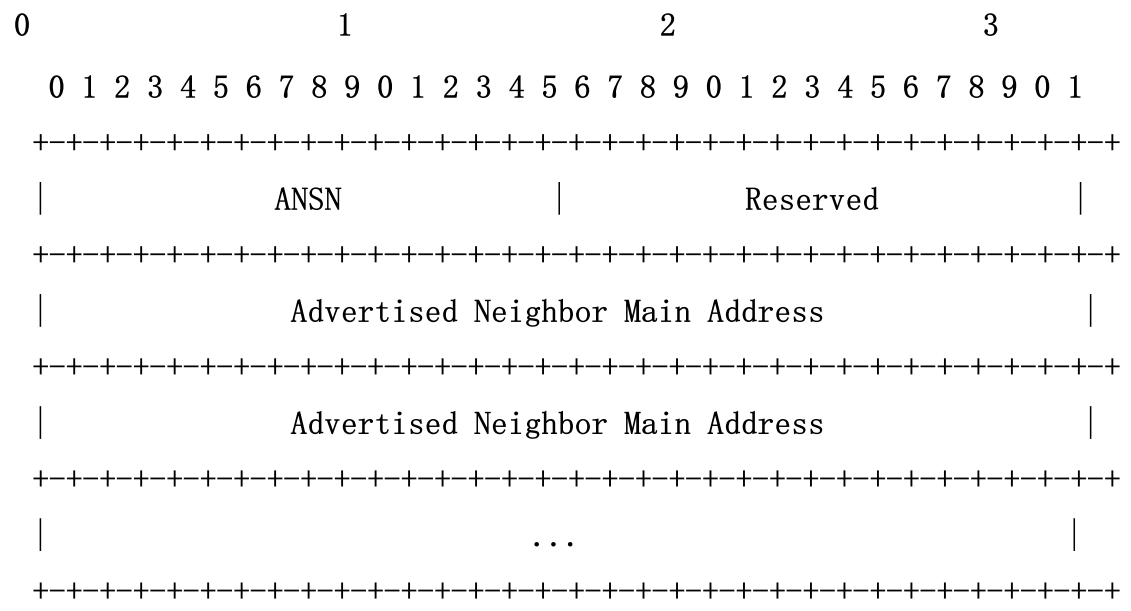
9.拓扑发现

• 该协议的链路检测和邻居检测部分基本上为每个节点提供了可以与其直接通信的邻居列表，并与数据包格式和转发部分结合，通过MPR优化了泛洪机制。基于此，通过网络传播拓扑信息。

• 本节描述了由链路检测和邻居检测给出的信息的哪一部分被散布到整个网络，以及如何将其用于构造路由。路由是通过广告链接和与邻居的链接构建的。为了提供足够的信息以启用路由，节点必须至少在其自身与MPR选择器集中的节点之间传播链接。

9.1 TC消息格式

TC消息的建议格式如下：



• 生存时间应设置为255（最大值），以将消息传播到整个网络，并且将Vtime设置TOP_HOLD_TIME的值。

广告邻居序号（ANSN）

·节点每次在其通告的邻居集中检测到更改时，都会递增此序列号。该编号在TC消息的此ANSN字段中发送，以跟踪最新信息。当节点接收到TC消息时，它可以基于此通告邻居序列号来确定所接收的有关始发者节点的通告邻居的信息是否比其已经拥有的信息更新。

广告邻居主地址

·该字段包含邻居节点的主地址。发起方节点的已通告邻居的所有主要地址都放入TC消息中。如果仍然有尚未通告的邻居地址尚未插入TC消息时达到了最大允许消息大小（由网络施加），则将生成更多TC消息，直到发送了所有通告的邻居集。如果需要冗余，则可以包括邻居节点的额外主地址。

预留字段

·该字段是保留字段，必须设置为“ 0000000000000000”

9.2 播发相邻组

·网络中的节点发送TC消息，以声明一组链接，称为广播链接集，该链接集必须至少包括指向其MPR选择器集的所有节点的链接，即已选择发送者节点作为邻居的链接，简称MPR。

·与通告的邻居集关联的序列号（ANSN）也与列表一起发送。当从发布的邻居集中删除链接时，必须增加ANSN编号；当链接添加到通告的邻居集时，应该增加ANSN号。

9.3 TC消息生成

·为了构建拓扑信息库，已选择为MPR的每个节点都广播拓扑控制（TC）消息。 TC消息被泛洪到网络中的所有节点，并利用MPR。 MPR使拓扑信息的分布具有更好的可伸缩性[1]。

• 地址列表可以是每个TC消息中的一部分（例如，由于网络施加的消息大小限制），但是描述节点的广告链接集的所有TC消息的解析必须在一定的刷新周期内完成（TC_INTERVAL ）。这些TC消息在网络中传播的信息将帮助每个节点计算其路由表。

• 当节点的广告链接集变为空时，该节点仍应在等于其先前发出的TC消息的“有效时间”（通常等于TOP_HOLD_TIME）的持续时间内发送（空）TC消息。 ，以使之前的TC消息无效。然后应该停止发送TC消息，直到某个节点插入其通告的链接集中。

• 一个节点可以发送附加的TC消息，以增加其对链路故障的反应性。当检测到对MPR选择器集的更改，并且该更改可归因于链路故障时，应在比TC_INTERVAL短的间隔后发送TC消息。

9.4 TC邮件转发

• TC消息由MPR广播和重新发送，以便在整个网络中传播消息。 TC消息必须根据“默认转发算法”进行转发

9.5 TC消息处理

• 收到TC消息后，“有效时间”必须从消息头的Vtime字段中计算出来。然后应按以下方式更新拓扑集：

1 如果该消息的发送者接口不在该节点的对称1跳附近，则该消息必须为丢弃

2 如果拓扑集中存在一些元组，则：

T_last_addr ==发起者地址AND

T_seq> ANSN，

则不得对该TC消息进行进一步处理执行，并且消息必须被静默丢弃（收到的邮件无序）。

3 拓扑集中的所有元组，其中：

T_last_addr ==发起者地址AND

T_seq <ANSN

必须从拓扑集中删除。

4 对于接收到的每个通告的邻居主地址的 TC消息：

4.1 如果拓扑集中存在一些元组，则：

T_dest_addr ==通告的邻居主地址，并且

T_last_addr ==发起者地址，

那么该元组的保持时间必须设置为：

T_time =当前时间+有效时间。

4.2 否则，必须在拓扑中记录一个新的元组设置为：

T_dest_addr =通告的邻居主地址，

T_last_addr =发起者地址，

T_seq = ANSN，

T_time =当前时间+有效时间。

10. 路由表计算

• 每个节点维护一个路由表，该路由表允许其路由数据面向网络中的其他节点。

•路由表是基于本地链接信息库中包含的信息和拓扑集。因此，如果这些集合中的任何一个发生了变化， 重新计算路由表以更新路由信息。

•关于网络中的每个目的地。路线条目以下列格式记录在路由表中：

1. R_dest_addr R_next_addr R_dist R_iface_addr
2. R_dest_addr R_next_addr R_dist R_iface_addr
3. ,, ,, ,, ,,

•该表中的每个条目均由R_dest_addr，R_next_addr，R_dist和R_iface_addr组成。该条目指定由R_dest_addr标识的节点估计为距本地节点R_dist跳，具有接口地址R_next_addr的对称邻居节点是到达R_dest_addr的路由中的下一跳节点，并且该对称邻居节点可通过以下路径到达地址为R_iface_addr的本地接口。对于网络中已知路由的每个目的地，条目都记录在路由表中。该表中未记录路由已断开或仅部分已知的所有目的地。

• 更确切地说，当在任一路由表中检测到更改时，更新路由表
链接集，邻居集合，2跳邻居集，拓扑集，多接口关联信息库，

• 更准确地说，当邻居出现或丢失，创建或删除2跳元组，创建或删除拓扑元组或多个接口关联信息更改时，将重新计算路由表。此路由信息的更新不会在网络中或在1-hop邻居中生成或触发任何要传输的消息。为了构造节点X的路由表，对包含弧X-> Y的有向图运行最短路径算法，其中Y是X的任何对称邻居（邻居类型等于SYM），圆弧Y-> Z，其中Y是相邻节点，其意愿与WILL_NEVER不同，并且在2跳邻居集合中存在一个条目，其中Y为N_neighbor_main_addr，Z为N_2hop_addr，圆弧U-> V，其中存在一个条目在拓扑中，将V设置为T_dest_addr，将U设置为T_last_addr。

• 以下示例给出了计算（或重新计算）路由表的示例：

1 删除路由表中的所有条目。

2 从对称邻居（h = 1）作为目标节点开始添加新的路由条目。因此，对于邻居集中的每个邻居元 组，其中：

N_status = SYM

（存在到邻居的对称链接），并且对于邻居节点的每个关联链接元组，使得L_time> =当前时间，将记录新的路由条目在路由表中：

相关联的链接元组的
R_dest_addr = L_neighbor_iface_addr

相关联的链接元组的

R_next_addr = L_neighbor_iface_addr

R_dist = 1;

相关联的链接元组的

R_iface_addr =的L_local_iface_addr

如果在上面，没有R_dest_addr等于邻居的主地址，则必须添加另一个具有以下内容的新路由条目

R_dest_addr =邻居的主地址；

R_next_addr =相关链接元组之一的L_neighbor_iface_addr，其中L_time> =当前时间；

R_dist = 1;

R_iface_addr =关联的链接元组的L_local_iface_addr。

3 对于N2中的每个节点，即不是邻居节点的2跳邻居或节点本身，并且使得在2跳邻居集中存在至少一个条目，其中N_neighbor_main_addr对应于意愿不同的邻居节点在WILL_NEVER中，选择一个2跳元组，并在路由表中创建一个条目，其中包含：

R_dest_addr = 2跳邻居的主地址；

R_next_addr =条目中的R_next_addr
路由表具有：

R_dest_addr == N_neighbor_main_addr的2跳元组；

R_dist = 2；

R_iface_addr =条目中的R_iface_addr
路由表具有：

R_dest_addr == N_neighbor_main_addr的2跳元组；

3 路由表中记录了h + 1跳远的目标节点的新路由条目。必须对h的每个值执行以下过程，从h = 2开始，每次将其递增1。如果在迭代中没有记录新的条目，则执行将停止。

3.1 对于拓扑表中的每个拓扑条目，如果其T_dest_addr与路由表中任何路由条目的R_dest_addr不对应，并且其T_last_addr与R_dist等于h的路由条目的R_dest_addr相对应，则必须为新的路由条目记录在路由表中（如果尚不存在）：

R_dest_addr = T_dest_addr;

R_next_addr =所记录的R_next_addr
路由条目，其中：

R_dest_addr == T_last_addr

R_dist = h + 1; 和

R_iface_addr =记录的R_iface_addr
路线条目，其中：

R_dest_addr == T_last_addr。

3.2 几个拓扑条目可用于选择下一跳R_next_addr，以到达节点R_dest_addr。当h = 1时， 应断开联系，以便将具有最高意愿的节点和MPR选择器作为下一跳。

4 对于多接口关联库中存在路由条目的每个条目，例如：

R_dest_addr == I_main_addr（多个接口的关联条目）

并且没有路由条目，例如：

R_dest_addr == I_iface_addr

然后在路由表中使用以下内容创建路由条目：

R_dest_addr = I_iface_addr（多个接口的关联条目）

R_next_addr = R_next_addr（已记录的路线入口）

R_dist = R_dist（已记录的路线入口）

R_iface_addr = R_iface_addr（已记录的 路线条目）。

11.节点配置

•本节概述了应如何配置节点，以便在OLSR MANET中运行。

11.1 地址分配

•应该在定义的地址序列中为MANET网络中的节点分配地址，即，可以通过网络地址和网络掩码对MANET中的节点进行寻址。 Clausen& Jacquet实验[第50页] RFC 3626优化的链接状态路由2003年10月同样，应该为每个关联网络中的节点分配来自定义的地址序列的地址，该地址序列与MANET中使用的地址序列不同。

11.2 路由配置

•任何具有关联网络或主机的MANET节点都应进行配置，以使其路由设置为与关联主机或网络的接口。

11.3 数据包转发

• OLSR本身不执行数据包转发。而是，它在基础操作系统中维护路由表，假定该路由表按照RFC1812中的规定转发数据包。