# Ad Hoc网络和路由协议

## Ad Hoc网络

Ad Hoc网络是一个没有有线基础设施支持的移动网络。在Ad Hoc网络中，所有的节点都是由移动主机构成的。最初是应用于军事领域，是为了在战场环境下分组无线网络数据的通信。

Ad Hoc网络是一种独具特色的网络，作为一种新型的无线、多跳、无中心分布式控制网络，它无需网络基础设施和管理中心，而是由一组自主的移动节点临时组成，通过移动节点间的相互协作和自我组织，保持网络连接和实现数据的传递，主要应用于军事战场、医疗抢险以及抗洪救灾等特殊紧急环境。具有很强的自组织性、鲁棒性、抗毁性和容易构建的特点。

Ad Hoc网络组网灵活、快速，使用非常方便，但必须为Ad Hoc设计专门的协议和技术，传统固定网络和移动蜂窝网络中的技术和协议无法直接复制到Ad Hoc网络。

## 2. 路由协议

路由协议是Ad Hoc网络的重要组成部分。

**与传统网络协议相比，Ad Hoc网络路由协议的开发更具挑战性**：

这是因为传统网络的路由方案都假设网络的拓扑结构是相对稳定的，而事实上Ad Hoc网络的网络拓扑结构是不断变化的。

传统网络的路由方案主要依靠大量的分布式数据库，这些数据库保存在某些网络节点和特定的管理节点中，而Ad Hoc网络中的节点不会长期存储路由信息，并且这些存储的路由信息也不总是可靠的。

因此，大量的研究表明，理想的Ad Hoc网络路由协议必须具备以下功能：a）维护网络拓扑的连接。b）及时感知网络拓扑结构的变化。c）高度的自适应性。

根据路由表的维护特点，Ad Hoc网络的路由协议大致可分为：a）表驱动路由协议。b）按需驱动路由协议。c）混合路由协议。

目前，大多数Ad Hoc网络路由协议采用的是按需驱动路由方式，其中，具有代表性的有动态资源路由协议（DSR）、Ad Hoc请求距离向量协议（AODV）和定位辅助路由协议（LAR）等。

Ad Hoc网络\_百度文库 <https://wenku.baidu.com/view/5e3b0bb269dc5022aaea00c6.html>

# DSR路由协议分析

## 1. 定义

DSR即Dynamic Source Routing（动态源路由协议），是在移动自组网(MANET)中使用的一种路由协议，它工作在TCP/IP协议族的网际层。是一个专门为多跳无线Ad Hoc网络设计的简单且高效的路由协议。它的特别之处在于它运用了源路由算法，即只有发送节点知道完整的到目的节点的多跳路由。所有的路由都是由DSR路由协议动态地、自动地确定和维护，它提供快速反应式服务，以便帮助确保数据分组的成功交付，即使在节点移动或者其他网络状况变化的条件下也是如此。

DSR路由协议有两个主要机制组成——路由发现(Route Discovery）机制和路由维护(Route Maintenance)机制。路由发现和路由维护协同工作保证了节点维持的到达目的节点的路由的及时性和有效性。

### 1.1 路由发现

当源节点S要发送一个数据分组到目的节点D，但是源节点并不知道到达目的节点的路由信息时，源节点就会发起一次路由发现过程。为了建立一条路由，源节点广播一个路由请求(Route Request)分组，当该请求分组到达目的节点，或者是到达某个中间节点且该节点具有到达目标节点的路由信息时，这些节点就向源节点发送一个包含着S到D的完整路由信息的路由应答(Route Reply)分组，源节点s就会根据这些信息建立新的路由。

### 1.2 路由维护

由于网络中各节点的移动性，网络拓扑随时会发生变化，一条路径中的某两个节点可能会因距离超出双方的传输半径或其它的原因诸如中间节点故障等而导致现存路由信息的失效。当路由维护指明某个源路由失效时，就使用路由错误(Route Error)分组通知源节点S，源节点S就会尝试使用其它可以到达目的节点D的路由路径，或者再一次发起路由发现过程来寻找一条新的路由路径。这个过程被称为路由维护。

安全DSR路由协议分析与设计\_百度文库 <https://wenku.baidu.com/view/f6080818a76e58fafab003f3.html>

基于DSR的路由协议综述\_百度文库 <https://wenku.baidu.com/view/f51a0589cc22bcd126ff0cf3.html>

DSR路由协议分析与仿真 - 豆丁网（找到里面介绍DSR那一块感觉很有用）

<http://www.docin.com/p-163766231.html>

## DSR分组结构

它的首部采用扩展性良好的TLV格式（一种可变格式，“T”-Type：类型；“L”-Length：长度；“V”-Value：值），除固定部分外，不同类型的选项(option)以TLV格式附加在固定部分之后。选项的种类包括：

* + 路由请求(Route Request)：它是在路由发现过程中源节点发出的路由请求分组中的Option类型；
  + 路由应答(Route Reply)：它是在路由发现过程中目的节点或者能够到达目的区域的节点发给源节点的路由应答中的Option类型；
  + 确认请求(ACK Request)：它是在路由维护阶段中间转发节点定期地向其下一跳所发出的确认路由分组中的Option的类型；
  + 确认（ACK)：当在维护阶段，发出ACK Request报文节点的下一跳所使用的路由确认请求，用以确认路由的有效性，而这个确认分组中Option的类型就是ACK；
  + 错误路由（Route Error）：在路由维护阶段，当路径中的某个节点发现路径失效（由于拓扑变化或信道差错）时，就发起一个Option类型为Route Error的路由错误报文给源节点，源节点就尝试使用缓存中的其他可以到达目标节点的路径，或者再发起一次路由发现来寻找一条新路径；
  + 源路由(Source Route)：路由建立完成之后，源节点开始向目的节点传送数据，这是分组中的Option类型为Source Route，其中保存了源节点到目的节点的一条原路由，这样分组就可以径直地通过这条源路由到达目的区域。

其分组结构如图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| Next Header | F | Reserved | Payload length |
| Options | | | |

在DSR分组头中，Next Header代表了下一跳；F是标识符，能够从中识别出它是DSR分组；Reserved是保留字段，以便未来的扩展用途；Payload length为有效载荷长度；最后的Options作为不同阶段DSR分组的类型。

## 数据结构

使用DSR协议进行通讯的任何节点都需要维护如下四个概念性数据结构，即路由缓存区（Route Cache）、发送缓存区（Send Buffer）、路由请求表（Route Request Table）和重传缓冲区（Retransmission Buffer）。

### 路由缓存

一个使用DSR参与无线传感器网络的结点所需的所有路由信息都存储在路由缓存中。

网络中的每个节点维护自己的路由缓存。当一个结点听到传感器网络中结点之间的新链路时会将该信息添加进缓存中；同样的当一个结点得知现存路由信息失效时，将会从缓存中删除该信息如果路由缓存溢出，需要采用LRU(Least Recently Used)算法来进行淘汰处理。

### 发送缓冲区

结点的发送缓冲区是一些由于该结点没有一个到达目的节点的源路径而不能被该结点发送的分组组成的队列。

一般来讲，节点在将分组插入发送缓冲区的同时就会发起一个路由发现过程，如果路由发现过程成功的话，这些分组就会被发送出去。在发送缓冲区中的每个分组都标记了它进入发送缓冲区的时间，当在被放入SEND\_BUFFER\_TIMEOUT秒后，它将被丢弃。

如果必须的话，可以采用FIFO策略将分组在没有超时的情况下将其丢弃以防止发送缓冲区的溢出。

### 路由请求表

如果某节点发送或转发了发往某目的结点的路由请求分组，则在接下来的DSR\_REQ\_TIMEOUT时间间隔内，该节点将不允许再次向这个目的结点发送路由请求分组。

节点的路由请求表是最近该结点发出的或转发的路由请求分组的集合，其中每一个条目都记录了其被插入路由请求表的时间，当超时后这些条目将会被删除，如果路由请求表溢出，需要采用LRU算法来进行淘汰处理。

### 重传缓冲区

DSR路由维护的基础就是每个节点需要在发出或者转发分组时保证分组能够顺利地被下一跳的节点所接收；

如果不能确认下一跳已经成功接收到了的分组，该节点需要做出相应的反应，如重传分组或者到达最大重传次数MAX\_RETRANSMITE\_TIME后丢弃该分组，同时，向源节点发送路由错误分组。

重传缓冲区是节点发送或转发的分组中那些尚未得到下一跳确认的分组的集合。

## 改进设计

针对DSR路由协议存在的不足，现有一些改进设计。

首先针对DSR缓冲器先引入以下三个参数：

1. 形成时间路由参数：定义加入路由最晚的节点的时间为形成时间路由参数，记为TB。t表示DSR缓冲器中路由的某一节点加入到路由中的时间，则TB=max(t)；
2. 生存时间路由参数：定义路由中相邻的节点之间构成的链路从形成到失效的时间为生存时间路由参数，记为TL；
3. 剩余生存时间路由参数：定义生存时间路由参数与路由已存活的时间参数之差为剩余生存时间路由参数，记为TR。TC表示当前时间，则TR=TL-(TC-TB)。

1.针对DSR缓冲策略存在的不足，对DSR的缓冲策略进行以下改进：

选择最佳路径时，首先选择路由长度最短的路由，在最短路由不止一条的情况下，则选择最短路径中剩余生存时间最长max(TR)的一条路由。当缓存器满时,就舍弃剩余生存时间路由参数最小min(TR)的路由。

2.优化DSR路由的自动缩短机制，主要进行以下两方面改进：

1. 寻找跳数最小、所产生的新路由的路由质量最好的路由。TL决定路由的质量，取TL最大值的路由；
2. 通过新形成链路的移动节点速度、移动方向、位置估算出新形成链路的TR，来判断是否要让节点使用新的链路路由。如果产生的新链路的剩余时间参数TR小于原来链路的TR，则路由缩短机制产生的路由是没有意义的，这条新链路被判断为失效。采用本方案设计的路由自动缩短机制不仅能产生“跳数尽可能小”的路由，同时还确保了新形成路由的TR不会很小。

一种自组织网络DSR路由协议改进方法的研究 <http://tech.hqew.com/fangan_1574074>

# DSR路由仿真

## NS2概述

NS2是一个离散事件模拟器。离散事件模拟，是几种常用的系统模拟模型之一。简单地说，事件规定了系统状态的改变，状态的修改仅在事件发生时进行。在一个网络模拟器中，典型的事件包括组到达、时钟超时等。

NS2的核心部分是一个离散事件模拟引擎。NS2中有一个“调度器”（Scheduler）类，负责记录当前时间，调度网络事件队列中的事件，并提供函数产生新事件，指定事件发生的时间。

NS2的构件库所支持的网络类型包括广域网、局域网、移动通信网、卫星通信网等，所支持的路由方式包括层次路由、动态路由、多播路由等。NS2还提供了跟踪和监测的对象，可以把网络系统中的状态和事件记录下来以便分析。另外，NS2的构件库中还提供了大量的数学方面的支持，包括随机数产生、随机变量、积分等。

## 分裂对象模型

NS2使用分裂对象模型，是出于兼顾模拟性能和灵活性两方面的考虑。

一方面，C++是高效的编译执行语言，使用C+实现功能的模拟，可使模拟过程的执行获得较好的性能。

另一方面，Otcl 是解释执行的，用Otcl进行模拟配置，可以在不必重新编译的情况下随意修改模拟参数和模拟过程，提高了模拟的效率。

同时，这种分裂对象模型增强了构件库的可扩展性和可组合性，用户通常只需要编写Otcl脚本就可以把一些构件组合起来，成为一个“宏对象”。

这种分类对象模型也起到了抽象的作用，它对用户屏蔽了功能实现的细节。用户通过Otcl进行模拟配置,很多情况下只需要了解构件的使用和配置接口就可以了，而不需要了解这些构件的功能是如何是实现的。

NS2中所体现的这些先进的设计思想使得NS2成为了一种实用的网络模拟器。

同时，NS2是免费的、开放源代码的。

这使得利用NS2进行网络模拟的研究者可以很方便地扩展NS2的功能，也可以很方便地共享和交流彼此的研究成果。NS2吸纳了这些NS2开发者贡献的各方面的模块，从而使它的构件库不断地丰富。

## 添加路由协议的步骤

NS2中所运行的路由协议代码一般用C++编写，在编写路由协议时，我们需要注意，一般的路由协议是由五个最基本的文件构成(要添加的路由协议这里先用proto代指)，它们分别是protoname.h, protoname.cc, protoname\_pkt.h, protoname\_rtable.h，protoname\_rtabl.cc。也就是说，实现了这五个文件，路由协议也就可以仿真运行了。

protoname.h：这是路由协议的头文件，里面定义了所有必须的计时器以及实现路由协议功能的路由代理；

protoname.cc：在这个文件里实现所有的定时器、路由代理以及与Tcl的部分绑定；

protoname\_pkt.h：这个文件定义了在节点之间交换的所有路由协议分组的格式；

protoname\_rtable.h：这个文件定义了路由协议的路由表；

protoname\_rtable.cc: 这个文件里面的内容是对路由表的实现

基于NS2下的DSR路由协议的研究与优化(具体仿真代码步骤)\_百度文库 <https://wenku.baidu.com/view/ceb2bb8014791711cd791733.html>