

**网络协议设计与性能仿真**

## OLSR和AODV路由协议性能比较与分析

**学号：22120397**

**姓名：李秀文**

**目录**

[1 摘要 3](#_Toc136247191)

[2 Ad Hoc网络路由协议 3](#_Toc136247192)

[3 Ad Hoc网络性能评价指标 5](#_Toc136247193)

[4 实验设计 6](#_Toc136247194)

[5 实验结果及分析 9](#_Toc136247195)

[6 总结 18](#_Toc136247196)

# 摘要

**摘要：**移动Ad Hoc网络是移动节点动态临时组建的自组织网络，路由协议一直都是Ad Hoc网络研究的关键问题。使用NS2网络仿真软件，选取分组投递率、端到端平均时延、归一化路由开销和路由发现频率4个指标对，在不同的场景下分析比较移动Ad Hoc网络的两种路由协议OLSR和AODV的性能，得出不同路由协议在不同场景下的优缺点和适用的网络环境。

关键词:移动Ad Hoc网络;路由协议;网络仿真

# Ad Hoc网络路由协议

我们经常提及的移动通信网络一般都是有中心的，要基于预设的网络设施才能运行。例如，蜂窝移动通信系统要有基站的支持；无线局域网一般也工作在有AP接入点和有线骨干网的模式下。但对于有些特殊场合来说，有中心的移动网络并不能胜任。比如，战场上部队的快速展开和推进，地震或水灾后的营救等，这些场合的通信不能依赖于任何预设的网络设施，而需要一种能够临时快速自动组网的移动网络，Ad hoc网络可以满足这样的要求。目前已提出很多用于Ad Hoc网络的路由协议，按照驱动模式的不同，可以分为表驱动路由协议、按需路由协议以及混合路由协议。常用的表驱路由协议有DSDV、OLSR等；按需路由的代表协议有DSR、AODV等。

**（1）OLSR**

OLSR协议是一种基于链路状态算法的先验式路由协议，它是对标准链路状态路由协议进行了一定的优化而形成的。由于Ad Hoc网络中的节点具有移动性，网络拓扑结构处于不断变化中，因此就需要实时更新本地的路由信息，也即需要网络中的每个节点不断地向周围节点广播自己在当前网络中的位置。但无线网络中的带宽资源有限与广播需要花费较大开销成为了主动路由协议中的一个矛盾点，OLSR的主要目的就是为了有效地限制节点的广播信息。

OLSR的核心技术是多点中继机制，多点中继的基本思想是通过减少同一区域内的网络控制分组的重复转发次数来降低洪泛开销。网络节点N从它的一跳邻居节点之中选择一部分节点作为多点中继节点MPR (Multi Point Relay)，使得节点N通过MPR集合转发的分组能够被节点N的所有两跳节点接收到，节点N被称为MPR集合的多点中继选择节点(MPR Selector )。根据这种设计思想，只有MPR节点才负责转发网络控制分组，而非MPR节点只接收控制分组不负责转发。OLSR协议按照MPR集合来计算由源节点到目的节点的路由,网络中每个节点经过各自MPR集合广播自己的 MPR Selector信息，并接收其他节点的信息后更新自己的路由表。

**（2）AODV**

AODV为一种反应式路由协议，当需要路由时才构建。在自组网中，AODV提供了快速适应动态链接条件、自启动、多跳路由。实现较低的处理和内存开销以及较低的网络利用率，并且决定在自组网内单播路由到目的地。在自组网中，用每个路由表入口的目的序列号来确保循环自由和确定最近期的路径。在AODV协议里，当源要求一个到目的地的路径时，路由发现将被执行，路由请求RREQ（Route Request）消息在网络中从源扩散出去，直到收到一个这样的消息，节点检查它的本地路由缓存来检验新的到要求的目的地路由是否有效。如果有效，节点单播一个路由应答RREP（Route Reply）消息到源节点。否则，使用本地重复消除纯扩散机制再传输一个RREQ。作为优化，为了更宽地控制RREQs的广播，当扩散消息时（即一个受限的TTL（Time To Live）消息发布时），AODV使用扩展环搜索技术。如果在某一时间没有收到RREP消息，那么一个大的TTL消息被发布；如果依然没有回复，TTL逐步增加，直到到达一个门限值。在路由被发现后，路由就建立起来了，任意缓存在源节点的IP分组将被传输出去；如果没有路由被发现，分组就被丢弃。当一个检测链路被打断时，监测点就发布一个路由错误RERR（ Route Error）消息给它的邻居，而此时该邻居已经使用了这个被打断的链路。因此，该节点将不得不发布一个新的RREQ来修复被打断的路由。

# Ad Hoc网络性能评价指标

**（1）端到端传输时延：**所有数据分组从源节点发出到目的节点正确接收所用时间的平均值，亦即数据包从源节点发出到目的节点接收之间的平均时间差，反映了路由的有效性。

**（2）数据分组投递率：**节点接收到的总分组个数与发送出去的总分组个数的比值，反映了网络传输的可靠性。分组投递率越大，可靠性越高。数据分组投递率是衡量网络丢包率的重要指标，反映了网络性能的好坏。

**（3）归一化路由开销：**每发送一个数据分组所需要的路由分组数。其中，路由分组每一跳的传输均认为是一个新的路由分组，反映了网络的拥塞程度和路由效率。对于Ad hoc网络中的路由协议，路由开销越低越好。

**（4）网络吞吐量：**网络吞吐量是指网络在规定时间内传输的数据量，是评估路由协议性能的重要指标之一，它反映了路由协议的数据传输能力。一个高效的路由协议应该能够在高负载情况下保持较高的网络吞吐量，以保证网络数据的高效传输。通过评估网络吞吐量，可以评估路由协议在网络传输方面的性能，以及识别网络瓶颈和优化网络性能的机会。

# 实验设计

NS2.35 是一款网络仿真软件，基于C++和OTCL两种语言，主要用于网络协议、网络性能、网络安全等方面的仿真和研究。支持多种网络拓扑结构，如总线型、环型、星型、树型、网状型等。在NS2.35中，用户可以自定义拓扑结构，并设置拓扑参数，如节点数、总线长度、环长度等，可以根据自己的需求设置仿真参数、网络拓扑、协议参数、设备参数等，以模拟真实的网络环境，并进行网络性能分析、安全分析等，是一款功能强大的网络仿真软件。

本实验的目的是对比Ad Hoc网络中AODV和OLSR两种路由协议的性能，共设置了六种实验场景，分别对比了在不同节点个数、不同场地范围大小，不同的节点移动速度、不同的节点停留时间、不同CBR连接数、不同节点发包率设置下的两种路由协议的端到端时延、分组投递率、归一化路由开销以及吞吐量等性能的对比。

**（1）不同节点个数的场景设置**

仿真的业务场景设置：节点个数由10到100，以固定码率为业务源（产生UDP业务），10对通信连接，每秒发送2个分组（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下，其中j的取值范围为10~100，取值间隔为10，具体设置如下：

|  |
| --- |
| ns cbrgen.tcl -type cbr -nn $j -seed 1 -mc 5 -rate 2.0 > cbr-30n-10c-2p |

节点运动场景设置：设置由10到100的节点个数，在600m×600m的区域范围内以10m/s的移动速度随机运动，具体设置如下，其中j的取值范围为10~100，取值间隔为10，具体设置如下：

|  |
| --- |
| setdest $j -p 30 -M 10 -t 150 -x 600 -y 600 > scene-30n-0p-10M-150t-600-600 |

**（2）不同场地范围大小的场景设置**

仿真的业务场景设置：20个节点，以固定码率为业务源（产生UDP业务），10对通信连接，每秒发送2个分组（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下：

|  |
| --- |
| ns cbrgen.tcl -type cbr -nn 20 -seed 1 -mc 10 -rate 2.0 > cbr-30n-10c-2p |

节点运动场景设置：20个移动节点，在从100m×100m到1000m×1000m的范围内以10m/s的移动速度随机移动，具体设置如下，其中j的取值范围为10~100，取值间隔为10，具体设置如下：

|  |
| --- |
| setdest -n 20 -p 30 -M 10 -t 150 -x $j\*100 -y $j\*100 > scene-30n-0p-10M-150t-600-600 |

**（3）不同节点移动速度的场景设置**

仿真的业务场景设置：20个节点，以固定码率为业务源（产生UDP业务），10对通信连接，每秒发送2个分组（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下：

|  |
| --- |
| ns cbrgen.tcl -type cbr -nn 20 -seed 1 -mc 10 -rate 2.0 > cbr-30n-10c-2p |

节点运动场景设置：20个节点，在600m×600m的范围内，以10m/s到170m/s的移动速度范围随机移动，具体设置如下，其中j的取值范围为0~170，取值间隔为20，具体设置如下：

|  |
| --- |
| setdest -n 20 -p $j -M 10 -t 150 -x 600 -y 600 > scene-30n-0p-10M-150t-600-600 |

**（4）不同节点停留时间的场景设置**

仿真的业务场景设置：20个节点，以固定码率为业务源（产生UDP业务），10对通信连接，每秒发送2个分组（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下：

|  |
| --- |
| ns cbrgen.tcl -type cbr -nn 20 -seed 1 -mc 10 -rate 2.0 > cbr-30n-10c-2p |

节点运动场景设置：20个节点，在600m×600m的范围内，以10m/s的移动速度范围随机移动，停顿时间取值范围为0到100，具体设置如下，其中j的取值范围为0~100，取值间隔为10，具体设置如下：

|  |
| --- |
| setdest -n 20 -p 10 -M $j -t 150 -x 600 -y 600 > scene-30n-0p-10M-150t-600 |

**（5）不同CBR连接数的场景设置**

仿真的业务场景设置：50个节点，以固定码率为业务源（产生UDP业务），通信连接对数的取值范围为2~20，每秒发送2个分组（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下，其中j的取值范围为2~20，取值间隔为2，具体设置如下：

|  |
| --- |
| ns cbrgen.tcl -type cbr -nn 50 -seed 1 -mc $j -rate 2.0 > cbr-30n-10c-2p |

节点运动场景设置：50个节点，在600m×600m的范围内，以10m/s的移动速度范围随机移动，停顿时间为10s，具体设置如下：

|  |
| --- |
| setdest -n 50 -p 10 -M 20 -t 150 -x 600 -y 600 > scene-30n-0p-10M-150t-600 |

**（6）不同节点发包速率的场景设置**

仿真的业务场景设置：50个节点，以固定码率为业务源（产生UDP业务），5对通信连接，每秒发送分组个数范围为1~10（每个数据包512B）的以CBR为业务源的业务场景文件。具体设置如下：

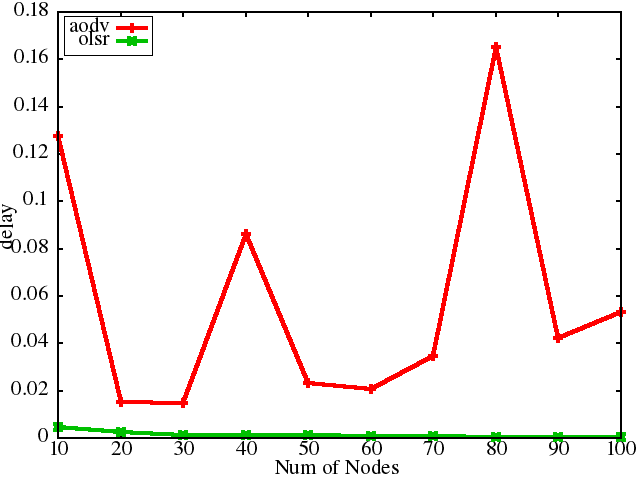
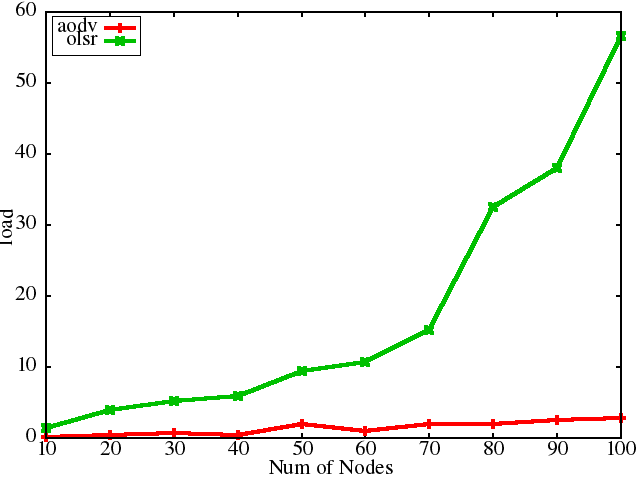
|  |
| --- |
| ns -type cbr -nn 50 -seed 1 -mc 5 -rate $j > cbr-30n-10c-2p |

节点运动场景设置：50个节点，在600m×600m的范围内，以10m/s的移动速度范围随机移动，停顿时间为10s，具体设置如下：

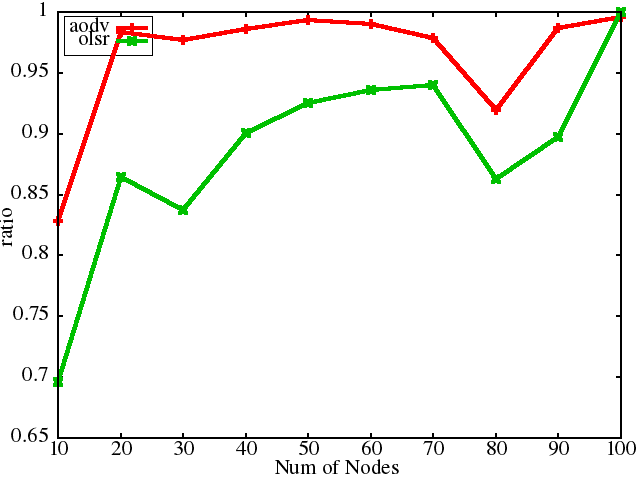
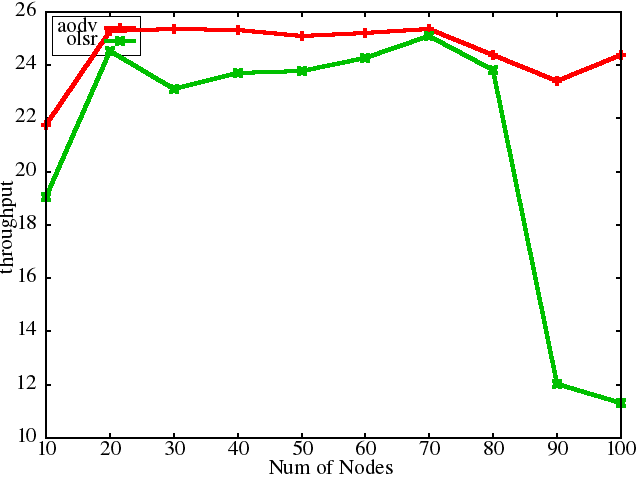
|  |
| --- |
| setdest -n 50 -p 10 -M 20 -t 150 -x 600 -y 600 > scene-30n-0p-10M-150t-600 |

# 实验结果及分析

**（1）不同节点个数下的仿真结果**

（a）delay (b) load

（c）ratio (d) throughput

图1 不同节点数目下的仿真结果

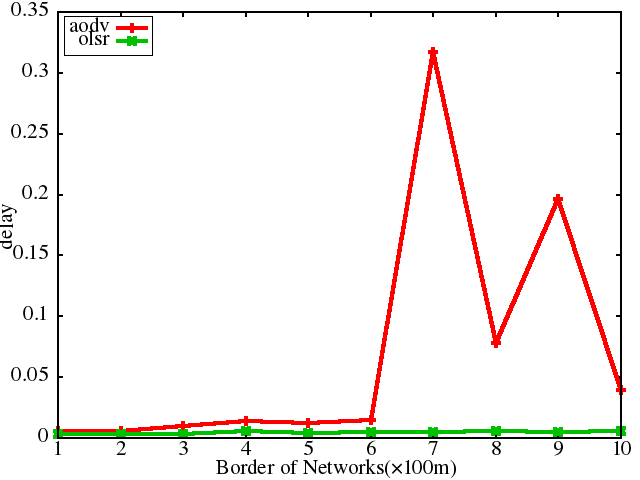
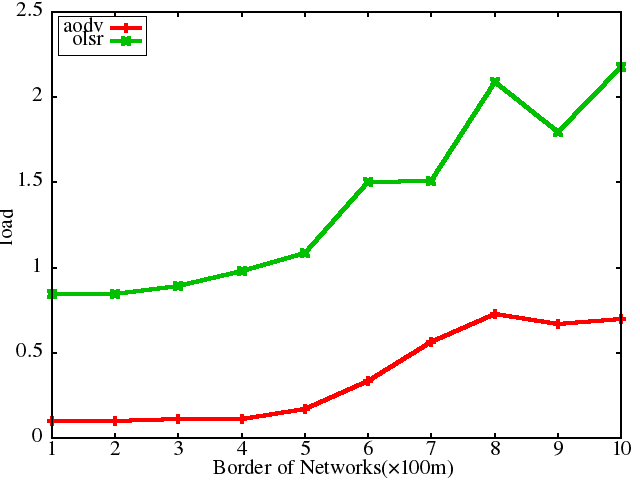
由图1（a）可以看出，随着节点数目的增加，OLSR路由协议在端到端延迟这一性能方面的表现相对稳定，且一直优于AODV协议，主要是因为OLSR能够立即转发和丢弃数据，并且分组不需要等待，但是为了传送引起长延时的分组﹐AODV需要执行路由发现过程。

由图1（b）可以看出，随着节点数目的增加，OLSR和AODV这两个路由协议的路由开销都有所增加，因为在路由发现过程中源和目的地间距离和跳数的增加使得路由开销增加，但AODV协议只有在节点有数据要发送时才发起路由寻找过程，这样就不需要通过周期性的广播路由信息来维护节点间的路由路径，进而相比于OLSR有较低的路由开销。

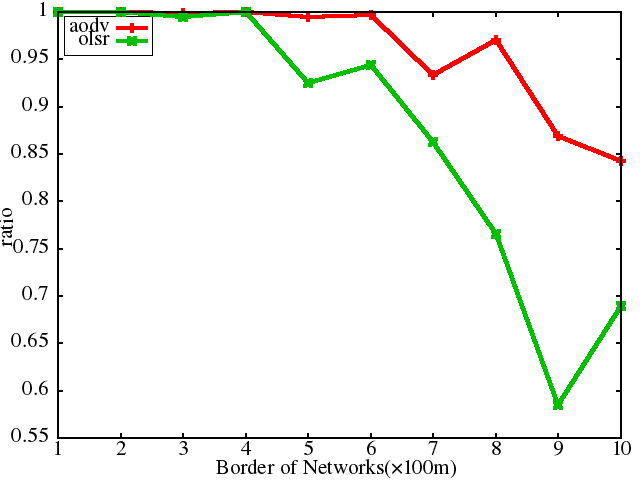
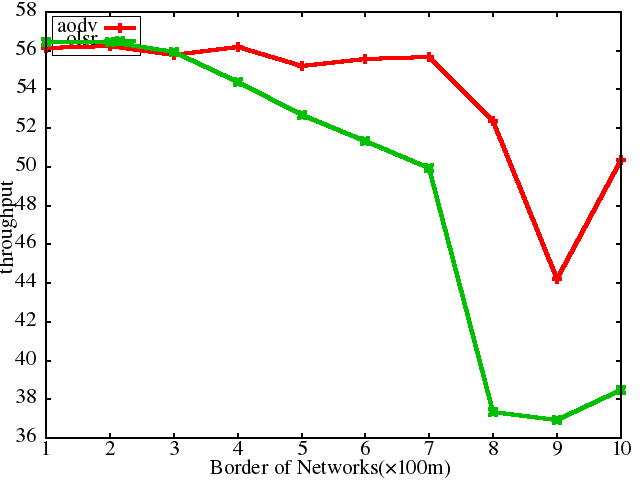
由图1（c）可以看出，随着节点数目的增加，在分组投递率方面，这两个路由协议的表现都总体呈现下降趋势，这是因为随着节点的增多，节点间的碰撞和冲突也会增多。

由图1（d）可以看出，由于节点数目的增多，拥塞的可能性增大，会导致吞吐量下降，但是AODV的吞吐量总体是大于OLSR路由协议的。

**（2）不同场地范围下的仿真结果**

（a）delay (b) load

（c）ratio (d) throughput

图2 不同区域大小下的仿真结果

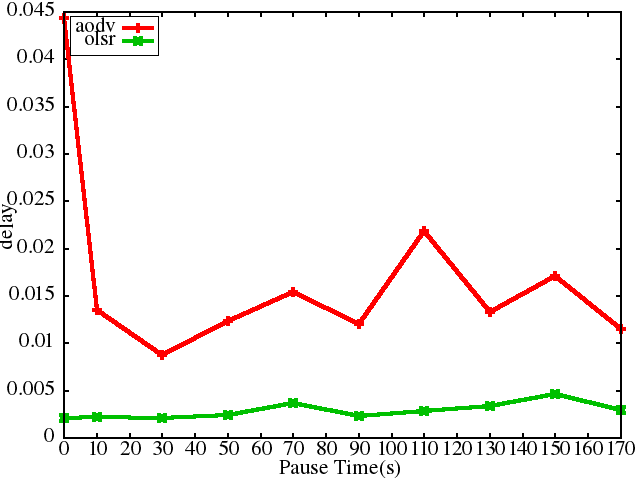
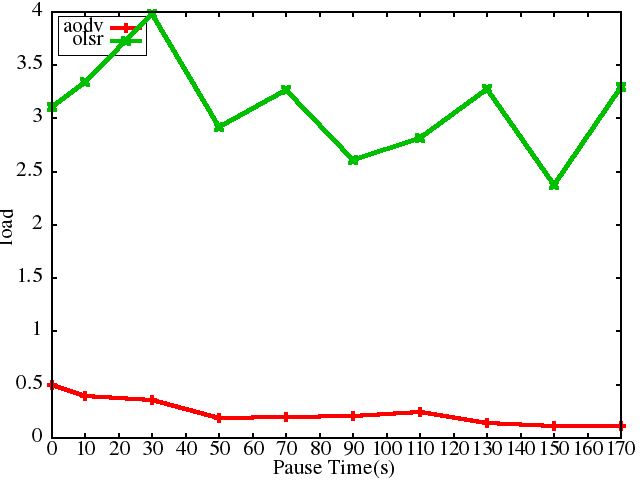
图2（a）表明，随着距离的增加，AODV路由协议的端到端延迟逐渐增加，而OLSR路由协议的端到端延迟虽然有所增加，但总体上来说仍然保持稳定。

图2（b）表明，随着距离的增加，AODV和OLSR路由协议的网络负载都有所上升，因为距离增大导致的链路质量降低，而相比于AODV协议，OLSR需要维护全局拓扑信息，所以OLSR路由协议的路由开销大于AODV。

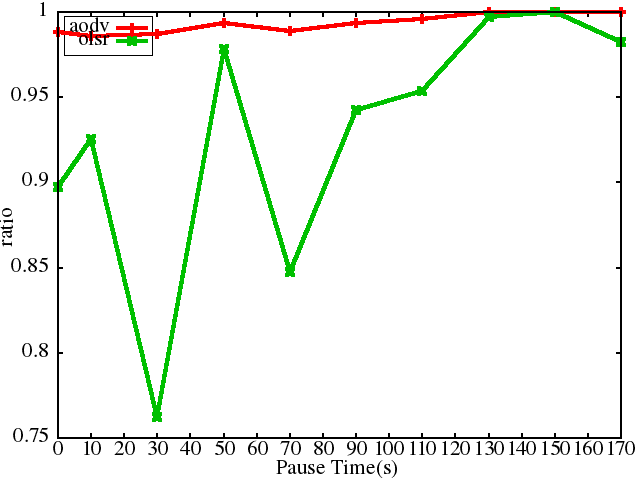
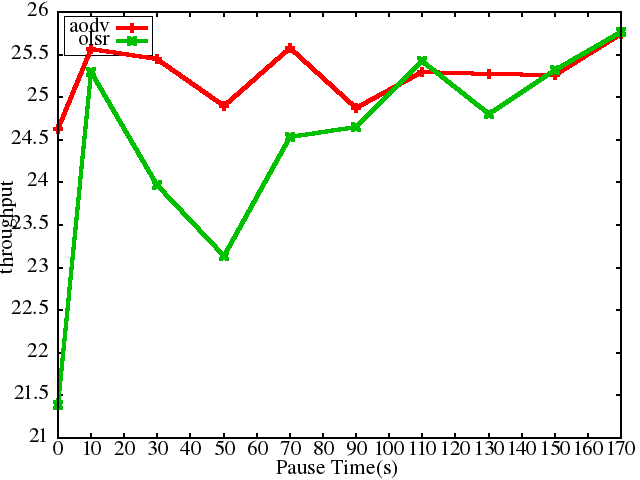
图2（c）表明，随着区域范围的增大，节点间的通信距离增大，通信质量下降，分组投递率都逐渐降低。在AODV中，当距离越远时，节点需要更多的时间来发现目标节点并建立路由，因此分组投递率会下降。在OLSR中，当距离越远时，节点之间的链路质量会下降，因此分组投递率也会下降。

图2（d）表明，随着区域范围的增大，两种路由协议的吞吐量都有所下降，因为通信距离变长，信号范围变差，单位时间内通过网络的实际数量会有所降低，故都有所下降。

**（3）不同节点停留时间下的仿真结果**

（a）delay (b) load

（c）ratio (d) throughput

图3 不同节点停留时间下的仿真结果

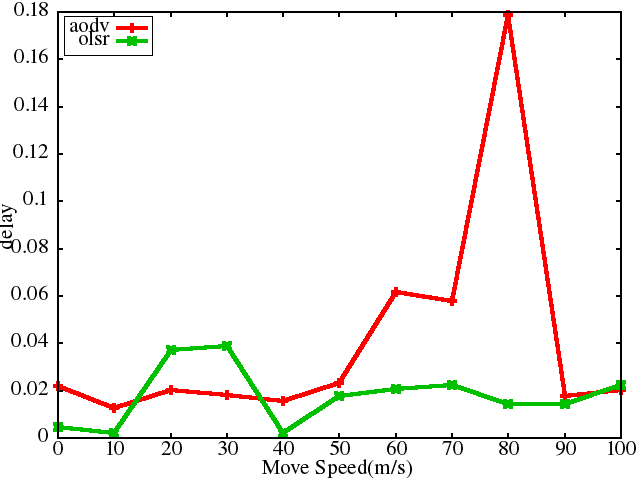
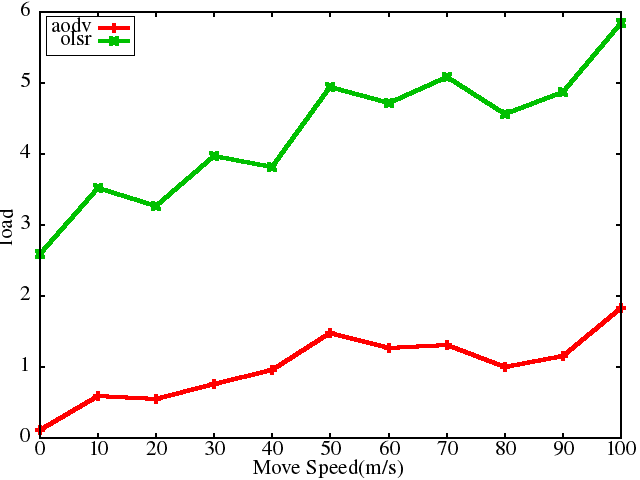
由图3（a）可以看出，随着节点停留时间的延长，AODV路由协议的端到端延迟显著下降，这是因为AODV路由协议是一种按需路由协议，它只在需要时才建立路由，而不是像其他协议那样在网络中建立所有可能的路由。因此，当停留时间增加时，节点之间的连接更加稳定，AODV路由协议的端到端延迟也会降低。而OLSR路由协议的端到端延迟随着停留时间的延长几乎不会发生变化，这是因为OLSR路由协议是一种基于多点中继的路由协议，它使用逐跳路由，即每个节点使用其本地信息为分组选择传输路由。因此，当停留时间增加时，节点之间的连接更加稳定，但是OLSR路由协议的端到端延迟不会发生变化。

由图3（b）得到，随着停留时间的增加，AODV协议的负载持续下降，因为AODV时是按需路由协议，当节点静止时间增加时，网络的动态拓扑性减弱，需要更新的路由信息大大减少，从而路由开销下降，而OLSR路由协议的路由开销变化呈现波动态，下降并不明显，因为OLSR需要一直维护全局拓扑信息。

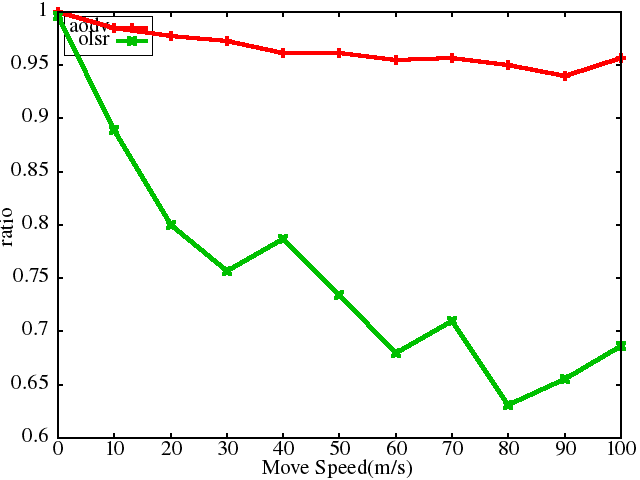
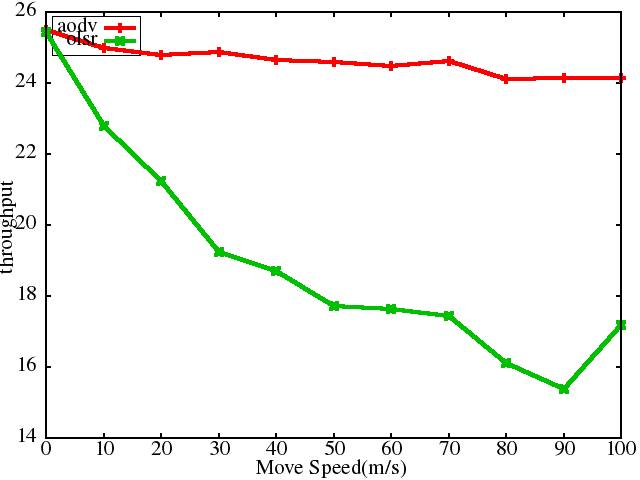
图3（c）显示，随着节点停留时间的增加，两种协议的分组投递率都有所上升，因为节点停留时间越长，网络拓扑越稳定，节点间的链路变化程度降低，故正确发包收包的概率上升。

图3（d）表明随着节点停留时间的增加，两种协议的网络吞吐量都有明显上升，可类比于分组投递率上升的原因。

**（4）不同节点移动速度下的仿真结果**

（a）delay (b) load

（c）ratio (d) throughput

图4 不同节点移动速度下的仿真结果

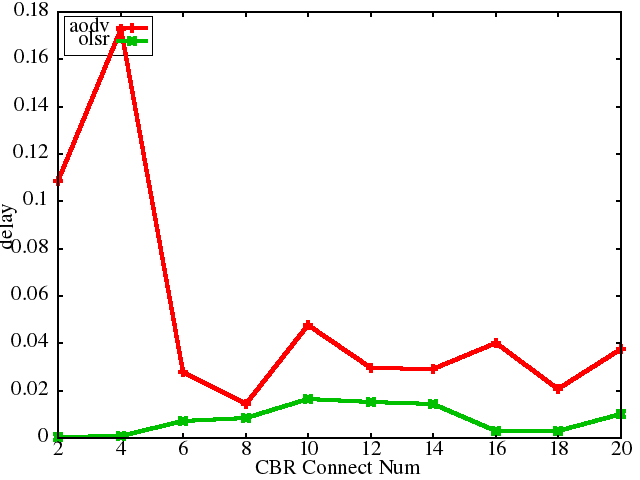
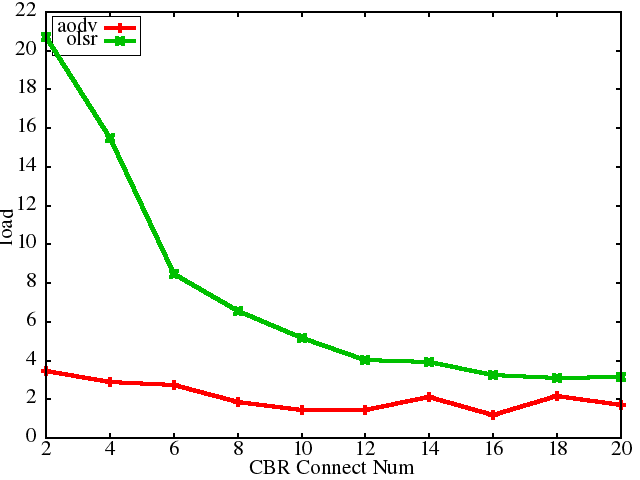
由图4（a）看出，节点移动速度持续增加，OLSR一直呈现较低延迟，因为OLSR维护全局拓扑信息，有数据需要发送的时候可以立即发送到目的地。

图4（b）显示，随着节点移动速度增加，两种路由协议的负载都有所上升，但OLSR的路由开销一直大于AODV，因为OLSR协议的路由通信需要定期广播出去，而AODV协议如果没有数据需要发送，节点就不需要发送路由通信。

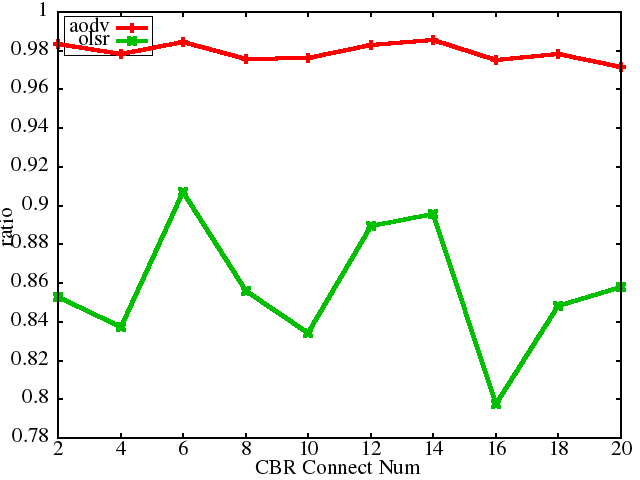
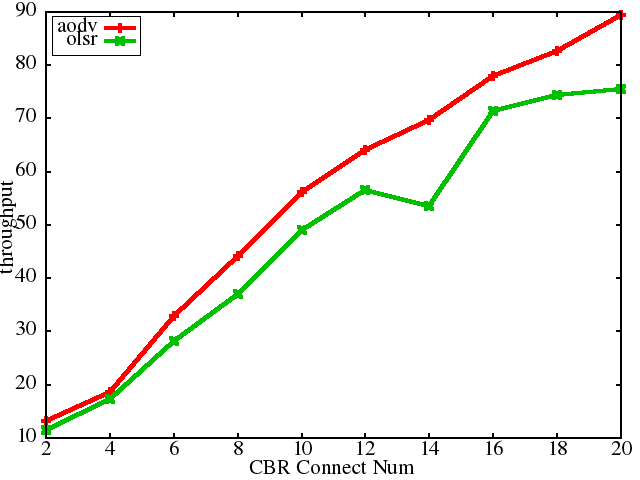
图4（c）显示，随着节点移动速度增加，OLSR的分组投递率有明显的下降趋势，而AODV则相对稳定。因为在高速移动情况下，分组从源产生的过程中，OLSR协议里的节点可以立即发送和转发分组到目的地，如果该目的地已经移动到另一个地点，该分组将被丢弃。但在AODV协议中，在到达目的地的路由被发现前，分组将被存储在缓存器里，因此，移动性对PDR的影响不是很大。

正如图4（c）解释的，节点移动速度增加，OLSR协议中的吞吐量有明显的下降趋势，而AODV则相对稳定。

**（5）不同CBR连接数下的仿真结果**

（a）delay (b) load

（c）ratio (d) throughput

图5 不同CBR连接数目下的仿真结果

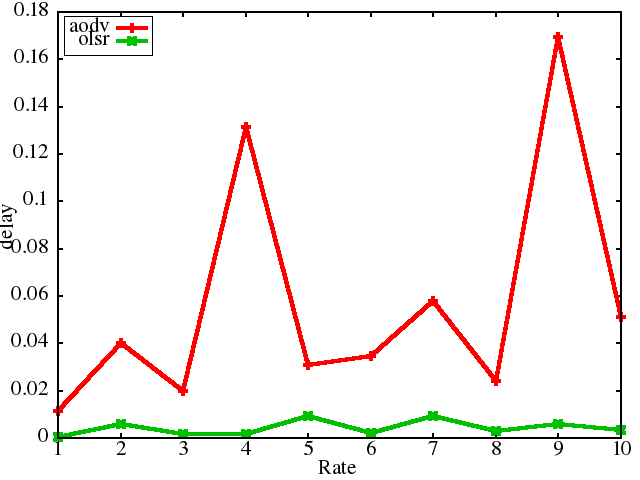
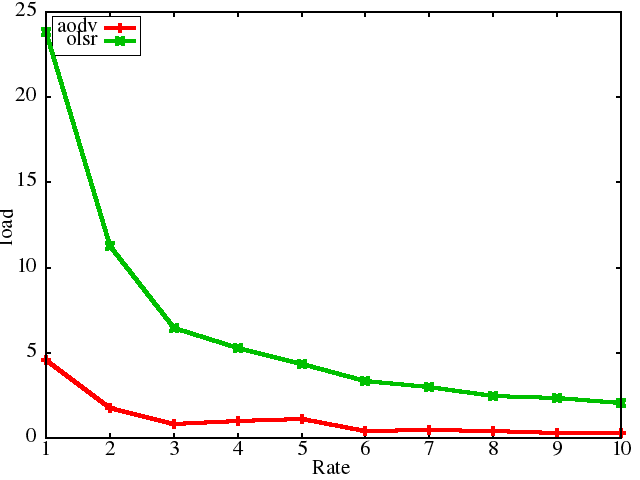
由图5（a）得到，随着CBR连接数增加，AODV路由协议的端到端延迟降低，而OLSR路由协议的端到端延迟变化不明显，这是因为AODV是一种反应式路由协议，而OLSR是一种主动式路由协议。

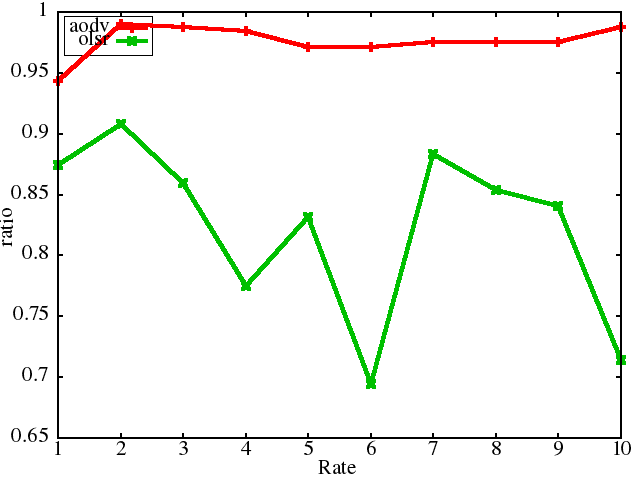
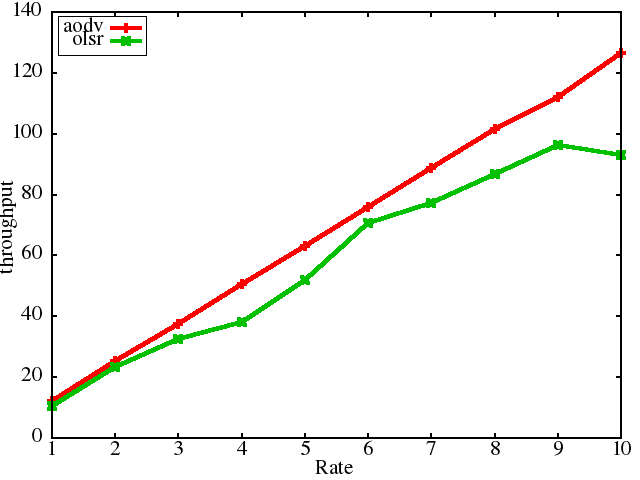
由图5（b）可以得到，随着CBR连接数增加，AODV协议的负载基本稳定，变化不大，因为AODV协议中，节点通过向其他节点发送路由请求来获取路由信息，即使连接数量增加，路由表的更新可能会导致新的路由路径出现，但这些路径通常是不同的，并且不会增加路由开销。而OLSR不同，OLSR 协议采用动态路由机制，每个节点都维护自己的路由表，并根据链路状态信息更新路由表，如果连接数量增加，邻居节点的数量也会增加，并且每个邻居节点可能需要处理更多的数据包，因此路由开销会降低。

图5（c）显示，随着CBR连接数的增加，AODV的分组投递率相对稳定，但是OLSR波动起伏较大且一直低于AODV，主要因为CBR连接数的增加会导致网络中数据包的增加，AODV基于节点间的距离来确定路由，而OLSR使用贪婪算法，数据通信量的增加会导致网络延迟甚者拥塞。

图5（d）显示，随着CBR连接数的增加，两种协议的网络吞吐量都持续上升，这是因为CBR连接数增加，导致单位时间内网络中的数据包也增加，故吞吐量增加。

**（6）不同发包速率下的仿真结果**

（a）delay (b) load 

（c）ratio (d) throughput

图6 不同发包速率下的仿真结果

由图6（a）看到，AODV路由协议的延迟高于OLSR协议的延迟，并且OLSR协议的延迟保持稳定。这是因为OLSR时先应式路由协议，意味着当一个数据包到达一个节点时，它可以立即被转发或丢弃，但是在AODV协议里，如果没有到达目的地的路由，在进行路由发现过程中，要达到目的地的分组被存在缓冲器里，这可能导致较长的延迟。

由图6（b）看到，随着发包速率增大，两种协议的路由开销都有所下降。相对于OLSR，AODV的路由开销更小，是因为当发包速率增加时，节点之间的通信会更加频繁，平均下来每发送一个数据包的路由信息降低，从而减少了路由开销。

由图6（c）得到，随着发包速率的增加， OLSR的分组投递率下降，这是因为节点总是选择最短路径作为默认路由，当网络拥塞时，节点之间的通信量增加，从而导致网络延迟增加，并且可能最终导致网络瘫痪。在这种情况下，节点无法正确地路由分组，从而导致分组丢失和分组投递率下降。而AODV 路由协议使用距离向量路由算法，该算法基于节点之间的距离来确定路由。因此，当数据发送速率增加时，节点之间的通信量会增加，但这些增加的通信量并不会增加节点之间的距离。这意味着 AODV 路由协议仍然能够正确地路由分组，并且分组投递率不会受到数据发送速率的影响。

由图6（d）可以看出，随着发包速率的增大，单位时间内通过网络的实际数量会有所上升，故两种协议的吞吐量都有所上升。

# 总结

本文介绍了AODV和OLSR两种路由协议，设置了六种实验场景对这两种协议的端到端延迟、分组投递率、归一化路由开销以及吞吐量等性能进行了模拟分析。通过以上实验结果，可以得出以下结论：从端到端延迟方面考虑，OLSR路由协议性能一般要优于AODV协议，但随着节点移动速度的提高和节点数据传输速度的提高，AODV路由协议的性能逐渐接近OLSR路由协议，并且可以预计得到，在一些因素的影响之下，AODV路由协议的性能有可能要优于OLSR路由协议。在归一化路由开销方面，AODV路由协议的性能一般优于OLSR，因为AODV是被动式路由协议，仅当有数据发送时才发送路由信息。在分组投递率方面，OLSR一般差于AODV，但随着节点停留时间的增加，与AODV协议的表现逐渐接近。总结来说，OLSR路由协议更适合于节点运动速度缓慢节点传输能量不高的无线自组网络，而AODV更适合于节点移动速度快、节点传输能量高的移动无线自组网络。