)	115
12.1	The LSA Header	116
12.1.1	LS age	116
12.1.2	Options	117
12.1.3	LS type	117
12.1.4	Link State ID	117
12.1.5	Advertising Router	119
12.1.6	LS sequence number	120
12.1.7	LS checksum	121
12.2	The link state database	121
12.3	Representation of TOS	122
12.4	Originating LSAs	123
12.4.1	Router-LSAs	126
12.4.1.1	Describing point-to-point interfaces	130
12.4.1.2	Describing broadcast and NBMA interfaces	130
12.4.1.3	Describing virtual links	131
12.4.1.4	Describing Point-to-MultiPoint interfaces	131

12.4.1.5	Examples of router-LSAs	132
12.4.2	Network-LSAs	133
12.4.2.1	Examples of network-LSAs	134
12.4.3	Summary-LSAs	135
12.4.3.1	Originating summary-LSAs into stub areas	137
12.4.3.2	Examples of summary-LSAs	138
12.4.4	AS-external-LSAs	139
12.4.4.1	Examples of AS-external-LSAs	140
13	The Flooding Procedure	143
13.1	Determining which LSA is newer	146
13.2	Installing LSAs in the database	147
13.3	Next step in the flooding procedure	148
13.4	Receiving self-originated LSAs	151
13.5	Sending Link State Acknowledgment packets	152
13.6	Retransmitting LSAs	154
13.7	Receiving link state acknowledgments	155
14	Aging The Link State Database	156
14.1	Premature aging of LSAs	157
15	Virtual Links	158
16	Calculation of the routing table	160
16.1	Calculating the shortest-path tree for an area	161
16.1.1	The next hop calculation	167
16.2	Calculating the inter-area routes	178
16.3	Examining transit areas' summary-LSAs	170
16.4	Calculating AS external routes	173
16.4.1	External path preferences	175
16.5	Incremental updates -- summary-LSAs	175
16.6	Incremental updates -- AS-external-LSAs	177
16.7	Events generated as a result of routing table changes	177
16.8	Equal-cost multipath	178
	Footnotes	179
	References	183
A	OSPF data formats	185
A.1	Encapsulation of OSPF packets	185
A.2	The Options field	187
A.3	OSPF Packet Formats	189
A.3.1	The OSPF packet header	190
A.3.2	The Hello packet	193
A.3.3	The Database Description packet	195
A.3.4	The Link State Request packet	197
A.3.5	The Link State Update packet	199
A.3.6	The Link State Acknowledgment packet	201

A.4	LSA formats	203
A.4.1	The LSA header	204
A.4.2	Router-LSAs	206
A.4.3	Network-LSAs	210
A.4.4	Summary-LSAs	212
A.4.5	AS-external-LSAs	214
B	Architectural Constants	217
C	Configurable Constants	219
C.1	Global parameters	219
C.2	Area parameters	220
C.3	Router interface parameters	221
C.4	Virtual link parameters	224
C.5	NBMA network parameters	224
C.6	Point-to-MultiPoint network parameters	225
C.7	Host route parameters	226
D	Authentication	227
D.1	Null authentication	227
D.2	Simple password authentication	228
D.3	Cryptographic authentication	228
D.4	Message generation	231
D.4.1	Generating Null authentication	231
D.4.2	Generating Simple password authentication	232
D.4.3	Generating Cryptographic authentication	232
D.5	Message verification	234
D.5.1	Verifying Null authentication	234
D.5.2	Verifying Simple password authentication	234
D.5.3	Verifying Cryptographic authentication	235
E	An algorithm for assigning Link State IDs	236
F	Multiple interfaces to the same network/subnet	239
G	Differences from RFC 2178	240
G.1	Flooding modifications	240
G.2	Changes to external path preferences	241
G.3	Incomplete resolution of virtual next hops	241
G.4	Routing table lookup	241
	Security Considerations	243
	Author's Address	243
	Full Copyright Statement	244

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.