(a)AODV路由发现

　　AODV路由协议是一种典型的按需驱动路由协议,该算法可被称为纯粹的需求路由获取系统,那些不在活跃路径上的节点不会维持任何相关路由信息,也不会参与任何周期路由表的交换。此外,节点没有必要去发现和维持到另一节点的路由,除非这两个节点需要进行通信。移动节点间的局部连接性可以通过几种方法得到,其中包括使用局部广播Hello消息。这种算法的主要目的是:在需要时广播路由发现分组一般的拓扑维护;区别局部连接管理(邻居检测)和一般的拓扑维护;向需要连接信息的邻居移动节点散播拓扑变化信息。

　　AODV使用广播路由发现机制,它依赖中间节点动态建立路由表来进行分组的传送。为了维持节点间的最新路由信息,AODV协议借鉴了DSDV中的序列号的思想,利用这种机制就能有效地防止路由环的形成。当源节点想与另外一个节点通信,而它的路由表中又没有相应的路由信息时,它就会发起路由发现过程。每一个节点维持两个独立的计数器:节点序列号计数器和广播标识。源节点通过向自己的邻居广播 RREQ(Route Requests)分组来发起一次路由发现过程。

　　(b)AODV协议反向路由的建立

　　在RREQ分组中包含了两个序列号:源节点序列号和源节点所知道的最新的目的序列号?源节点序列号用于维持到源的反向路由的特性,目的序列号表明了到目的地的最新路由。当RREQ分组从一个源节点转发到不同的目的地时,沿途所经过的节点都要自动建立到源节点的反向路由。

　　节点通过记录收到的第一个RREQ分组的邻居地址来建立反向路由,这些反向路由将会维持一定时间,一该段时间足够RREQ分组在网内转发以及产生的RREP分组返回源节点。当RREQ分组到达了目的节点,目的节点就会产生RREP分组,并利用建立的反向路由来转发RREP。

　　(c)AODV协议正向路由的建立

　　RREQ分组最终将到达一个节点,该节点可能就是目的节点,或者这个节点有到达目的节点的路由。如果这个中间节点有到达目的的路由项,它就会比较路由项里的目的序列号和RREQ分组里的目的序列号的大小来判断自己已有的路由是否是比较新的。

　　如果RREQ分组里的目的序列号比路由项中的序列号大,则这个中间节点不能使用己有的路由来响应这个RREQ分组,只能是继续广播这个RREQ分组。中间节点只有在路由项中的目的序列号不小于RREQ中的目的序列号时,才能直接对收到的RREQ分组做出响应?如果节点有到目的地的最新路由,而且这个RREQ还没有被处理过,这个节点将会沿着建立的反向路由返回RREP分组。

　　在RREP转发回源节点的过程中,沿着这条路径上的每一个.节点都将建立到目的节点的同向路由,也就是记录下RREP是从哪一个邻居节点来的地址,然后更新有关源和目的路由的定时器信息以及记录下RREP中目的节点的最新序列号。对于那些建立了反向路由,但RREP分组并没有经过的节点,它们中建立的反向路由将会在一定时间(Active-Route-Timeout)后自动变为无效。收到RREP分组的节点将会对到某一个源节点的第一个RREP分组进行转发,对于其后收到的到同一个源的RREP分组,只有当后到的RREP分组中包含了更高的目的序列号或虽然有相同的目的序列号但所经过的跳数较少时,节点才一会重新更新路由信息,以及把这个RREP分组转发出去。

　　这种方法有效地抑制了向源节点转发的RREP分组数,而且确保了最新及最快的路由信息。AODV协议中规定，源节点将在收到第一个RREP分组后,就开始向目的节点发送数据分组。如果以后源节点了解到的更新的路由,它就会更新自己的路由信息。