(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108989223 B
(45)授权公告日 2021.09.03

(21)申请号 201810608027.7

(22)申请日 2018.06.13

(65)同一申请的已公布的文献号
申请公布号 CN 108989223 A

(43)申请公布日 2018.12.11

(73)专利权人 昆宇蓝程(北京)科技有限责任公司
地址 100089 北京市海淀区安宁庄西路9号院29号楼1502、1503

(72)发明人 董华

(74)专利代理机构 北京高航知识产权代理有限公司 11530
代理人 赵永强

(51)Int.Cl.
H04L 12/751(2013.01)
H04L 12/727(2013.01)
H04B 7/185(2006.01)

(56)对比文件
CN 106656302 A,2017.05.10
CN 105227483 A,2016.01.06
CN 107733518 A,2018.02.23

审查员 段燕辉

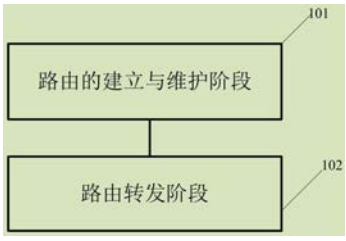
权利要求书2页 说明书12页 附图6页

(54)发明名称

一种强链路约束条件下的卫星路由方法

(57)摘要

本发明提供了一种强链路约束条件下的卫星路由方法,包括路由的建立与维护阶段和路由转发阶段,在路由的建立与维护阶段,每颗卫星根据卫星运行的周期性和网络拓扑的规律性,周期性地向邻居节点发送探测包,获取邻居卫星的地理位置信息,建立卫星节点的邻居表,并根据邻居表的信息初步计算生成多径路由表,该阶段会贯穿于卫星网络的整个运行过程;在路由转发阶段,卫星节点分别预估出该状态下每条路径的传输时延开销,选择传输时延开销最小的链路进行转发。有益效果:该路由算法通过逐跳地路径决策来解决节点内部、节点间的传输冲突,以适应于链路状态变化,保障数据路由传输的可靠性和网络性能的稳定性。



CN 108989223 B

1.一种强链路约束条件下的卫星路由方法，其特征在于，包括路由的建立与维护阶段101和路由转发阶段102，在路由的建立与维护阶段101，每颗卫星根据卫星运行的周期性和网络拓扑的规律性，周期性地向邻居节点发送探测包，获取邻居卫星的地理位置信息，建立卫星节点的邻居表，并根据邻居表的信息初步计算生成多径路由表，该阶段会贯穿于卫星网络的整个运行过程；在路由转发阶段102，卫星节点首先需要决策出下一跳节点，它会根据与目的节点的位置关系以及跨层搜集到的链路状态信息，分别预估出该状态下每条路径的传输时延开销，选择传输时延开销最小的链路进行转发；

所述路由方法通过候选路径计算模块、邻居信息维护模块、链路状态收集模块、路径决策模块实现，同时，每个节点需要维护3个信息表：邻居表、链路状态信息表、多径路由表；

在路由的建立与维护阶段，进行邻居信息维护、候选路径计算和链路状态收集，通过邻居信息维护模块独立运行并会贯穿运行于卫星网络的整个使用周期，每颗卫星周期性地向邻居节点发送1跳有效的探测包，获取邻居卫星的地理位置信息，维护自己的邻居表；链路状态收集模块用于收集并维护链路状态信息表，包括通过跨层感知的方式获取物理层天线状态信息和MAC层的负载信息，通过分析邻居信息探测包和应答包的时间戳信息来估计链路时延；候选路径计算模块用于维护多径路由表，它会根据当前卫星、邻居卫星、目的卫星三者的相对位置关系来确定下一跳候选节点集，并存储在多径路由表中，有效期内并不需要重新计算；

候选路径的计算采用以下方式进行：

每对卫星之间提供3条备选路径，因此每个目标节点对应于3个下一跳地址，其中路径长度最短的为首选下一跳，其余两个为备选下一跳；

源卫星的地址标识分别为$\langle p_s, s_s\rangle$和$[lon_s, lat_s]$，目的卫星的地址标识分别为$\langle p_d, s_d\rangle$和$[lon_d, lat_d]$，多径路由表的计算过程如下：

(1)如果目的卫星属于当前卫星的邻居卫星，则当前卫星只需依据邻居表中的位置信息转发数据；

(2)若$p_s=p_d$，即两颗卫星在同一个轨道平面，那么首选下一跳从源卫星的上、下两颗邻居中选择；每个轨道平面有N颗卫星，则上下邻居与目的卫星之间路径长度的计算公式为：

$dist(i)=\min\{N-|s_i-s_d|,|s_i-s_d|\}$

于是，确定出首选下一跳节点$\langle p_i, s_i\rangle$，它满足$i=\arg\min_{i\in\{1,3\}}\{dist(i)\}$

它对应的路径长度为$D_i=\min\{dist(1),dist(3)\}$；其余两个备选下一跳节点则对应于该卫星左右两颗邻居卫星，路径长度为$2+D_i$；

(3)若$s_s=s_d$，即两颗卫星在同一个纬度平面，那么首选下一跳从源卫星的左、右两颗邻居节点中选择；一共有M个轨道平面，则左右邻居与目的卫星之间路径长度的计算公式为：

$dist(i)=\min\{M-|p_i-p_d|,|p_i-p_d|\}$

于是，确定出首选下一跳节点$\langle p_i, s_i\rangle$，它满足$i=\arg\min_{i\in\{2,4\}}\{dist(i)\}$，它对应的路径长度为$D_i=\min\{dist(2),dist(4)\}$；其余两个备选下一跳节点则对应于该卫星上下两颗邻居卫星，路径长度为$2+D_i$；

(4)若$p_s\neq p_d$且$s_s\neq s_d$，即两颗卫星既不在同一轨道平面也不在同一维度平面，为了确定

下一跳候选节点集并进行排序，这里需要分别计算出4颗邻居节点到目的节点之间的距离，而不再是计算路径的跳数；邻居卫星与目的卫星之间星间距离的计算公式如下，计算时使用的不是虚拟地址，而是卫星节点在第二轨道坐标系中的坐标；

$$length(i)=\sqrt{(x_i-x_d)^2+(y_i-y_d)^2+(z_i-z_d)^2}$$

于是，根据星间距离的远近，确定出首选下一跳节点$\langle p_i, s_i\rangle$，它满足：

$$i=\arg\min_{i\in\{1,2,3,4\}}\{length(i)\}$$

邻居卫星与目的卫星之间路径长度的计算公式为：

$$dist(i)=\min\{M-|p_i-p_d|,|p_i-p_d|\}+\min\{N-|s_i-s_d|,|s_i-s_d|\}$$

通过上面的方法，每颗卫星计算出到目的卫星的首选、备选下一跳节点和对应的路径长度。

2.根据权利要求1所述的一种强链路约束条件下的卫星路由方法，其特征在于，在路由转发阶段，进行路径决策和链路调度；路径决策时，首选需要查询多径路由表获取或者重新计算出候选路径的信息，然后根据收集到的链路状态信息，分别计算出每条候选路径的时间效用，最后贪心地选择传输时延开销最小的链路作为决策链路。

一种强链路约束条件下的卫星路由方法

技术领域

[0001] 本发明涉及路由技术领域,特别是涉及一种强链路约束条件下的卫星路由方法。

背景技术

[0002] 卫星网络具有覆盖范围广且灵活的特点,可以实现全球网络的无缝覆盖,已经逐渐成为下一代互联网的基础部分。卫星网络自提出以来,路由技术一直是其研究热点。截至目前,国内外研究人员结合不同卫星网络的特点,已提出了许多代表性的星上路由解决方案。从发展阶段上来看,卫星网络路由技术经历了由单层卫星网络路由到多层卫星网络路由;再到多业务需求路由;其中,前两种都属于基于特定网络拓扑的路由,后一种是面向QoS保障的路由。从发展趋势来看,卫星网络路由技术经历了从传统的面向连接的路由发展为无连接的路由,寻求与Internet无缝对接的可扩展路由方式。从优化目标上来看,路由策略从基本的单目标、最短路径策略向多目标、多路径、自适应策略转变,综合考虑网络吞吐、负载均衡等整体性能;对QoS路由的优化,从单目标代价函数值最小化算法向启发式多目标优化算法演变。

[0003] ADOV协议是一种按需路由技术,我们在后面也对我们提出的路由算法和ADOV算法做了对比实验。

[0004] 按需距离矢量路由(Ad Hoc On Demand Distance Vector,AODV)协议是一种典型的按需路由协议。它只有在需要进行通信时才会建立路由,并且只对正在通信的节点路由信息进行维护。AODV路由协议有一个重要特征:在每个节点的每个路由条目中都包含一个计时器,如果经过一段时间一条路由条目没有被使用,就说明该路由条目已经过期。AODV协议借鉴了DSDV协议的思想,可以看成是DSDV协议和DSR协议两者的综合版。为了避免缓存的路由信息出现过期以及环路的情况,它使用了DSDV路由协议中的“目的节点序列号”,同时也避免了无穷计数的问题。路由建立与DSR协议的方法相似,但AODV协议采用逐跳路由而不在是源路由,这种处理机制能有效地避免路由出现闭环的现象。

[0005] 参见图11,AODV协议的路由过程具体为:当网络中需要一条路径进行通信传输数据分组时,如果通过查找发现不存在可用路径,此时发送分组的节点则广播一个RREQ分组给其所有的邻居节点。若接收到RREQ分组的节点中没有相应到达目的节点的路由时则更新自身路由表信息,并继续广播给其它所有邻居节点。反之,如果存在达到目的节点的路由条目,则将收到的RREQ序列号与本地路由信息的序列号进行对比,然后按如下两种操作处理:

[0006] (1)若新收到分组消息中的节点序列号与路由信息表中的序列号相比较大,则更新本地路由信息条目,并继续广播该RREQ分组至其他邻居节点。

[0007] (2)若收到分组消息中的序列号与路由信息表中的序列号相比较小,则向源节点发送包含本地路由信息的RREP分组。

[0008] AODV路由协议主要有以下几个特点:

[0009] (1)支持单、多播的形式,可同时向多个节点传输相同的数据分组。

[0010] (2)为防止出现路由环路以及路由信息过期,采用了路由序列号。

[0011] (3)能够适用于较大规模的无线网络。

[0012] (4)建立路由时采用RREQ/RREP询问方式。

[0013] (5)路由信息表包含源节点IP地址及其序列号、目的节点地址及其序列号。

[0014] 除上述特点外，AODV协议同时还存在一些缺点。AODV协议只适应于对称信道且路由建立时延大，对于实时性要求较高的网络往往会受到一定的限制。另外，AODV协议只支持双向链路。

[0015] 小卫星网络的星间链路资源极为宝贵，可能存在通信方向、链路数量，链路输出缓存空间，链路启动开销等多种约束条件，星间链路的建立和保持需要同时满足多种约束条件，因此实际网络中的星间链路并不是简单的双向全连通链路。现有的路由、调度算法，在设计时普遍采用了双向全连通的链路模型，并没有考虑实际链路约束。然而星间链路连通状态将会直接影响卫星网络路由、调度算法使用时的性能。

发明内容

[0016] 针对上述问题，本发明旨在提供一种强链路约束条件下的卫星路由方法。

[0017] 本发明的目的采用以下技术方案来实现：

[0018] 本发明提供了一种强链路约束条件下的卫星路由方法，包括路由的建立与维护阶段和路由转发阶段，在路由的建立与维护阶段，每颗卫星根据卫星运行的周期性和网络拓扑的规律性，周期性地向邻居节点发送探测包，获取邻居卫星的地理位置信息，建立卫星节点的邻居表，并根据邻居表的信息初步计算生成多径路由表，该阶段会贯穿于卫星网络的整个运行过程；在路由转发阶段，卫星节点首先需要决策出下一跳节点，它会根据与目的节点的位置关系以及跨层搜集到的链路状态信息，分别预估出该状态下每条路径的传输时延开销，选择传输时延开销最小的链路进行转发。

[0019] 有益效果：采用多径路由的思想，首先根据相对位置关系计算出候选路径集，然后根据收集到的最新链路状态信息分别计算每条路径的代价，以决策出适应当前链路状态的最优下一跳。该路由算法通过逐跳地路径决策来解决节点内部、节点间的传输冲突，以适应于链路状态变化，保障数据路由传输的可靠性和网络性能的稳定性。

附图说明

[0020] 利用附图对本发明作进一步说明，但附图中的实施例不构成对本发明的任何限制，对于本领域的普通技术人员，在不付出创造性劳动的前提下，还可以根据以下附图获得其它的附图。

[0021] 图1是本发明步骤示意图；

[0022] 图2是本发明功能实现框图；

[0023] 图3是本发明邻居卫星示意图；

[0024] 图4是本发明邻居表的建立、维护的过程；

[0025] 图5是本发明探测包、应答包格式；

[0026] 图6是本发明跨层示意图；

[0027] 图7是本发明基于链路状态的路径决策算法流程图；

[0028] 图8是本发明数据丢包率的仿真结果；

[0029] 图9是本发明在不同的单数据流发送速率下的网络平均吞吐量值；

[0030] 图10是不同数据发送速率下本发明路由算法、ELB、AODV和SNAPSHOT四种路由算法的平均端到端延迟对比

[0031] 图11是AODV协议的路由过程。

[0032] 附图标记：

[0033] 路由的建立与维护阶段101、路由转发阶段102。

具体实施方式

[0034] 结合以下实施例对本发明作进一步描述。

[0035] 下面将结合附图和实施例对本发明作进一步的详细说明。通过足够详细的描述这些实施示例，使得本领域技术人员能够理解和实践本发明。在不脱离本发明的主旨和范围的情况下，可以对实施做出逻辑的、实现的和其他的改变。因此，以下详细说明不应该被理解为限制意义，本发明的范围仅仅由权利要求来限定。

[0036] 参见图1，本实施例一种强链路约束条件下的卫星路由方法(Constrained ISLsStatus Routing)，包括路由的建立与维护阶段101和路由转发阶段102，在路由的建立与维护阶段101，每颗卫星根据卫星运行的周期性和网络拓扑的规律性，周期性地向邻居节点发送探测包，获取邻居卫星的地理位置信息，建立卫星节点的邻居表，并根据邻居表的信息初步计算生成多径路由表，该阶段会贯穿于卫星网络的整个运行过程；在路由转发阶段102，卫星节点首先需要决策出下一跳节点，它会根据与目的节点的位置关系以及跨层搜集到的链路状态信息，分别预估出该状态下每条路径的传输时延开销，选择传输时延开销最小的链路进行转发。这两个阶段在时间上相互重叠，功能上互为辅助，从而保证了路由的有效性。

[0037] 有益效果：采用多径路由的思想，首先根据相对位置关系计算出候选路径集，然后根据收集到的最新链路状态信息分别计算每条路径的代价，以决策出适应当前链路状态的最优下一跳。该路由算法通过逐跳地路径决策来解决节点内部、节点间的传输冲突，以适应于链路状态变化，保障数据路由传输的可靠性和网络性能的稳定性。

[0038] 参见图2，所述路由算法通过候选路径计算模块、邻居信息维护模块、链路状态收集模块、路径决策模块实现，同时，每个节点需要维护3个信息表：邻居表、链路状态信息表、多径路由表；

[0039] 在路由的建立与维护阶段，主要进行邻居信息维护、候选路径计算和链路状态收集，通过邻居信息维护模块独立运行并会贯穿运行于卫星网络的整个使用周期，每颗卫星周期性地向邻居节点发送1跳有效的探测包，获取邻居卫星的地理位置信息，维护自己的邻居表。链路状态模块用于收集并维护链路状态信息表，主要包括通过跨层感知的方式获取物理层天线状态信息和MAC层的负载信息，通过分析邻居信息探测包和应答包的时间戳信息来估计链路时延。候选路径计算模块主要用户维护多径路由表，它会根据当前卫星、邻居卫星、目的卫星三者的相对位置关系来确定下一跳候选节点集，并存储在多径路由表中，有效期内并不需要重新计算。

[0040] 在路由转发阶段，主要进行路径决策和链路调度。路径决策时，首选需要查询多径路由表获取或者重新计算出候选路径的信息(下一跳地址和路径长度)，然后根据收集到的

链路状态信息，分别计算出每条候选路径的时间效用，最后贪心地选择传输时延开销最小的链路作为决策链路。

[0041] 路由建立与维护阶段、路由转发阶段在时间上相互重叠，功能上互为辅助，从而保证了路由的有效性。使用基于该路由算法的节点在收到数据传输请求时，首先会分析该数据包是探测包还是普通数据包，如果它探测包，节点会更新自己的邻居信息，并直接沿原路径发送生存期为一跳的应答包，并不需要路由；如果是它是普通的数据包，需要判断目标卫星和自己的位置关系，如果目标卫星是自己的邻居，则查询邻居位置信息后发送即可；否则，需要调用路径决策模块，根据链路状态(天线状态、链路负载、链路维持时间、建链时延)和候选路径信息(下一跳、路径长度)进行路径决策。最后，链路调度模块将根据链路决策的结果将数据包插入链路缓存队列中的合适位置。

[0042] 该路由算法的路由转发过程可用表1描述：

[0043]

输入	候选路径信息，链路状态信息，邻居信息
输出	决策链路
1	**if** 检测到数据传输需求
2	**if** 包类型为探测包 / 应答包 查询邻居位置信息； 决策链路为邻居节点对应的输出链路。
3	**else if** 包类型为普通数据包 **if** 目标节点为邻居节点 • 查表获取邻居位置信息 • 决策链路为邻居节点对应的输出链路 **else** • 查表/计算下一跳候选节点集 • 获取节点链路状态信息 • 链路决策模块计算候选路径效用，输出决策链路
4	将决策结果和数据包发交给链路调度模块 链路调度模块将分组插入链路缓存队列中的合适位置
5	返回第一步

[0044] 表1

[0045] 该路由算法在设计时，考虑链路资源非常有限，为了避免由频繁的控制包引发的网络拥塞，在路由的过程中取消了中间节点的应答机制，只进行端到端的重传确认。因此，该路由算法进行传输控制时只考虑了端到端的连接管理和重传确认机制。此外，该路由算法由于使用了多径策略，在进行路由决策，必须避免路由环路的发生。本文采用的方法是在数据包的头部增加一定比特，用于对历史路径进行编码存储。

[0046] 路由表的建立与维护具体方法：

[0047] 为了使每个节点能建立和维护多条到达目的节点的路径，需要确定从每颗卫星到达目的卫星的多条路径对应的下一跳节点集H。

[0048] 该路由算法利用源节点和目的节点之间地理位置相对关系来进行多径路由的计算，它使用基于虚拟节点的策略以屏蔽卫星的相对运动，通过无连接的路由方式消除了路

由算法对于拓扑结构的依赖性。每颗卫星可以通过GPS装置获取自己的经纬度信息，作为自己在卫星网络中的物理地址，用[lon_s,lat_s]表示；同时，该路由算法根据经纬度信息来对地球表进行区域划分，整个地球可以划分为一个个逻辑分区，每个逻辑分区由最近的卫星覆盖，因此每颗卫星还会有一个逻辑地址对应于它覆盖的逻辑区域，用＜p,s＞表示，其中p为分区轨道平面号，s为轨道内卫星号，可以看出当卫星覆盖区域改变后，它的逻辑地址也会改变。至此，卫星网络可以抽象成一个Grid型拓扑，其中每个节点同时拥有逻辑地址和物理地址两种位置属性作为其地址标识。

[0049] 每颗卫星最多可以与其周围4颗邻居卫星建立星间链路，在路由的建立与维护阶段，每颗卫星周期性地向邻居节点发送探测包，以便获知其上、下、左、右4颗直连邻居卫星的(逻辑、物理)地址标识，维护自己的邻居表，用作路由计算时的参照。5颗卫星的上下左右的相对关系是由卫星的移动方向确立的，具体参见图3，邻居表的格式参见表2。

[0050]

邻居 ID	逻辑地址	物理地址	更新时间
1	$<p_1,s_1>$	$[lon_1,lat_1]$	t_1
2	$<p_2,s_2>$	$[lon_2,lat_2]$	t_2
3	$<p_3,s_3>$	$[lon_3,lat_3]$	t_3
4	$<p_4,s_4>$	$[lon_4,lat_4]$	t_4

[0051] 表2

[0052] 参见图4，邻居表的建立、维护的过程为：

[0053] (1)每颗卫星在自己的轨道周期内，根据预先设定好的探测时间间隔，周期性地向四周广播生存期只有1跳邻居节点探测包(包含自身地址标识和时间戳)。

[0054] (2)邻居卫星接到探测包后，首先判断探测包是否有效，如果探测包有效，则沿接收的方向将包含自己位置信息的应答包发送给源卫星。探测包和应答包格式参见图5。

[0055] (3)源卫星根据收到有效的应答包后，更新自己邻居表信息。

[0056] (4)如果卫星在自己的轨道周期内没有探测到邻居卫星的信息，即邻居表为空，该卫星会一直处于广播探测状态。

[0057] 候选路径的计算采用以下方式进行：

[0058] 该路由算法通过源、目的两颗卫星的相对地理位置关系，从邻居节点中确定出下一跳候选节点集H和对应的路径长度D，以提供从源卫星到目的卫星的多条路径信息，用于路径决策。该路由算法为网络中每对卫星之间提供3条备选路径，因此每个目标节点将对应于3个下一跳地址，其中路径长度最短的为首选下一跳，其余两个为备选下一跳。假设源卫星的地址标识分别为<p_s,s_s>和[lon_s,lat_s]，目的卫星的地址标识分别为<p_d,s_d>和[lon_d,lat_d]，多径路由表的计算过程如下。

[0059] (1)如果目的卫星属于当前卫星的邻居卫星，则当前卫星只需依据邻居表中的位置信息转发数据即可。

[0060] (2)若$p_s=p_d$，即两颗卫星在同一个轨道平面，那么首选下一跳应该从源卫星的上、下两颗邻居中选择。假设每个轨道平面有N颗卫星，则上下邻居与目的卫星之间路径长度的计算公式为：

[0061] $dist(i)=\min\{N-|s_i-s_d|,|s_i-s_d|\}$

[0062] 于是，可以确定出首选下一跳节点$<p_i,s_i>$，它满足$i=\arg\min_{i\in\{1,3\}}\{dist(i)\}$，它对应的路径长度(跳数)为$D_i=\min\{dist(1)$。其余两个备选下一跳节点则对应于该卫星左右两颗邻居卫星，路径长度为$2+D_i$。

[0063] (3)若$s_s=s_d$，即两颗卫星在同一个纬度平面，那么首选下一跳应该从源卫星的左、右两颗邻居节点中选择。假设一共有M个轨道平面，则左右邻居与目的卫星之间路径长度的计算公式为：

[0064] $dist(i)=\min\{M-|p_i-p_d|,|p_i-p_d|\}$

[0065] 于是，可以确定出首选下一跳节点$<p_i,s_i>$，它满足$i=\arg\min_{i\in\{2,4\}}\{dist(i)\}$，它对应的路径长度(跳数)为$D_i=\min\{dist(2),dist(4)\}$。其余两个备选下一跳节点则对应于该卫星上下两颗邻居卫星，路径长度为$2+D_i$。

[0066] (4)若$p_s\neq p_d$且$s_s\neq s_d$，即两颗卫星既不在同一轨道平面也不在同一维度平面，为了确定下一跳候选节点集并进行排序，这里需要分别计算出4颗邻居节点到目的节点之间的距离，而不再是计算路径的跳数。邻居卫星与目的卫星之间星间距离的计算公式如下，计算时使用的不是虚拟地址，而是卫星节点在第二轨道坐标系中的坐标。

[0067] $$length(i)=\sqrt{(x_i-x_d)^2+(y_i-y_d)^2+(z_i-z_d)^2}$$

[0068] 于是，根据星间距离的远近，可以确定出首选下一跳节点$<p_i,s_i>$，它满足：

[0069] $$i=\arg\min_{i\in\{1,2,3,4\}}\{length(i)\}$$

[0070] 邻居卫星与目的卫星之间路径长度的计算公式为：

[0071] $dist(i)=\min\{M-|p_i-p_d|,|p_i-p_d|\}+\min\{N-|s_i-s_d|,|s_i-s_d|\}$

[0072] 通过上面的方法，每颗卫星可以方便地计算出到目的卫星的首选、备选下一跳节点和对应的路径长度。可以看出，该路由算法并没有维护全局拓扑信息，每颗卫星只是维护了邻居信息，根据邻居信息和收集的链路状态用来路径决策。这样消除了路由算法对于拓扑结构的依赖性，同时减少了卫星之间控制信息的交换，为小卫星网络节约了链路资源。

[0073] 路由表的结构参见表3，每个路由表表项由目的卫星的逻辑地址唯一检索，依次包含了首选路径、备选1、备选2的三项路由信息。这样，只需一次检索便能获得所有候选路径的路由信息。需要注意的是，路由信息中存储的是路径差Δx(即x_{n2}和x_{n3})，而非路径跳数。路径差的定义为每条路径和首选路径之间的跳数差。实际存储时为了节约空间，将每条路径的下一跳地址和路径差对应的二进制编码拼接成一个32位整型字段存储。

[0074]

目的地址标识	下一跳|跳数	备选1|跳数	备选2|跳数	时间戳
卫星1	$S_{11}\|0$	$S_{12}\|x_{12}$	$S_{13}\|x_{13}$	t_1
卫星2	$S_{21}\|0$	$S_{22}\|x_{22}$	$S_{23}\|x_{23}$	t_2
……		……		
卫星n	$S_{n1}\|0$	$S_{n2}\|x_{n2}$	$S_{n3}\|x_{n3}$	t_n

[0075] 表3

[0076] 候选路径效用的计算采用以下方法：

[0077] 小卫星网络链路资源极为有限，为了避免给网络增加额外的负担，该路由算法在路由转发时并不会同时使用多条备选路径发送分组，而是从备选路径中根据特定需求进行决策选择一条最优的路径进行路由转发。因此，该路由算法的路由选择过程是逐跳地决策出下一跳节点(输出链路)的过程，本质上属于一种序列决策行为。于是，该路由算法这种逐跳的路由选择过程可以看成是一个有限阶段的马氏决策模型(Markov Finite DecisionModel)。

[0078] 考虑到每一次路径决策的效果(Effect)受到多种链路约束的共同影响，而不应该只是单一的考虑路径长度。因此，根据链路约束对路由性能影响的分析，计算决策行为的报酬值时，需要综合考虑链路负载、天线工作状态、链路剩余存活时间、路径长度等多个参数的共同约束。为了将不同参数对决策效果的影响进行综合，统一将每个参数对决策效果的影响都转换成对路径传输时延的影响。于是，把决策因某个参数所增加的额外的传输时间，看成由该参数给决策造成的额外的开销(Cost)；反之，如果因某个参数可以缩短传输时延，将它看成该参数给决策带来的回报(Reward)。一个决策行为的效果E(s,a)是综合了所有链路参数带来的时延回报和所造成的时延开销的加权值。本发明考虑的数据传输时延包括分组处理时延t_d，链路建链时延t_s，数据传播时延t_h，分组排队时延t_q。

[0079] 假设某颗卫星sat_i收到一个目的地址不是自己的新分组，需要对其进行路由。通过查询路由表，获取到的路由下一跳候选节点集H＝{a_1,a_2,a_3}以及对应的路径差X＝{Δx1，Δx2，Δx3}。路由协议通过跨层搜集的方法获取到的候选节点集对应的链路状态集{L_j(t)}(j＝1,2,3)，链路状态L_j(t)包括链路分组数目n_j(t)，(n_j(t)∈{0,1,...,C}，C为链路容量)、天线工作状态ant_j(t)、状态维持时间$life_j$(t)等。

[0080] 其中，天线的工作状态包括发送、接收、空闲三种状态，表示为如下：

[0081]
$$ant_j(t)=\begin{cases}1, \text{发送}\\ 0, \text{空闲}\\ -1, \text{接收}\end{cases}$$

[0082] 下面，需要依次计算每条路径的E(s,a_i)。

[0083] 路径差对传输时延的影响，路径长度每增加1跳，相应会增加1个额外的传播时延

t_h和建链时延t_s，并且路径越长，传输时延的波动性越大，这里引入衰减因子γ（γ＜1)来刻画路径时延的波动性，因此E(Δx,a)可以使用下式计算。

[0084] $E(\Delta x,a)=\Delta x\cdot(t_h+t_s)+(\Delta x-1)\cdot(t_h+t_s)\cdot\gamma+L+(t_h+t_s)\cdot\gamma^{\Delta x}$

[0085] 一条链路，输出缓存队列长度越长，对于新到达的分组来说，它需要排队等待发送的时间就越长，因此链路负载会增加分组的传输时延，即$E(n,a)=n_i(t)$。当链路缓存已满时，应该直接将其开销置为100，此链路无效。

[0086] 对于某候选路径a_i，如果链路天线ant_i已经处于发送状态，则选择该链路发送分组将会节省链路重新启动需要的时间t_s以及一跳的传输时延t_h，如果链路天线处于接收、或者空闲，则选择该链路并不能带来传输时延上的回报，因此$(ant,a)=-ant_i(t)\cdot(t_s+t_d)$。

[0087] 状态维持时间长的链路相比剩余时间存活时间短的链路，其传输性能更好，因此：

[0088] $E(life,a)=-ant_i(t)\cdot life_i(t)$

[0089] 最后，综合所有的链路状态参数，计算出每条候选路径的E(s,ai)：

[0090] $E(s,a_i)=E(n_i,a_i)+E(ant_i,a_i)+E(\Delta x_i,a_i)+E(life_i,a_i)$进行路径决策时，选择时间开销最小的下一跳(NH)：

[0091] $NH=Min\{E(s,a_1),E(s,a_2),E(s,a_3)\}$

[0092] 链路状态的获取采用以下方法：

[0093] 根据上述内容可知，节点会根据所获取的链路状态信息，计算出每条候选路径的效用E(s,ai)以进行路径决策；因此有效的链路状态信息作为路径决策算法的输入之一，成为该路由算法得以有效应用的前提条件。该路由算法进行路径决策时所考虑的链路状态主要包括链路负载情况、天线模式(半双工、全双工)、天线工作状态(空闲、接收、发送)、链路剩余存活时间、链路时延参数(传播时延、建链时延等)。使用该路由算法的每个节点需要额外维护一张链路状态信息表，参见表4，与其邻居表相对应，以存储获取到的最新的链路状态信息。

[0094]

邻居ID	链路负载	天线状态	有效期	传播时延	建链时延	更新时间
1	n_1	收	t_{l1}	t_{p1}	t_{s1}	t_1
2	n_2	收	t_{l2}	t_{p2}	t_{s2}	t_2
3	n_3	收	t_{l3}	t_{p3}	t_{s3}	t_3
4	n_4	发	t_{l4}	t_{p4}	t_{s4}	t_4

[0095] 表4

[0096] 参见图6该路由算法跨层示意图，为了获取到链路负载和天线状态这类来自MAC层、物理层的信息，路由协议需要通过“跨层搜集”的方法间接感知并获取这些低层参数。该路由算法设计时使用了外在跨层设计的方法，以完成对物理层、MAC层、网络层相关信息的

跨层设计。通过在协议结构中加入了一个中间层，物理层与网络层之间的交互，将不需要经过MAC层。通过这样的设计，物理层的天线状态信息、MAC层的链路负载信息可以直接反馈给网络层路由协议，用于计算候选链路的效用。由于路径决策时只使用了节点内部局部的链路状态信息，因此只需通过跨层感知的方法即可满足需求，并不需要通过广播洪泛包来获取全局的链路状态信息，一定程度上节约了网络的信令开销。

[0097] 对于星间链路的时延参数，随着卫星不断的运动，相邻两颗卫星之间的星间距离大小也可能不断变化，对应的链路传播时延也会不断变化。为了节省网络控制包的开销，该路由算法会在邻居探测的过程中捎带获取链路时延参数，并不需要进行额外的探测。具体方法是，每个节点将最新收到的探测包(或者应答包)中时间戳与当前时刻之间的差值作为最新的链路传播时延的估值并记录，建链时延的估算方法已经在第二章中给出。

[0098] 基于链路状态的路径决策算法：

[0099] 参见图7，给出了基于链路状态的路径决策过程。该路径决策算法的输入是前节点的链路状态信息(天线状态、链路负载、链路维持时间、建链时延)和目标节点对应的候选路径信息；算法的输出是经过一系列计算后得到的决策路径。作为算法的输入，候选路径信息可以通过查询多径路由表计算或者利用当前节点与目的节点的位置关系重新计算；链路状态信息的收集方法将在下一小节中给出。

[0100] 当需要进行路径决策时，节点首先检测当前的链路状态信息，如果所有的链路都空闲或者在接收，直接选择首选下一跳对应的路径作为决策路径，不再需要额外的计算；否则，需要按照上一节中的路径效用计算方法，分别计算出每条候选路径的效用，节点会基于贪心的原理选择效用最小的路径作为决策路径。

[0101] 基于链路状态的路径决策算法将收集到的最新的链路状态信息作为输入，因而其所作的决策可以适应链路状态的变化，起到减少由链路冲突所造成的阻塞，均衡节点间的负载，改善传输性能的作用。

[0102] 为了验证该路由算法的性能，在仿真软件Qualnet中搭建星间链路模型、并添加实现该方法的路由协议，以对其进行仿真和分析。仿真时使用铱星星座，卫星经过极区(纬度大于70°)时，关闭轨间ISLs，主要卫星网络拓扑参数参见表4所示。数据包长度设置为1000Bytes，链路缓存链路队列采用FIFO队列，队列缓存容量为50KB，即最多可以存放50个数据包。为减少错误丢包对路由算法性能分析的影响，仿真时将链路误码率设置为零。仿真时的业务源模型为CBR业务源，业务源的分布符合均匀通信的模型。所有的星间链路均为半双工链路，并且一个卫星一次只能有一个链路处于发送状态，建链时延设置为2倍的传播时延。

[0103]

卫星数	轨道类型	高度	轨道平面倾角	轨道平面数	轨内链路数	轨间链路数	反向缝链路
66	极轨	780k	86°	6	2	2	无

[0104]

m

[0104] 表5

[0105] 仿真时分别使用基于Dijkstra的快照路由(SNAPSHOT)、按需距离向量路由(AODV),具有抗拥塞功能的显示负载均衡路由(ELB)作为该路由算法的比较对象。由于卫星轨道周期为1小时40分,2小时仿真时间能够反应整个网络变化过程对协议影响,所以除特殊说明,仿真时间持续2小时,所有的实验独立重复3次。

[0106] 数据丢包率是指丢失的数据包占总传输数据包的比例,令s_n表示所有节点总的发送数据包数量,l_n表示所有节点丢失数据包的总量,则数据丢包率的计算公式如下:

[0107] $$r_{loss}=\frac{l_n}{s_n}$$

[0108] 数据丢包率测试结果如图8所示:可以看出,当数据发送速率较低,随着发送速率的上升,4种路由算法的丢包率普遍上升平缓。此时,该路由算法、ELB和SNAPSHOT算法的丢包率均稳定在12%左右并且上升非常缓慢,相比之下AODV路由算法的丢包率一直在缓慢地线性增长。

[0109] 当单数据流发送速率达到0.05Mbps后,4种路由算法的丢包率均开始出现明显的上升趋势,说明这时候由于大量数据包积压,网络已经开始拥塞,造成数据包的丢失。在系统出现拥塞的情况下,AODV路由算法的丢包率急剧上升,最高已经达到近60%,这时SNAPSHOP的丢包率上升至40%,ELB的丢包率上升至35%左右,该路由算法的丢包率上升至28%左右。

[0110] 上述时延结果表明在链路资源紧缺的小卫星网络中,网络非常容易发生拥塞,丢包问题相对严重并且无法避免。AODV路由算法作为反应式路由,由于路由建立的开销较大,并且缺乏负载均衡功能导致丢包非常严重;快照路由的丢包性能虽然好于AODV,但是由于它是单径路由,没有考虑到链路负载,灵活性差,数据流发送速率大于0.05Mbps后误码率也迅速上升;相比之下,ELB路由和该路由算法的丢包率均较低。该路由算法相比于ELB路由,不仅考虑了链路负载,还可以适应于链路工作状态进行选路,因此可以一直保持较低的丢包率。

[0111] 参见图9在不同的单数据流发送速率下的网络平均吞吐量值,该吞吐量计算时使用的是接收方单位时间内收到数据量的平均值。可以看出发送速率达到0.05Mbps后,系统的吞吐量基本已经达到瓶颈。数据发送速率大于0.05Mbps后,如果使用AODV路由,系统的吞吐量不仅不会上升,反而会有所下降,说明网络发生了严重的拥塞导致丢包严重;相比之下使用该路由算法、ELB和SNAPSHOT路由算法能使网络吞吐量趋于稳定。相比ELB和SNAPSHOT两种路由,该路由算法不仅在低数据发送速率下能获得最好的吞吐性能,达到网络吞吐量瓶颈时对应的数据发送速率也最大,此时该路由算法的吞吐量饱和值比ELB的饱和值高20%左右,比SNAPSHOT的饱和值高30%左右。因此,该路由算法相较于其它算法能保证系统更好的吞吐性能。

[0112] 参见图10不同数据发送速率下该路由算法、ELB、AODV和SNAPSHOT四种路由算法的平均端到端延迟对比。可以看出,发送速率较低时,系统负载情况良好,4种算法的端到端延迟均较低,这时SNAPSHOT、ELB和该路由算法的时延性能相当,并稍低于AODV的时延性能。但是当数据发送速率增加达到0.06Mbps时,由于系统发送拥塞,SNAPSHOT、ELB和该路由算法的传输时延都开始明显增加,其中SNAPSHOT路由的时延增长最快,ELB路由的次之,该路由

算法的时延增长最慢，仅为SNAPSHOT传输时延的35％左右，ELB传输时延的65％左右。值得注意的是，数据发送速率较高时，虽然使用AODV路由得到的传输时延最低，但是根据前面的分析可知这时系统的丢包和吞吐性能非常差，因此即使AODV能获得低时延也不具有实际应用价值。

[0113] 同现有技术相比，该路由算法具备以下特点：

[0114] 1、每个节点独立地进行逐跳地分布式按需路由，以适应链路状态变化和网络拓扑的动态性，从而避免了集中式路由存在的对拓扑的依赖性以及重路由问题，提高了路由算法的鲁棒性。

[0115] 2.采用多径路由的思路，路由表中每个目标节点对应于多个下一跳节点，使源节点和目的节点之间存在多条候选路径，它可以通过快速地重路由来应对链路状态的动态变化；在对多条路径进行决策时，该算法通过预估每条路径传输时延开销，选择时延开销最小的路径进行传输。

[0116] 3.为了使路由协议能够感知链路状态的变化，采用跨层设计的方法，使上层路由协议感知底层协议的信息，并进行自适应的按需路由。

[0117] 4.跨层设计时，通过在协议结构中加入了一个中间层，物理层与网络层之间的交互，将不需要经过MAC层

[0118] 4.将链路约束纳入路由算法的考虑范围，链路不再是简单的全双工模型。

[0119] 5.每颗卫星可以通过GPS装置获取自己的经纬度信息，作为自己在卫星网络中的物理地址，用$[lon_s, lat_s]$表示。同时，该路由算法根据经纬度信息来对地球表进行区域划分，整个地球可以划分为一个个逻辑分区，每个逻辑分区由最近的卫星覆盖，因此每颗卫星还会有一个逻辑地址对应于它覆盖的逻辑区域，用＜p,s＞表示，其中p为分区轨道平面号，s为轨道内卫星号，可以看出当卫星覆盖区域改变后，它的逻辑地址也会改变。

[0120] 6.候选路径的计算。

[0121] 7.该路由算法的路由选择过程是逐跳地决策出下一跳节点(输出链路)的过程，本质上属于一种序列决策行为。

[0122] 8.考虑到每一次路径决策的效果(Effect)受到多种链路约束的共同影响，而不应该只是单一的考虑路径长度。因此，根据链路约束对路由性能影响的分析，计算决策行为的报酬值时，需要综合考虑链路负载、天线工作状态、链路剩余存活时间、路径长度等多个参数的共同约束。本文中，为了将不同参数对决策效果的影响进行综合，统一将每个参数对决策效果的影响都转换成对路径传输时延的影响。

[0123] 9.其中，路径差对时延的影响：

[0124] $E(\Delta x,a)=\Delta x\cdot(t_h+t_s)+(\Delta x-1)\cdot(t_h+t_s)\cdot\gamma+L+(t_h+t_s)\cdot\gamma^{\Delta x}$.

[0125] 链路负载对传输时延的影响：

[0126] $E(n,a)=n_i(t)\cdot t_d$。

[0127] 天线状态对时延的影响：

[0128] $E(ant,a)=-ant_i(t)\cdot(t_s+t_d)$。

[0129] 链路存活时间对时延的影响：

[0130] $E(life,a)=-ant_i(t)\cdot life_i(t)$

[0131] 10.该路由算法在设计时，考虑链路资源非常有限，为了避免由频繁的控制包引发

的网络拥塞，在路由的过程中取消了中间节点的应答机制，只进行端到端的重传确认。

[0132] 11.当需要进行路径决策时，节点首先检测当前的链路状态信息，如果所有的链路都空闲或者在接收，直接选择首选下一跳对应的路径作为决策路径，不再需要额外的计算；否则，需要按照上一节中的路径效用计算方法，分别计算出每条候选路径的效用，节点会基于贪心的原理选择效用最小的路径作为决策路径。

[0133] 链路状态获取模块中，路由协议需要通过“跨层搜集”的方法间接感知并获取这些低层参数。该路由算法使用的是外在跨层设计，但内在跨层设计同样可以达到获取底层参数的目的。内在跨层设计，协议栈各层之间并没有信息交互，只是在协议设计阶段考虑各层之间的信息交互。这种方法因为没有改变现有的协议栈结构，容易实现。

[0134] 我们采取的多径路由协议，是相对于传统的单径路由协议提出的，单径路由协议中源、目的节点之间往往只有一条路由；在多径路由协议中，会在源、目的节点之间同时提供多条路径，并确保路径信息的有效性。作为对单径路由的扩展，多径路由相比于单一路径的路由，有利于保证网络的健壮性，并且可以为网络流量提供负载均衡。但是单径路由同样可以实现本发明。

[0135] 最后应当说明的是，以上实施例仅用以说明本发明的技术方案，而非对本发明保护范围的限制，尽管参照较佳实施例对本发明作了详细地说明，本领域的普通技术人员应当理解，可以对本发明的技术方案进行修改或者等同替换，而不脱离本发明技术方案的实质和范围。

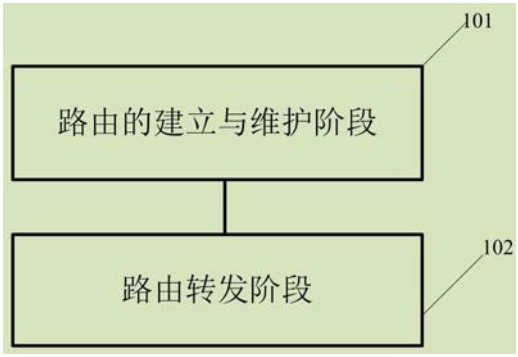

101
路由的建立与维护阶段
102
路由转发阶段

图1

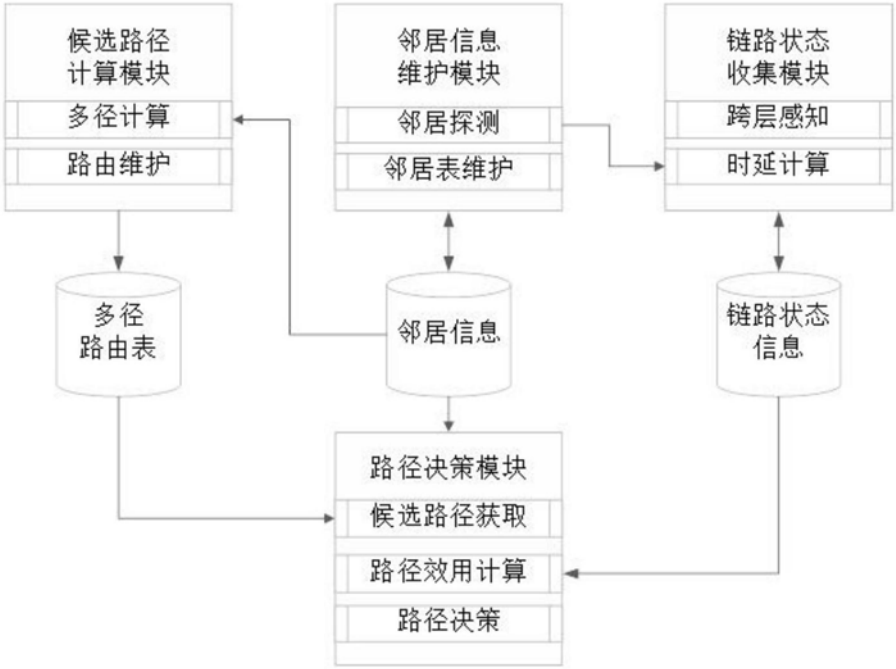

候选路径
计算模块
多径计算
路由维护
邻居信息
维护模块
邻居探测
邻居表维护
链路状态
收集模块
跨层感知
时延计算
多径
路由表
邻居信息
链路状态
信息
路径决策模块
候选路径获取
路径效用计算
路径决策

图2

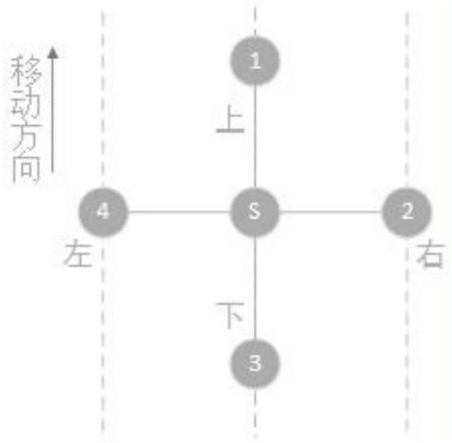

移动方向
1
上
4
S
2
左
右
下
3

图3

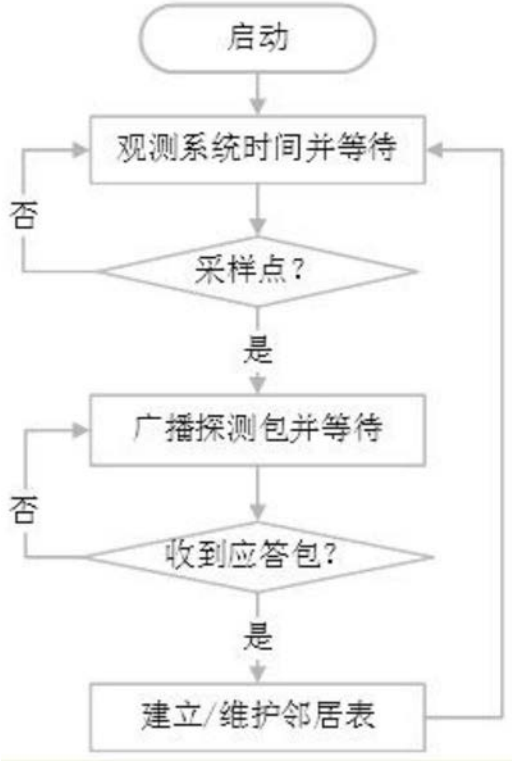

启动
观测系统时间并等待
否
采样点？
是
广播探测包并等待
否
收到应答包？
是
建立/维护邻居表

图4

探测包格式

包类型	源地址	广播地址	生存期	时间戳

应答包格式

包类型	源地址	目的地址	生存期	时间戳

图5

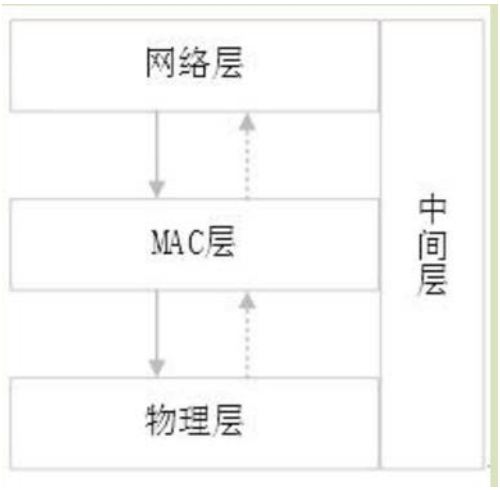


图6

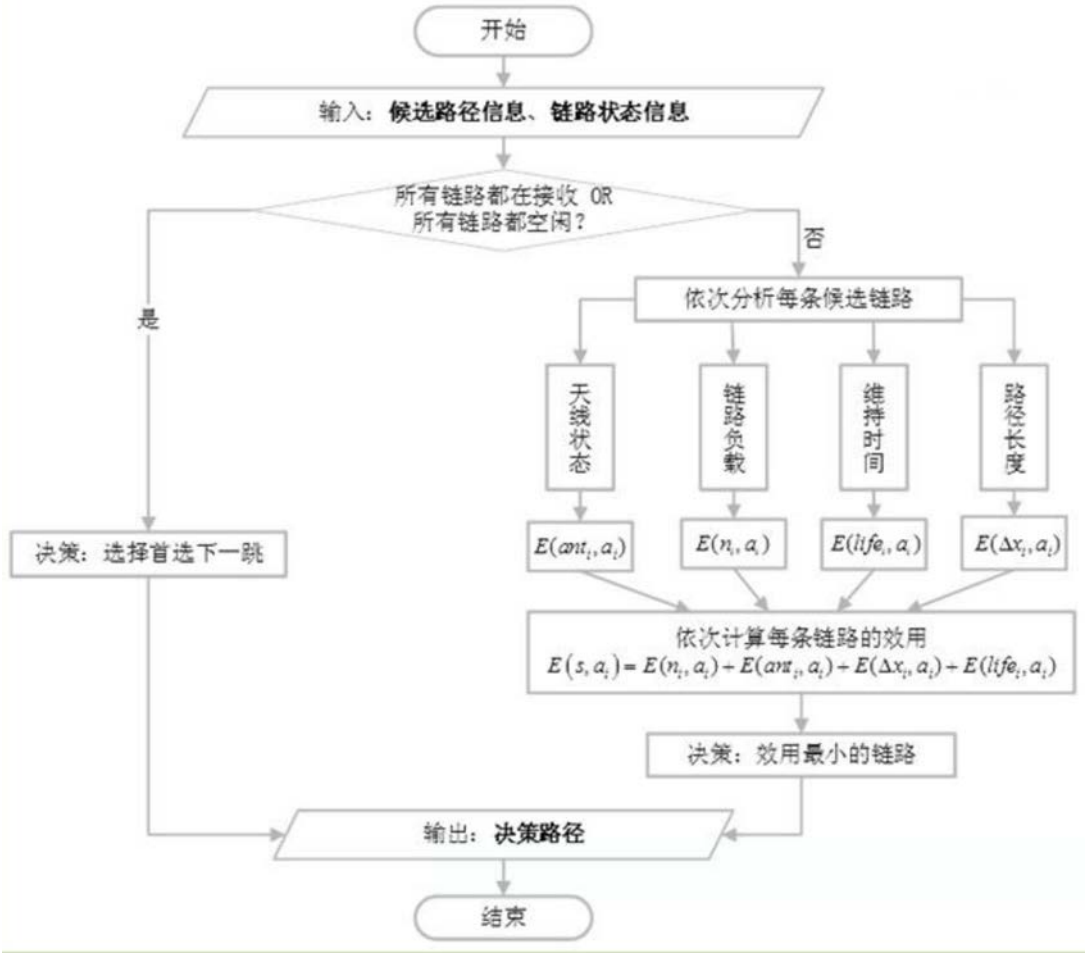

开始
输入：候选路径信息、链路状态信息
所有链路都在接收 OR
所有链路都空闲？
否
是
依次分析每条候选链路
天线状态
链路负载
维持时间
路径长度
$E(ant_i, a_i)$
$E(n_i, a_i)$
$E(life_i, a_i)$
$E(\Delta x_i, a_i)$
决策：选择首选下一跳
依次计算每条链路的效用
$E(s, a_i) = E(n_i, a_i) - E(ant_i, a_i) - E(\Delta x_i, a_i) - E(life_i, a_i)$
决策：效用最小的链路
输出：决策路径
结束

图7

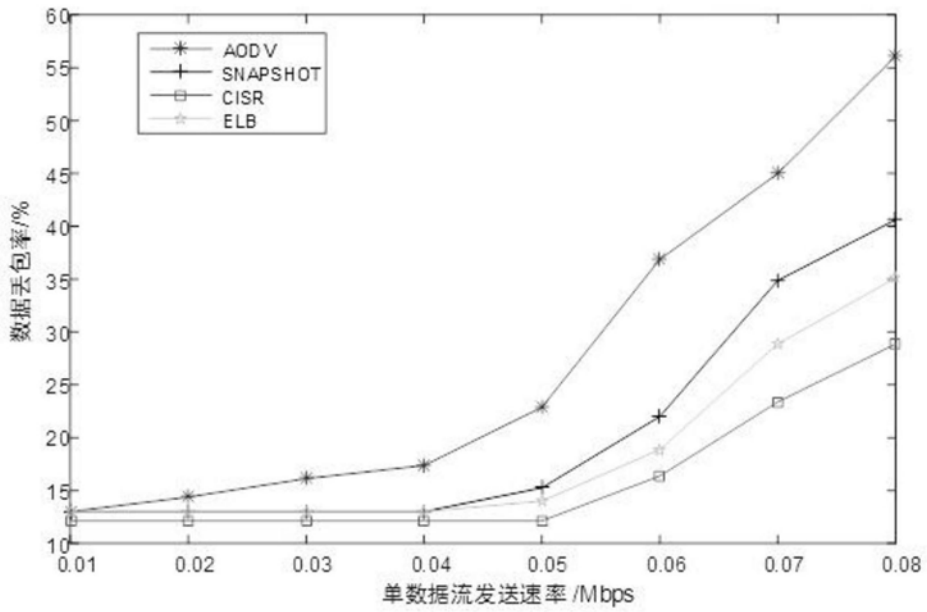

AODV
SNAPSHOT
CISR
ELB
数据丢包率/%
单数据流发送速率 /Mbps
60
55
50
45
40
35
30
25
20
15
10
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08

图8

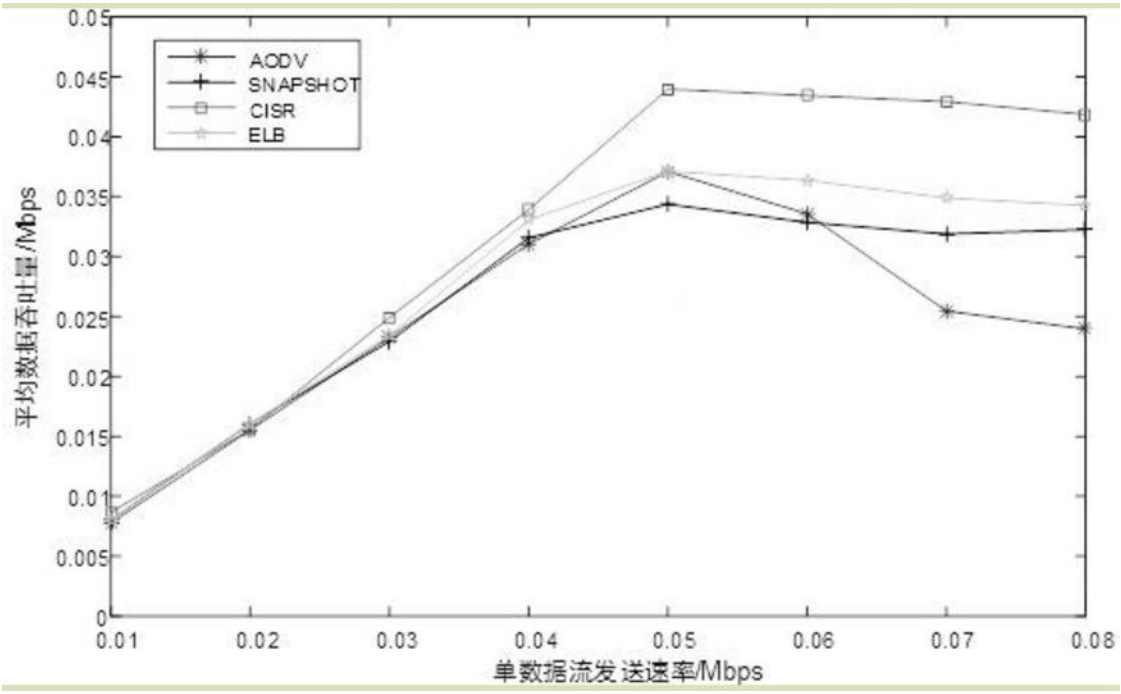

AODV
SNAPSHOT
CISR
ELB
平均数据吞吐量/Mbps
单数据流发送速率/Mbps
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08

图9

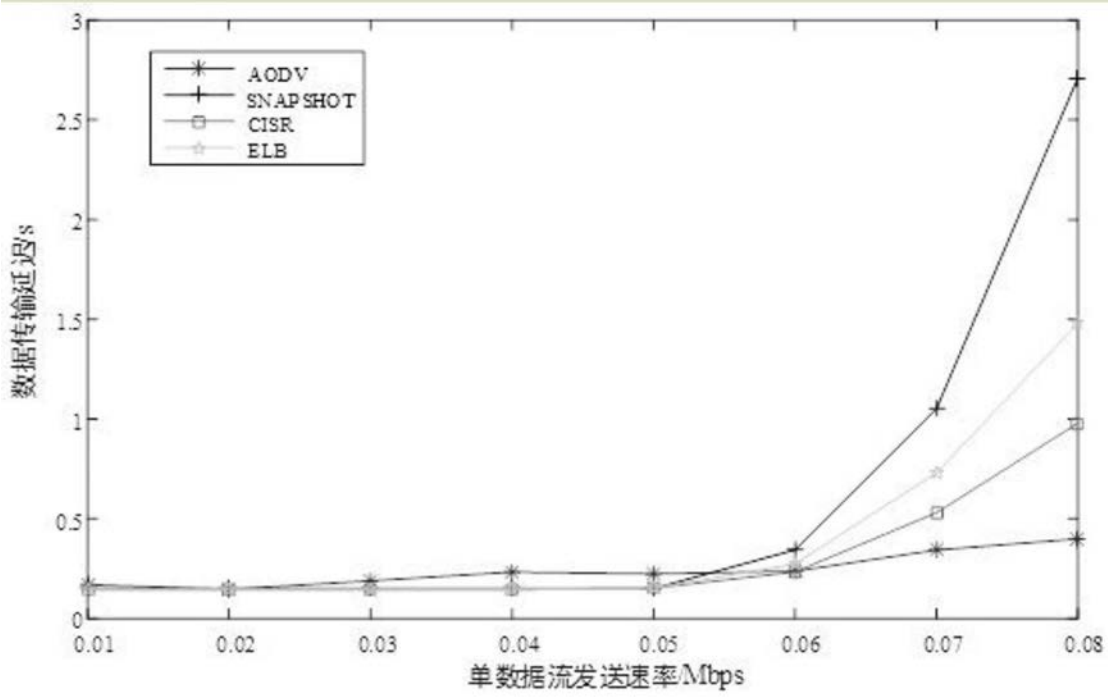

AODV
SNAPSHOT
CISR
ELB
数据传输延迟/s
单数据流发送速率/Mbps
3
2.5
2
1.5
1
0.5
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08

图10

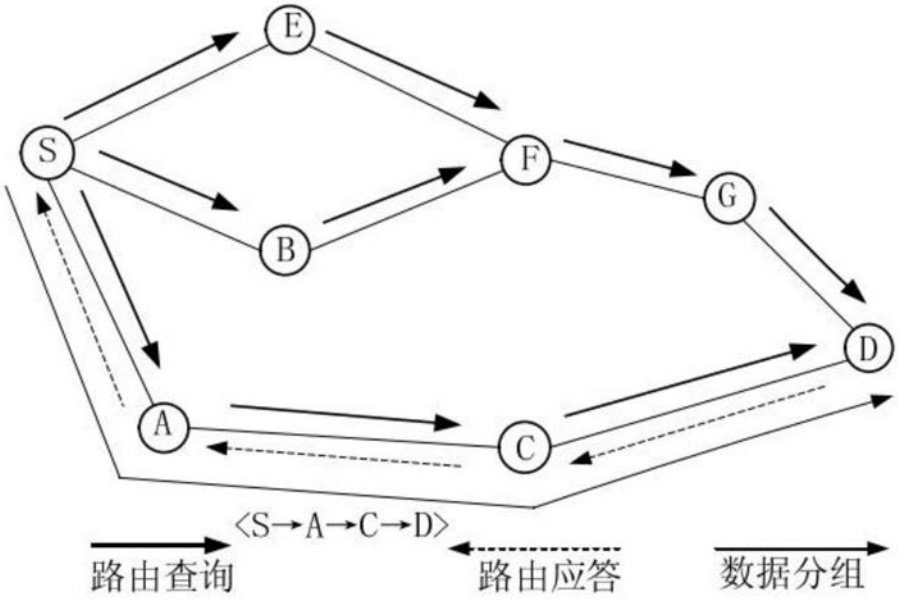


图11