Pee-Wee OSPF Protocol Details

Protocol Overview

PWOSPF is a greatly simplified link state routing protocol based on OSPFv2. Like OSPFv2, routers participating in a PWOSPF topology periodically broadcast HELLO packets to discover and maintain a list of neighbors. Whenever a change in a link status is detected (for example the addition or deletion of a router to the topology) or a timeout occurs, each router floods its view of the network throughout the topology so that each router has a complete database of the network connectivity. Djikstra's algorithm is used by each router independently to determine the next hop in the forwarding table to all advertised routes.

Data Structures

PWOSPF Router

Like OSPF, PWOSPF operates within an "area" of routers, defined by a 32 bit value. A router can only participate in one area at a time. Each router in an area must have a unique 32 bit router ID. By convention, the IP address of the 0th interface is used as the router ID. 0 and 0xffffffff are invalid router IDs and can be used internally to mark uninitialized router ID fields.

Each router must therefore define the following values:

- 32 bit router ID
- 32 bit area ID
- 16 bit lsuint - interval in seconds between link state update broadcasts
- List of router interfaces

PWOSPF Interface

The interface is a key abstraction in PWOSPF for logically decomposing the topology. Interfaces between neighboring routers are connected by links which must have an associated subnet number and mask. All links are assumed to be bi-directional. Unlike OSPF, PWOSPF requires that each interface only have one neighbor.

An interface within a PWOSPF router is defined by the following values:

- 32 bit ip address - IP address of interface
- 32 bit mask - subnet mask of interface
- 8 bit helloint - interval in seconds between HELLO broadcasts
- 32 bit neighbor id - ID of neighboring router for this interface if any.
- 32 bit neighbor ip - IP address of neighboring router's interface this interface is directly connected to.

PWOSPF Hello Protocol

To discover and maintain the state of available links, a router participating in a PWOSPF topology periodically listens for and broadcasts HELLO packets. HELLO packets are broadcasted every helloint seconds with a destination address of ALLSPFRouters, that is defined as "224.0.0.5" (0xe0000005). This implies that all participating routers must be configured to receive and process packets sent to ALLSPFRouters. On receipt of a HELLO packet, a router may do one of three things. If the packet is invalid or corrupt the router will drop and ignore the packet and optionally log the error. If the packet is from a yet-to-be-identified neighbor and no other neighbors have been discovered off of the incoming interface, the router will add the neighbor to the interface. If the packet is from a known neighbor, the router will mark the time the packet was received so it can track the uptime of this neighbor. The set of links of “routers to neighbors” provides the basic connectivity information for the full topology.

PWOSPF routers use HELLO packets to monitor the status of a neighboring router. If a neighboring router does not emit a HELLO packet within NEIGHBOR_TIMEOUT seconds (default value is 20) of the last HELLO received, the router is assumed down, removed from the interface and a link state update flood is initiated. Note that only HELLO packets are used to determine link status. Even in the case where the router is actively routing packets and generating link state update packets, if no HELLO packets are generated, it will be considered disconnected from the topology.

PWOSPF Link State Updates

Global network connectivity is obtained by each router through link state updates in which local link connectivity information is flooded throughout the area by each router. Link state updates are sent periodically every LSUINT seconds (default value of 30) *and* whenever a change in link status is detected. If a link state change initiates a link state update, the lsuint counter is reset to wait another LSUINT seconds before triggering another flood.

The link state advertisements generated by each router list the subnet numbers of each of the router's interfaces and all neighboring routers. Link state updates operate via a simple sequenced, unacknowledged (unlike OSPF) flooding scheme in which received packets are flooded to all neighbors except the neighbor from which the packet was received. Generated packets are flooded to all neighbors. LSU packets are used to build and maintain the network topology database at each router. If the LSU packet does not advertise a change in the state of the topology as is already reflected in the database, it is discarded and the sequence number is updated. Otherwise, the information is used to update the database and the router's forwarding tables are recalculated using Dijkstra's algorithm.

A gateway router may advertise an additional default subnet number for an interface that is connected to a separate network. In the typical case, this interface will be the network’s link to the Internet and will advertise a default subnet number of 0.0.0.0. All traffic not destined to a subnet number on the PWOSPF network will be routed to this interface as a gateway to the Internet.

The Topology Database

Every router in a PWOSPF area maintains a full representation of the area network topology. This topology database is used to calculate the next hop for each destination in the network. A typical implementation of the topology database will contain an adjacency list of all the routers in the network as well as the subnet numbers/masks associated with each link. Dijkstra's algorithm is used on the adjacency list to determine the best next hop for each router. The forwarding table is then built using the advertised routes from each router and the next hop to those routers as determined by Dijkstra.

If there are discrepancies in advertisements from two different routers about the same link, the link is assumed invalid and not added to the database.

This may happen in the following cases:

- A advertises that it is connected to a subnet with mask 255.255.255.0 and neighbor B. But B does not advertise that A is a neighbor.
- A advertises that it is connected to a subnet with mask 255.255.255.0 and neighbor B. But B advertises it is connected to a subnet with mask 255.255.255.240 with neighbor A.

In both of these cases the link should not be added to the advertised database.

Each entry in the database is time-stamped with the last time an LSU for the associated router was received. If an LSU is not received from the router within LSU_TIMOUT seconds (default is 35) from the last, the entry is invalidated and removed from the database.

Handling Incoming PWOSPF Packets

Each router participating in a PWOSPF topology must check the following values on incoming PWOSPF packets:

- The version number field must specify protocol version 2.
- The 16-bit checksum on the PWOSPF packet's contents must be verified. (the 64-bit authentication field must be excluded from the checksum calculation)
- The area ID found in the PWOSPF header must match the area ID of the receiving router.
- The Authentication type specified must match the authentication type of the receiving router.

Note that PWOSPF does not support authentication, however it is kept for compatibility towards OSPFv2. For this reason, PWOSPF uses the full OSPFv2 header format which contains both an authentication type and data field. These fields should be set to 0 for all valid PWOSPF packets.

Handling Incoming HELLO Packets

This section explains the detailed processing of a received Hello packet. The generic input processing of PWOSPF packets will have checked the validity of the IP header and the PWOSPF packet header. Next, the values of the network mask and hellointfields in the received Hello packet must be checked against the values configured for the receiving interface. Any mismatch causes processing to stop and the packet to be dropped. In other words, the above fields should really describe the attached network's configuration.

At this point, an attempt is made to match the source of the Hello packet to one of the receiving interface's neighbors. If the receiving interface is a multi-access network (either broadcast or non-broadcast) the source is identified by the IP source address found in the Hello's IP header. The interface's current neighbor is contained in the interface's data structure. If the interface does not have a neighbor, a neighbor is created. If the interface already has a neighbor which does not match the IP of the incoming packet, the packet is ignored and optionally an error message is generated. Finally, if the Hello packet matches the current neighbor, the neighbor's "last hello packet received" timer is updated.

Handling Incoming LSU Packets

Each received LSU packet must go through the following handling procedure. If the LSU was originally generated by the router itself, the packet is dropped. If the sequence number matches that of the last packet received from the sending router, the packet is dropped. If the packet contents are equivalent to the contents of the packet last received from the sending router, the database entry is updated and the packet is ignored. If the LSU is from a router not currently in the database, the packet’s contents are used to update the database and Dijkstra's algorithm is used to recompute the forwarding table. Finally, if the LSU data is for a router currently in the database but the information has changed, the LSU is used to update the database, and Dijkstra's algorithm is run to recompute the forwarding table.

All received packets with new sequence numbers are flooded to all neighbors except towards the neighbor that sent the packet. The TTL header is only checked in the forwarding stage and should not be considered when handling the packet locally. The TTL field of all flooded packets must be decremented before exiting the router. If the field after decrement is 1 or zero, the packet must not be flooded.

PWOSPF IP Packets

PWOSPF packets are expected to be encapsulated in IPv4 packets with IP protocol number 89 (the same as OSPFv2). OSPF HELLO packets are sent to destination IP address ALLSPFRouters which is "224.0.0.5" (0xe0000005). All LSU packets are sent point to point using the IP address of the neighboring interface as the destination.

PWOSPF Packet Header Format

All PWOSPF packets are encapsulated in a common header that is identical to the OSPFv2 header. Using the OSPFv2 header will allow PWOSPF to converge on OSPF compliance in the future and is recognized by protocol analyzers such as wireshark which can greatly aid in debugging. The PWOSPF header is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |     Type      |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             Autype            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Version #

The PWOSPF/OSPF version number. This specification documents version 2 of the protocol.

Type

The OSPF packet types are as follows. The format of each of these packet types is described in a later section.

Type	Description
1	Hello
4	Link State Update

Packet length

The length of the protocol packet in bytes. This length includes the standard OSPF header.

Router ID

The Router ID of the packet's source. In OSPF, the source and destination of a routing protocol packet are the two ends of an (potential) adjacency.

Area ID

A 32 bit number identifying the area that this packet belongs to. All OSPF packets are associated with a single area. Most travel a single hop only.

Checksum

The standard IP checksum of the entire contents of the packet, excluding the 64-bit authentication field. This checksum is calculated as the 16-bit one's complement of the one's complement sum of all the 16-bit words in the packet, excepting the authentication field. If the packet's length is not an integral number of 16-bit words, the packet is padded with a byte of zero before checksumming.

AuType

Set to zero in PWOSPF

Authentication

Set to zero in PWOSPF

HELLO Packet Format

Hello packets are PWOSPF packet type 1. These packets are sent periodically on all interfaces in order to establish and maintain neighbor relationships.

In addition, Hello broadcasts enable dynamic discovery of neighboring routers.

All routers connected to a common network must agree on certain parameters (network mask and helloint). These parameters are included in Hello packets, so that differences can inhibit the forming of neighbor relationships. A full HELLO packet with PWOSPF header is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       1       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             Autype            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Network Mask                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         HelloInt              |    padding (All Zeros)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Network mask

The network mask associated with this interface. For example, if the interface is to a class B network whose third byte is used for subnetting, the network mask is 0xffffff00.

HelloInt

The number of seconds between this router's Hello packets.

LSU Packet Format

LSU packets implement the flooding of link states and are used to build and maintain the network topology database at each router. Each link state update packet carries a collection of link state advertisements on hop further from its origin. Several link state advertisements may be included in a single packet. A link state packet with full PWOSF header looks as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |       4       |         Packet length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             Autype            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Sequence                |          TTL                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      # advertisements                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-                                                            +-+
|                  Link state advertisements                    |
+-                                                            +-+
|                              ...                              |
```

Sequence

Unique sequence number associated with each Link State Update. Incremented by the LSU source for each subsequent update. Duplicate LSU packets (same or old sequence number) are dropped by the receiver.

TTL

Hop limited value decremented each time the packet is flooded. The TTL value is only considered during packet forwarding and not during packet reception.

advertisements

Total number of link state advertisements contained in the packet

Link state advertisements

Each link state update packet should contain one or more link state advertisements. The advertisements are the reachable routes directly connected to the advertising router. Routes are in the form of the subnet number, mask, router neighbor for the attached link. Link state advertisements look specifically as follows:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Subnet Number                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Mask                                |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         Router ID                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Subnet

Subnet number of the advertised route. Note that default routes will have a subnet value of 0.0.0.0.

Mask

Subnet mask of the advertised route

Router ID

ID of the *neighboring router* on the advertised link. If there is no connected router to the link the Router ID should be set to 0.