

能量感知多路径负载均衡路由算法

王小平^{1,2}, 苏赛语¹, 林秦颖^{1,2}, 王路通¹

(1. 空军工程大学航空航天工程学院, 西安, 710038; 2. 西安交通大学电信学院, 西安, 710045)

摘要 针对移动自组织网络的网络拥塞问题, 基于能量感知技术并结合负载均衡和拥塞控制方法, 提出了一种能量感知多路径负载均衡路由算法。该算法利用能量感知选择满足条件的节点作为路由节点, 建立多条连接源节点和目的节点的有效路径; 同时分析路径的跳数、节点缓冲区的占用情况, 从有效路径中选出用于传输的最优路径; 然后对最优路径上的节点和路径的负载情况进行建模分析, 当节点能量、节点负载、路径负载到达设定的阈值, 就将最优路径上的流量分流到其它路径。利用 NS2 仿真软件, 在不同的场景下对该算法以及 QMRB、SMORT 进行仿真测试。仿真结果显示: 提出的算法与其它路由算法相比将网络性能提升了近 20%, 起到了均衡负载的作用, 能有效地解决网络拥塞问题。

关键词 移动自组织网络; 多路径; 负载均衡; 服务质量; NS2

DOI 10.3969/j.issn.1009-3516.2017.03.015

中图分类号 TP393 **文献标志码** A **文章编号** 1009-3516(2017)03-0085-07

An Energy-Aware Multipath Load Balancing Routing Algorithm

WANG Xiaoping^{1,2}, SU Saiyu¹, LIN Qinying^{1,2}, WANG Lutong¹

(1. Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China; 2. School of Electronic and Information Engineering, Xi'an 710043, China)

Abstract: Aimed at the network congestion problem for mobile ad hoc network based on energy aware technology and load balancing combined with congestion control method, this paper proposes an energy-aware multipath load balancing routing algorithm. The algorithm utilizes energy-aware for selecting the node of satisfying the conditions as the routing node to establish multiple effective paths of connection source node and destination node. At the same time, the paper analyzes the number of hops in the paths and the buffer occupancy of the node, selects the optimal path for transmission from the efficient paths and establishes a model of the load of nodes and paths on the optimal path. When the energy of node, the load of node and the load of the path reach the set threshold, the flow of the optimal path is diverted to other paths. The simulation software NS2 is utilized for testing this algorithm, QMRB and SMORT in different scenarios. The simulation results show that the proposed algorithm improves the network performance by about twenty percent compared with the other routing algorithms.

Key words: MANETs; multipath; load balancing; QoS; NS2

收稿日期: 2016-09-25

基金项目: 陕西省自然科学基金(2015JM6323); 航空科学基金(20145196023)

作者简介: 王小平(1972—), 男, 江苏常州人, 教授, 博士, 主要从事飞行器控制理论及仿真研究. E-mail: xiaoxiao123su666@163.com

引用格式: 王小平, 苏赛语, 林秦颖, 等. 能量感知多路径负载均衡路由算法[J]. 空军工程大学学报(自然科学版), 2017, 18(3): 85-91.
WANG Xiaoping, SU Saiyu, LIN Qinying, et al. An Energy-Aware Multipath Load Balancing Routing Algorithm[J]. Journal of Air Force Engineering University (Natural Science Edition), 2017, 18(3): 85-91.

MANETs 使用方便灵活,具有无中心、自我配置、动态变化的特点^[1]。MANETs 方便灵活的特点使其在灾难救援、军事领域、紧急通讯等方面应用十分广泛^[2]。

设计 MANETs 的路由协议具有挑战性,因为需要考虑许多物理限制^[3]。如:节点处理能力有限使 CPU 的能力有限,节点存储资源有限^[4],节点能量有限^[5],有限的带宽^[6],节点的传输范围有限^[7],快速变化的拓扑结构等^[8]。

MANETs 中提出了许多路由协议,但是很少涉及 MANETs 的负载均衡问题^[9]。负载均衡最主要的目的就是缓解拥塞路径或节点的拥塞情况。拓扑动态变化和节点移动的问题,导致 MANETs 实现拥塞控制和负载均衡变得十分困难^[10],为了平衡负载和控制拥塞就必须准确的测量网络中负载的情况。

1 研究现状

针对负载平衡问题,M. Ali 等提出了一种 QoS 感知多路径主干路由^[11],其主干路由节点具有丰富的带宽、处理能力和剩余能量,该协议通过将流量平均分配到多条主干路径上传输的方式,充分利用了网络中的带宽。文献[12]中提出了基于蚁群算法(Ant based Algorithm)的负载均衡路由,该算法使用蚁群算法寻找优化路径,并用多路径传输数据起到了负载均衡的效果。文献[13]中提出了解决负载均衡问题的新方法 FMLB(Fibonacci Multipath Load Balancing Protocol, FMLB),该算法先根据路径跳数用斐波那契序列(Fibonacci Sequence)给每条路径一个权重值,再根据权重值将传输的数据包分流到多条路径上,因此减少了拥塞,提高了传输效率。通常最短路径比其它路径使用的更加频繁,FMLB 在一定程度上提高了最短路径的利用率,它利用多路径减轻了网络的拥塞。LUNAR(Load equilibrium Neighbor Coverage Routing)结合了邻居覆盖感知和负载均衡技术的优点,通过计算每个中间节点的累积有效路径数,来决定是否重新广播路由请求数据包。它解决了广播风暴问题,同时也减少了路由开销^[14]。Reddy 等提出了 SMORT(Scalable Multipath On-demand Routing Protocol, SMORT)^[15],它用次级路径,减少了由恢复断裂路径所产生的路由开销,并能为主路径上的节点计算出多条到目的节点的路径。在高动态和高负载的情况下,SMORT 的性能要优于 AODV 和 DMRP(Disjoint Multipath Routing Protocol)。SMORT

虽然提供了安全防故障多路径(见图 1),但是并未考虑单个节点带宽、能量及负载等方面的 QoS 特点。

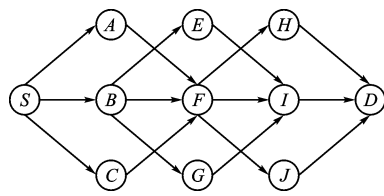


图 1 安全防故障多路径

Fig. 1 Fail-safe multiple path

多路径路由是解决负载均衡问题的基础,它能充分利用网络中的资源减缓拥塞,但同时应考虑节点是否满足 QoS 要求^[16],避免因节点质量引起的链路频繁断裂和拥塞。本文基于 SMORT,提出了改进的新的能量感知多路径负载均衡路由(Energy-aware Multipath Load Balancing Routing, EMLBR)。该协议从多条源节点到目的节点的路径中选出最优路径进行传输:当最优路径上链路的平均负载超过了定义的阈值,且链路中可用的带宽和节点剩余的能量低于设定的阈值,最优路径上的流量便会分布到多条路径传输,因此减少了拥塞提高了传输效率。

2 路由过程

本节主要对能量感知、路由发现、和路由保持过程进行了详细介绍。随机设置节点 i 的初始能量 $E_{\text{initial}}(i)$,设置节点的能量下限 $E_{\text{threshold}}$,假定 $E_{\text{threshold}}$ 的值是 25 J。

2.1 路由发现过程

源节点想要向目的节点发送数据包时,它就会向它传输范围内的邻居节点广播路由请求(Route Request, RREQ),RREQ 包的头部包含以下信息 $\{Type, Source_id, Destination_id, Source_seq, Path_traversed, Buffer_size(i)\}$ 。Type 代表数据包类型,例如:RREQ、RREP 或 RERR; $Source_id$ 表示源节点的 id, $Destination_id$ 表示目的节点的 id; $Source_seq$ 是一组单调递增的序列号,且每个包都有唯一的序列号,能识别出每一个 RREQ,它用来识别过时的 RREQ 和过时的缓存路径; $Path_traversed$ 能记录 RREQ 传输的路径信息; $Buffer_size(i)$ 表示参与到路径 P 上的中间节点,链路 i 上的缓冲区占用大小。

节点接收到 RREQ 后会将自己的 $E_{\text{initial}}(i)$ 与 $E_{\text{threshold}}$ 进行比较。如果节点的 $E_{\text{initial}}(i) > E_{\text{threshold}}$ 则接收和保存 RREQ,节点将自己的 id 等信息添加到 RREQ 中,再将其广播出去。当目的节点接收到

自每条路径上的第1个RREQ时,目的节点选择每条路径上RREQ的最后一跳节点作为向源节点发送RREP(Route Reply)的上游节点。RREP数据包的头部包含以下信息 $\{Type, Source_id, Destination_id, Reverse_Path, Buffer_size(i)\}$,这里的 $Type$ 表示RREP, $Reverse_Path$ 能记录目的节点到源节点的反向路径,其它项与RREQ中的相同。

具体算法如下:

Algorithm 1: Route discovery process

- 1: Begin
- 2: if $E_{initial}(i)$ of n_i is more than the $E_{threshold}$ value. Then
- 3: n_i received the RREQ packet and store in its header field.
- 4: n_i forwards the RREQ packet further to its neighbor nodes.
- 5: else
- 6: n_i discards the RREQ packet
- 7: end if
- 8: When the destination node receives the first RREQ from the different routes, the destination node selects the last hop of each RREQ as the upstream node to send the RREP to the source node.
- 9: End

n_i 表示源节点与目的节点之间的中间节点 i 。

如图2所示, S 、 D 分别是源节点和目的节点, A 、 E 、 F 、 H 都在 S 的传输范围内。但是节点 H 的 $E_{initial}(i)$ 低于 $E_{threshold}$,故它只能丢弃RREQ。

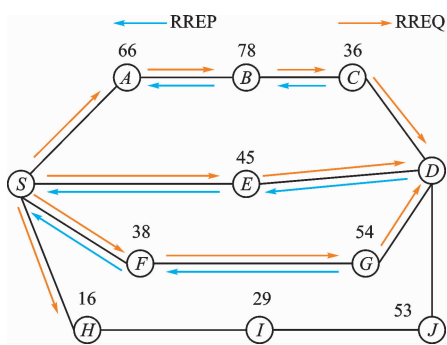


图2 路由发现和路由回复过程

Fig. 2 Route discovery and route response process

2.2 路由保持过程

若路径上的节点不能向下一跳节点传输数据,则说明该路径已经断裂了,该路径断裂的节点停止传输数据,并向源节点发送RERR(route error)报文。上游节点接收到RERR后,就会删除该路径,并将RERR传给源节点。如果多路径中的一条路径断裂了,源节点便使用剩下的有效路径进行传输;若整个路径都断裂了,源节点就重新启动路由发现过程寻找路径。RERR中含以下信息: $\{Type, Source_id, Destination_id, Reverse_Path, Buffer_$

$size(i)\}$ 。这里的 $Type$ 是RERR,其它信息与RREQ中相同,以下是路由保持算法的具体过程:

Algorithm 2: Route maintenance process

- 1: Begin
- 2: if the communication channel break down between the n_i and n_j then
- 3: n_i saves the current data packet in its routing table.
- 4: After channel break down n_i broadcasts a route repair request(R_RREQ) to n_j , n_i waits for the R_RREQ packet return from n_j within a fixed timeout ($T_{timeout}$).
- 5: else if the R_RREP packet is return to n_i within a $T_{timeout}$ then
- 6: n_i replaces the routing path and uses another path.
- 7: n_i continues packet transmission.
- 8: else
- 9: n_i sends a RERR packet to the source node and restarts the routing discovery process again.
- 10: end if
- 11: execute the step 8 in Algorithm 1
- 12: End

n_i 和 n_j 表示源节点和目的节点之间发生路径断了的2个节点。

3 路由度量标准

为提高网络资源的利用率,在选择最优路径时,同时考虑跳数和缓冲区的占用情况。为提高吞吐量、减小延时,本文综合考虑了最优路径上节点的带宽、负载及能量消耗情况。

3.1 最优路径的选取

AOMDV选择跳数最小的路径传输数据^[17],与跳数最小的路径相比,拥塞最小的路径可能端到端延时更小。为了选择从源节点到目的节点拥塞最小的路径,定义了一个新的选路标准:即考虑到路径的跳数,又考虑到路径上每个节点的负载情况。见式(1):

$$\text{Min}[\frac{1}{n_p} \sum_{i=1}^{n_p} (buffer_size(i))] \quad (1)$$

式中: $Buffer_size(i)$ 为参与到路径 P 上的中间节点,链路 i 上的缓冲区占用大小; n_p 为源节点到目的节点的路径 P 所经过的路由跳数。用式(1)计算源节点到目的节点的多条路径的值,并将取值最小的路径作为最优路径。

3.2 最优路径上带宽的计算

用文献[18]中的方法来计算可用带宽,可用带宽的计算公式如下:

$$BW_{av} = BW_{max} \times \left(\frac{Idle_t}{Int_t} \right) \quad (2)$$

式中: BW_{\max} 表示链路中的最大带宽; $Idle_t$ 表示在一定时间间隔 (Int_t) 内无线信道的空闲时间, BW_{av} 表示可用带宽。IEEE802.11 自带的感应机制能够监督信道传输状态, 它能判断无线信道是空闲还是占用。信道被占用的时间定义为忙时 $Busy_t$, 则 $Idle_t$ 可表示为:

$$Idle_t = Int_t - Busy_t \quad (3)$$

结合式(2)和式(3)就能计算出可用带宽。

3.3 最优路径上负载的计算

根据竞争窗口 (Contention Window, CW) 和队列长度来计算单个节点的负载^[19]。CW 能有效的反应信道竞争, 所以计算单个节点的平均信道竞争就能估算该节点的流量负载。每隔 t s 计算 CW 的值, 并将前一时刻的 CW 值 (CW_{old}) 和当前时刻的 CW 值 ($CW_{current}$) 加权计算如下:

$$CW = \alpha \times CW_{old} + (1 - \alpha) \times CW_{current} \quad (4)$$

为了更好地反应当前节点的状况, 就要更倾向于当前的 CW 值, 当 $\alpha=0.3$ 时效果最佳。

如果流过一个节点的流量较大, 那么该节点接口队列中数据包的数量也较多。因此, 一段时间平均的接口队列大小能反映节点流量负载情况。每隔 t s 计算接口队列大小如下:

$$Q_{len} = \beta \times qlen_{old} + (1 - \beta) \times qlen_{current} \quad (5)$$

式中: $qlen$ 为平均队列长度; $qlen_{old}$ 为前一时刻队列长度; $qlen_{current}$ 为当前时刻队列长度; β 可取 $[0, 1]$ 之间的任意值。结合 CW 和队列长度可将节点 i 的负载表示为:

$$Load_i = \gamma \times \left(\frac{CW}{\max CW} \right) + (1 - \gamma) \times \left(\frac{Q_{len}}{\max Q_{len}} \right) \quad (6)$$

为了综合考虑 CW 和 Q_{len} 的影响, γ 的值取 0.5, 即 CW 和 Q_{len} 的值越小节点的负载越小, $Load_i$ 的值越大表示节点负载越大。

3.4 最优路径上节点剩余能量的计算

本文所提出的能量感知机制来源于能量消耗模型。采用文献[20]中的方法计算单个节点的能量消耗, 传输数据包的能量消耗与传输距离成正比, 导出在 MANETs 中传送或接受 Kbit 数据包所需的能量为:

$$E_{Tx}(d) = e_{amp} \times d^2 \times k \quad (7)$$

$$E_{Rx} = e_{elec} \times k \quad (8)$$

式中: $E_{Tx}(d)$ 表示时间间隔 t 之内, 节点发送数据包所消耗的能量; d 表示传输节点之间的距离; e_{amp} 表示发送单位比特数据包放大器所需要的能量; E_{Rx} 表示时间间隔 t 之内节点接收数据包所需能量; e_{elec} 表示接受单位比特数据包所需能量。那么总的能量消耗记为:

$$E_c(t) = E_{Tx}(d) + E_{Rx} \quad (9)$$

式中: $E_c(t)$ 为时间间隔 t 之内, 单个节点消耗的总能量。则节点的剩余能量可表示如下:

$$E_R(t) = E_{initial}(i) - E_c(t) \quad (10)$$

式中: $E_R(t)$ 为时间间隔 t 之后, 节点的剩余能量。

3.5 流量负载均衡

负载均衡考虑 2 种情况: 一种是最优路径上节点的能量、负载、及可用带宽均未超过设定的阈值; 第 2 种是超过了设定的阈值, 则要进行负载分流处理。

Algorithm 3; Situation 1

- 1: If (no route to destination)
- 2: {Initiate route discovery as Algorithm 1}
- 3: If (single known route)
- 4: {Forward data packet to specified route}
- 5: * /if N routes are known from source to destination * /
- 6: {Distribute forwarding data packet to the best path}

Algorithm 4; Situation 2

- 1: Calculate the BW_{Av} of a link by using Eq (2)
- 2: If the $BW_{Av} < BW_{Th}$, then
- 4: Estimate the available load in the link ($Load_i$) using Eq (6)
- 5: If $Load_i > Load_{Th}$, then
- 6: Calculate $E_R(t)$ using Eq (10)
- 7: If $E_R(t) < ER_{Th}$, then
- 8: distribute the overload equally via multiple paths
- 9: Else send data through the best path.

BW_{Th} 取信道的最大带宽, $Load_{Th}$ 取信道所能承受最大负载的一半, ER_{Th} 取 $ER_{initial}$ 的一半。如图 3 所示, 源节点与目的节点之间有 3 条可用路径分别为 R_1 、 R_2 、 R_3 , 且 R_2 为最优路径。当 t_1 时刻 R_2 上的节点发生拥塞时 (红色节点为拥塞节点), 拥塞节点将流量负载 (假设为 210 KB/s) 平均分配到 3 条路径上。

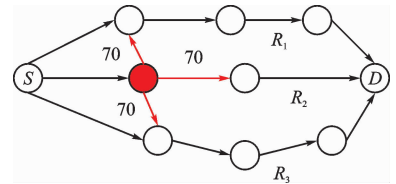


图 3 多路径负载均衡

Fig. 3 Multipath load balancing

4 仿真分析

在 NS2^[21] 上对提出的算法进行仿真, 同时测量数据传输效率 (Packet Delivery Ratio, PDR), 吞吐量 (Throughput), 端到端延时 (End to End Delay,

E2ED) 等 QoS 指标^[22]。将算法与 QMRB^[23] 及 SMORT 进行比较。

PDR 是指目的节点接收到的数据与源节点产生的 CBR 数据的比值:

$$PDR = \frac{\sum_1^n CBR_{rece}}{\sum_1^n CBR_{sent}} \times 100\% \quad (11)$$

端到端延时是指数据包从离开源节点到到达目的节点的延时,它是总的延时与总的接收到的数据包的比值:

$$E2ED = \frac{\sum_1^n (CBR_{rectime} - CBR_{sentime})}{\sum_1^n CBR_{rece}} \quad (12)$$

吞吐量是指单位时间内成功传输的数据量:

$$Throughput = \frac{\sum_1^n CBR_{rece}}{Simulationtime} \quad (13)$$

节点信道的最大传输速率设为 2 Mbps,仿真参数设置见表 1。

表 1 仿真参数设置

Tab. 1 Simulation parameter setting

Parameter	Value
Simulator	NS 2.34
MAC Type	IEEE 802.11 DCF
Signal Propagation Model	Two Way Ground
Channel Type	Wireless Channel
Simulation Area	1 250×1 250 m ²
Routing Protocol	EMLBR, QMRB, SMORT
Traffic Type	TCP/FTP
Interface Queue and Type	Droptail/PriQueue
Number of nodes	30, 50, 70, 90 and 110
Node Speed	10 m/s
Pause Time	5 s
Rate	From 250 kB/s~450 kB/s
Radio Range	250 m
Simulation Time	50 s
Traffic Source	CBR
Mobility Model	Random Way Point

首先实验测量节点数分别为 30, 50, 70, 90, 110 时且节点速度为 10 m/s 的场景;之后再测量节点数为 110, 节点速度为 10 m/s, 数据率分别为 250 kB/s, 300 kB/s, 350 kB/s, 400 kB/s, 450 kB/s 时的仿真结果。所有仿真场景中的流量来源数设为 10, 图 4~图 6 表示节点数变化的结果, 图 7~图 9 显示了数据率变化的结果。

图 4 结果显示, 特别是当节点数较大的时候,

EMLBR 与 SMORT、QMRB 相比, EMLBR 的延时较小, 因为随着节点数的增多, 它能建立多条到目的节点的安全防故障路径。EMLBR 节点数从 50~70 时, 延时变大了, 从 70 到 90 时延时减小了, 因为随着最优路径上负载的增大延时增大, 但超过某个值时负载被分流到其它路径上延时也相应减小。图 4 到图 6 显示在节点数相同的情况下 EMLBR 的数据传输效率和吞吐量与 SMORT 和 QMRB 相比有较大的提高。

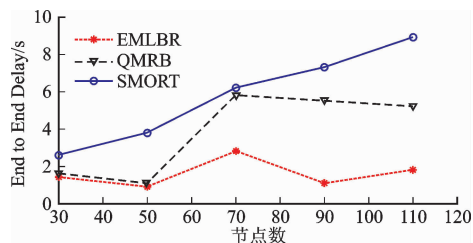


图 4 节点数与端到端延时

Fig. 4 Node vs end to end delay

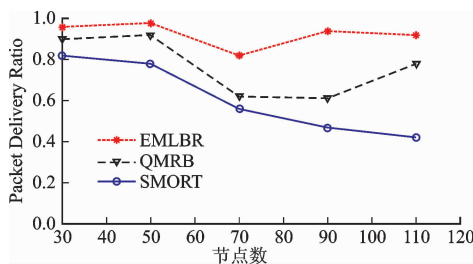


图 5 节点数与数据传输效率

Fig. 5 Node vs packet delivery ratio

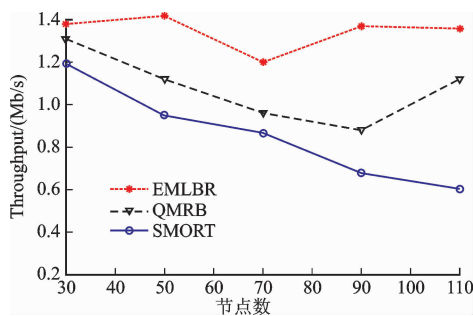


图 6 节点数与吞吐量

Fig. 6 Node vs throughput

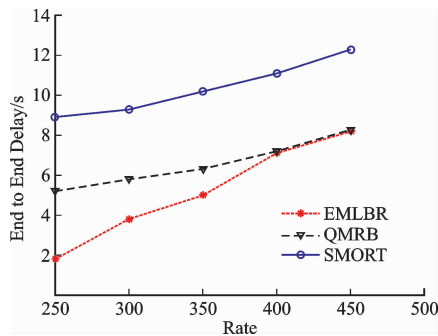


图 7 速率与端到端延时

Fig. 7 Rate vs end to end delay

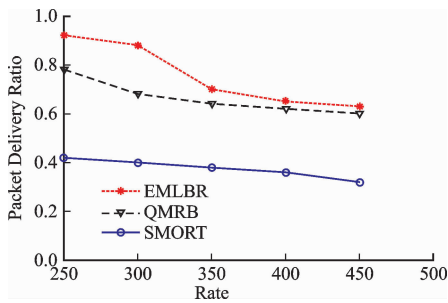


图8 速率与数据传输效率

Fig. 8 Rate vs packet delivery ratio

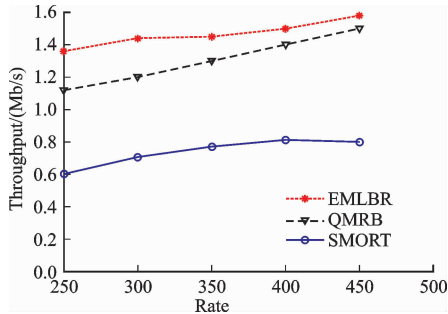


图9 速率与吞吐量

Fig. 9 Rate vs throughput

图7到图9显示当数据率变化时EMLBR在端到端延时、数据传输效率、及吞吐量等方面的性能仍然优于SMORT和QMRB,且EMLBR在数据率较高的情况下也有较好的性能。

5 结语

本文提出的EMLBR提高了吞吐量、有效地避免了拥塞,该算法考虑了可用带宽、负载、及节点剩余能量,并将流量分流到多条安全防故障路径上,能有效地管理和检测最优路径上每个节点的负载和拥塞情况。仿真结果显示,提出的算法性能要优于QMRB和SMORT。下一步工作就如何快速的选择路径展开,怎样将最优路径上的负载快速的分流将是未来工作的重点。

参考文献(References):

[1] 胡曦,王鑫,张斌. MANETs面向稳定性的自适应路由由开销控制算法[J]. 计算机科学, 2014, 41(3): 100-104.
HU X, WANG X, ZHANG B. Stability - orient Adaptive Overhead Control Algorithm in MANETs [J]. Journal of Computer Science, 2014, 41(3): 100-104. (in Chinese)

[2] 姜海龙. 移动Ad Hoc网络中具有QoS保障的多路径路由协议[D]. 西安: 西安电子科技大学, 2014.
JIANG H L. A Multipath Routing Protocol with QoS

Guarantees in Mobile Ad Hoc Networks [D]. Xi'an: Xidian University, 2014. (in Chinese)

[3] 冯慧芳,王梦茹. 节点移动性对MANETs网络拓扑特征的影响[J]. 计算机工程与应用, 2014, 50(9): 73-76.
FENG H F, WANG M G. Impact of Node Mobility on Topological Properties for MANETs [J]. Computer Engineering and Applications, 2014, 50(9): 73-76. (in Chinese)

[4] RAHIMI M R, VENKATASUBRAMANIAN N, MEHROTRA S, et al. MAPCloud: Mobile Applications on Anelastic and Scalable 2-tier cloud Architecture [C]// Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing. IEEE, 2012: 83-90.

[5] LI P, GUO S, YU S, et al. CodePipe: An Opportunistic Feeding and Routing Protocol for Reliable Multicast with Pipelined Network Coding [C]// INFOCOM 2012 Proceedings. IEEE, 2012: 100-108.

[6] 熊建,南建国,黄金科,等. 基于粒子群的多数据链任务分配模型及仿真[J]. 空军工程大学学报(自然科学版), 2017, 18(2): 95-100.
XIONG J, NAN J G, HUANG J K, et al. PSO-based Task Assignment Model and Simulation for Multi-Data Links [J]. Journal of Air Force Engineering University (Natural Science Edition), 2017, 18(2): 95-100. (in Chinese)

[7] SHENG Z, YANG S, YU Y, et al. A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities [J]. IEEE Wireless Communications, 2013, 20(6): 91-98.

[8] LI M, LI Z, VASILAKOS A V. A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues [J]. Proceedings of the IEEE, 2013, 101(12): 2538-2557.

[9] 夏浩伟,王国军,谢永明. 移动自组网中的分段式负载均衡路由协议[J]. 计算机工程, 2010, 36(4): 93-96.
XIA H W, WANG G J, XIE Y M. Segment-by-Segment Based Load Balancing Routing Protocol in MANETs [J]. 2010, 36(4): 93-96. (in Chinese)

[10] YANG M, LI Y, JIN D, et al. Software-defined and Virtualized Future Mobile and Wireless Networks: A survey [J]. Mobile Networks and Applications, 2014, 20(1): 4-18.

[11] ALI M, STEWART B G, SHAHRABI A, et al. Enhanced QoS Support in Mobile Ad Hoc Networks Using Multipath Routing Backbones [C]// Proceedings of 6th IEEE GCC Conference & Exhibition. Dubai, UAE, 2011: 315-318.

[12] GUJRAL R K, SINGH M, RANA S K. Ant Based

- Algorithm for Load Balancing in Mobile Ad Hoc Networks [J]. International Journal Computer Applications, 2012, 39(5): 35-42.
- [13] TASHTOUSH Y, DARWISH O, HAYAJNEH M, Fibonacci Sequence based Multipath Load Balancing Approach for Mobile Ad Hoc Networks [J]. Journal on Ad Hoc Networks, 2014, 16(2): 237-246.
- [14] JAIN S A, RAISINGHANI V T. Load eq Uilibrium Neighbor Aware Routing in Mobile Ad Hoc Network [C]// Annual IEEE India Conference. 2015, 2: 439-444.
- [15] REDDY L R, RAGHAVAN S V. SMORT: Scalable Multipath on-demand Routing for Mobile Ad Hoc Networks [J]. Journal on Ad Hoc Networks, 2007, 5(2): 162-188.
- [16] MOUSSAOUI A, SEMCHEDINE F, BOUKER-RAM B. A Link-state QoS Routing Protocol based on Link Stability for Mobile Ad Hoc Networks [J]. J Netw Comput Appl, 2014, 39: 117-125.
- [17] MARINA M K and DAS S R. Ad Hoc on-Demand Multipath Distance Vector Routing [J]. Wireless Communications and Mobile Computing, 2006(6): 969-988.
- [18] QIN F, LIU Y. Multipath Routing for Mobile Ad Hoc Network [C]// Proceeding of the International Symposium on Information Processing. Huangshan, China, 2009: 1024-1027.
- [19] GAO X M, ZHANG X M, DONG S, et al, Contention and Queue-aware Routing Protocol for Mobile Ad Hoc Networks [C]// International Conference on Wireless Communications, Networking and Mobile Computing. Shanghai, 2007: 21-25.
- [20] RISHIWAL V, YADAV M, VERMA S, et al. Power Aware Routing in Ad Hoc Wireless Networks [J]. Journal of Computer Science & Technology, 2009, 9(2): 101-109.
- [21] 于斌, 孙斌, 温暖, 等. NS2 与网络模拟 [M]. 北京: 人民邮电出版社, 2007.
YU B, SUN B, WEN N, et al. NS2 and network simulation [M]. Beijing: Post & Telecom Press, 2007. (in Chinese)
- [22] DEVI P R, RAO D S. QOS Enhanced Hybrid Multipath Routing Protocol for Mobile Ad Hoc Networks [J]. International Journal of Distributed & Parallel Systems, 2012, 3(6): 89-105.
- [23] IVASCU G I, PIERRE S, QUINTERO A. QoS Routing with Traffic Distribution in Mobile Ad Hoc Networks [J]. Journal on Computer Communications, 2009, 32(2): 305-316.
- (编辑: 徐楠楠)
- *****
- (上接第 30 页)
- [11] 钟大鹏, 焦志刚, 董兴. 穿甲爆破弹对钢筋混凝土靶板极限贯穿速度分析 [J]. 弹箭与制导学报, 2015, 35(6): 53-56.
ZHONG D P, JIAO Z G, DONG X. Limit Perforation Velocity Analysis of Armor-piercing High-explosive Projectile Penetrating Reinforced Concrete [J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2015, 35(6): 53-56. (in Chinese)
- [12] 黄俊卿. 基于 AUTODYN 的弹体侵彻靶板过程中仿真破片生成方法 [C]//第七届全国仿真器学术会议论文集. 沈阳, 2010.
HUANG J Q. The Fragment Generation Methods in Simulation for Projectile Penetration into Target Plate Based on AUTODYN [C]// Proceedings of the Seventh National Conference on Simulation. Shenyang, 2010. (in Chinese)
- [13] GRUJICIC M, PANDUIANGAN B, KOUDELA K L, et al. A Computational Analysis of The Ballistic Performance of Light-Weight Hybrid Composite Armors [J]. Applied Surface Science, 2006, 253(2): 730-745.
- [14] QUAN X, CLEGG R A, COWER M S, et al. Numerical Simulation of Long Rods Impacting Silicon Carbide Targets Using JH-1 Model [J]. International Journal of Impact Engineering, 2006, 33(11/12): 634-644.
- [15] JOHN F, MOXNES. Experimental and Numerical Study of the Fragmentation of Expanding Warhead Casings by Using Different Numerical Codes and Solution Techniques [J]. Defense Technology, 2014, 2(1): 161-176.
- [16] 宁建国. 爆炸与冲击动力学 [M]. 北京: 国防工业出版社, 2010.
NING J G. Explosion and Shock Dynamics [M]. Beijing: National Defense Industry Press, 2010. (in Chinese)
- [17] 王礼立. 冲击力学进展 [M]. 合肥: 中国科技大学出版社, 1992.
WANG L L. Development of Shock Dynamics [M]. Changsha: University of Science And Technology of China Press, 1992. (in Chinese)
- [18] Holmquist T J, Johnson G R, Cook W H. A Computation Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates and High Pressures [C]//Proceedings of the 14th International Symposium on Ballistics, 2003: 591-600.
- (编辑: 徐敏)