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 or a link), each router floods its local connectivity throughout the network so that each router has a complete database of the entire network topology. Each router independently computes the nexthop in the forwarding table to all advertised networks.

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. (In this project use Area ID = 0 for all routers.)

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 updates, default 30s.
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 and mask. All links are assumed to be bi-directional.

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

```
  32 bit ip address  - IP address of associated interface
  32 bit mask        - subnet mask of associated interface
  16 bit helloint    - interval in seconds between HELLO broadcasts, default 10s
  list [
  32 bit neighbor id - ID of neighboring router.
  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). At the Ethernet layer, use the broadcast address 0xffffffff. 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. If the packet is from a yet to be identified neighbor, 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 to track the uptime of its 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 (three times the neighbor's HelloInt) 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 links state update, the lsuint counter is reset to wait another LSUINT seconds before triggering another flood.

The link state advertisements generated by each router lists the subnets of each of the router's interfaces and all neighboring routers. Link state updates operate via a simple sequenced, unacknowledged flooding scheme in which received packets are flooded to all neighbors except the neighbor from whom the packet was received. Generated packets are flooded to all neighbors (they should be addressed directly to each neighbor - i.e., do not send them to the special ALLSPFRouters address). 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, the LSU update should be discarded and the sequence number is updated. Otherwise, the information is used to update the database and the router's forwarding tables are recalculated.

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

The Topology Database

Every router in a PWOSPF area maintains a full representation of the network topology. This topology database is used to calculate the next hop for each destination prefix 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 subnets associated with each link. The forwarding table, which contains the shortest-path nexthop to every destination subnet, is then computed using the topology database.

If there are discrepancies in advertisements from two different nodes 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 subnet with mask 255.255.255.0 and neighbor B. B does not advertise that A is a neighbor.

- A advertises that it is connected to subnet with mask 255.255.255.0 and neighbor B. 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 within LSU_TIMEOUT seconds (three times LSUINT) from the last, the entry is invalidated and removed from the database.

Handling Incoming PWOSPF Packets

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

- o The version number field must specify protocol version 2.
- o 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)
- o The area ID found in the PWOSPF header must match the Area ID of the receiving router.
- o The Authentication type specified must match the authentication type of the receiving router.

PWOSPF does not support authentication, however it is our plan to progress towards OSPFv2 compatibility. For this reason, we are using the full OSPFv2 header format which contains both an Authtication type and data field. These fields should be set to 0 for all valid PWOSPF packets.

Handling Incoming LSU Packets

Each received LSU packet must go through the following handling procedure. If the LSU was originally generated by the receiving router, the packet is dropped. If the sequence number equals or less than that of the last packet received from the originating router, the packet is dropped. If the packet contents are equivalent to the contents of the packet last received from the originating router, the database entry is updated with the new sequence number. If the LSU is from a router not currently in the database, the packets contents are used to update the database and 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 forwarding table is recomputed.

All received packets with newer sequence numbers are flooded to all neighbors but the incoming neighbor of 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 zero or less, the packet must not be flooded.

PWOSPF IP Packets

PWOSPF are expected to be encapsulated IPv4 packets with IP protocol number 89 (the same as OSPFv2). OSPF HELLO packets are sent to destination IP address ALLSPFRouters which is defined as "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 ethereal/tcpdump/wireshark which can greatly aid in debugging. The 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 #   |     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 succeeding 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, i.e., its eth0's IP address. In OSPF, the source and destination of a routing protocol packet are the two ends of an 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, Hellos broadcast enabling dynamic discovery of neighboring routers.

All routers connected to a common network must agree on certain parameters (network mask and hellohint). 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Network mask
    The network mask associated with this interface.

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. Multiple link state advertisements may be included in a single LSU 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 are dropped by the receiver.

TTL
   Hop limited value decremented each time the packet is forwarded. The

TTL value is only considered during packet forwarding and not during packet reception.

\# of advertisements
Total number of link state advertisements contained in the packet

Link state advertisements

Each link state update packet should contain 1 or more link state advertisements. The advertisements are the reachable routes directly connected to the originating router. Routes are in the form of the subnet, mask and router neighbor for the attached link. Link state advertisements look specifically as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Subnet                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Mask                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

subnet
Subnet number of the advertised route.

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 RID should be set to 0.

Example:

In the below topology with subnet 192.168.128 using IP addresses allocated as showing (xxx is intended to be 192.168.128).

```
            xxx.1      xxx.2     xxx.4      xxx.5   xxx.8  xxx.9
[Internet]-[FW]---------------- A ------------------ B ------- <endhost>
```

Assuming FW is not participating in the PWOSPF area.

A could advertise the following routes

```
1. (subnet between A and FW)
   Subnet 192.168.128.0
   Mask   255.255.255.252
   RID    0

2. (default route to the Internet)
   Subnet 0.0.0.0
   Mask   0.0.0.0
   RID    0.0.0.0

3. (link shared with B
   Subnet 192.168.128.4
   Mask   255.255.255.254
   RID    192.168.128.5  (B's router ID)
```

B could advertise the following routes

```
1. (link shared with A)
    Subnet 192.168.128.4
    Mask   255.255.255.254
    RID    192.168.128.4  (A's router ID)

2. (Link to end host)
```

```
Subnet 192.168.128.8
Mask   255.255.255.254
RID    0.0.0.0 (no attached PWOSPF router)
```

Workflow Summary

When a router first starts up, it learns about its interfaces' IPs and masks, which also represent the subnets of the links.

The router reads static routes from rtable file. In this project, only vhost1 will have one static route pointing to the Internet, while vhost2 and vhost3 will not have any static route.

The router sends HELLO messages out from all its interfaces. The router also learns its neighbor routers (i.e., their RIDs) from received HELLO messages.

If no HELLO message is received from a particular interface, the router thinks that there's no PWOSPF router across that link, thus it will use RID=0 for that non-PWOSPF neighbor. This applies to when the neighbor is an app server, when the neighbor router hasn't been up yet, or it’s the default route.

The router generates and floods its LSU. Each LSU contains announcements of *all* its directly connected subnets. In our topology, there should be 3 announcements per LSU, each corresponds to one of the direct links of a router.

Based on the LSUs, each router should be able to build a complete topology database. Then compute shortest paths to all subnets in the topology.

Whenever there is a change in a router's direct links, new LSU must be flooded. E.g., when a neighbor router starts up, the local router should receive a HELLO message, realize that the neighbor RID should change from 0 to the one contained in HELLO, and then flood new LSU to reflect this change. When a neighbor router/interface goes down, the local router will detect it due to missing HELLO message, change the neighbor's RID back to 0, and flood the new LSU.

Since each LSU contains all links that the origin router has, there is no explicit route withdrawal when a link fails.