

DTN 中基于节点价值的效用路由算法*

李家瑜, 张大方, 何施茗

(湖南大学 信息科学与工程学院, 长沙 410082)

摘要: 为了解决延迟容忍网络(DTN)中传统路由算法中消息被分配的网络资源不均衡及节点负载不均衡问题,结合消息效用值提出了一种基于节点价值的效用路由算法。算法根据动态改变的消息效用值选择最高优先级的消息(具有最小TTL和到目的节点最短距离的消息)进行转发,以使得为每个消息分配的网络资源相对均衡;同时,根据节点的价值(与节点速度和剩余缓存有关)选择下一跳节点,以平衡每个节点的负载;另外,算法还采用了一定的消息管理机制及时清除缓存空间。通过仿真实验及性能分析表明,该算法在传输成功率、传输延迟和网络开销上都有明显的改善。因此,通过充分利用网络资源提高了算法的整体性能。

关键词: 延迟容忍网络;路由算法;效用值;节点价值;缓存区管理;资源分配;负载均衡

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2012)09-3379-04

doi:10.3969/j.issn.1001-3695.2012.09.047

Utility routing algorithm based on value of node in DTN

LI Jia-yu, ZHANG Da-fang, HE Shi-ming

(College of Information Science & Engineering, Hunan University, Changsha 410082, China)

Abstract: In order to solve the problems that network resources allocated to messages were unbalanced and the load of node was not balanced in traditional routing algorithms in delay/disruption tolerant networks, this paper proposed a utility routing algorithm based on the value of node combined with the message utility. The algorithm selected the highest priority message (message with the least remaining TTL (time to live property) and the shortest distance to the destination) to forward based on message utility dynamic changing. It allowed allocating relatively balanced network resources to each message. Besides, it selected the next hop based on the value of node, which related to the speed and remaining buffer of the node. This work balanced the load of each node. What is more, it adopted some buffer management mechanisms to free buffer. Simulation results and performance analysis show that, this algorithm improves obviously delivery ratio, latency and overhead. Therefore, it improves overall performance of this algorithm by making full use of the network resources.

Key words: delay/disruption tolerant network (DTN); routing algorithm; utility; value of node; buffer management; resource allocation; load balancing

0 引言

DTN(容迟/容断网络)是近年来国内外兴起的一种新兴的网络体系结构。它是为了解决受限网络中由于节点移动、节点稀疏等造成的网络拓扑结构变化、网络频繁断开等情况下消息无法传递的问题。这些受限网络存在于星际网络、深海探测、野生动物追踪等。与传统的因特网和移动 Ad hoc 网络^[1]的前提条件不同的是:DTN 网络中可能不存在一条完整的端到端的路径,因此,传统的路由算法已经不适用于这样的网络环境。DTN 在传统网络中的传输层和应用层之间定义了一个新的协议层,称之为束层,束层以“存储—携带—转发”的模式实现束(又称消息)在节点之间的传递,以最终达到成功地将消息从源节点传到目的节点。

研究者设计了很多经典的路由算法^[2],目前许多算法都是以多复制路由策略^[3]来实现 DTN 网络中消息的传递,如传染病路由(epidemic forwarding)^[4]、spray and wait/focus^[5,6]等,这些多复制路由策略是指对于同一个报文在网络中复制多次

并分布到网络中不同的节点,通过增加复本数来增加消息传输的成功率和降低传输时延。Epidemic forwarding 是 DTN 中最原始的路由策略,只要两个节点相遇,就会从对方获得自己没有的消息。这种方法在网络带宽和缓存资源充足的情况下可以使算法效率达到最优,而现实网络中这些资源都是有限的,传染病算法会使整个网络的开销特别大,随着节点数的增多,反而会使网络的性能急剧下降。因此,spray and wait 算法提出通过消息拷贝值 L 来限制网络中复本的数量,按照二分喷洒的方式进行消息的复制,但在喷洒完成之后网络中只有固定的 L 个节点携带消息并等待目的节点。这种算法有效地减小了开销,但是由于在第二个阶段采用直接传输^[7],极大地增加了传输延迟。

本文结合了 epidemic forwarding 和 spray and wait 各自的优点,并克服了对方的缺点,结合消息效用值提出基于节点价值的贪婪复制算法(VoN)。VoN 算法采用 spray and wait 算法中的二分喷洒消息的方式。在喷洒阶段,选择优势节点(速度大、剩余缓存多的节点)进行喷洒,当消息的拷贝值 L 为 1 时进

收稿日期: 2011-12-17; **修回日期:** 2012-02-10 **基金项目:** 国家“973”计划资助项目(2012CB315801);国家自然科学基金资助项目(61173167)

作者简介: 李家瑜(1986-),女(土家族),湖南张家界人,硕士,主要研究方向为无线网络(jyleeathnu@163.com);张大方(1959-),男,教授,博导,主要研究方向为可信系统与网络、容错计算、网络测试、软件工程等;何施茗(1986-),女,博士研究生,主要研究方向为无线网络。

入第二阶段,将消息 m 复制给更具优势的节点(速度大、剩余缓存多的节点),同时不删除当前节点的消息 m 。本算法还利用 ORWAR^[8] (opportunistic DTN routing with window-aware adaptive replication)算法中的每比特效用值(消息效用值/消息大小, utility/bit)的思想,提出效用值动态改变的消息选择机制,得到基于消息的剩余 TTL 和与目的节点之间的距离的效用函数,根据每比特效用值由大到小选择所要转发、复制或删除的消息。该算法每次选择更有价值的节点,并充分利用其可用资源,有效地平衡了每个节点的负载,同时增加了消息传输成功率并减小了消息传输延时。另外,动态改变的消息效用值使得每个消息被选择的机会变得均衡,消息被分配资源的机会变得均衡,因此,消息在缓存区停留的时间变短。

1 相关工作

随着近年来研究者对 DTN 网络的关注,DTN 网络中的路由算法也成为研究者广泛关注的问题。文献[2]中全面介绍了当前 DTN 网络领域研究热点问题,包括多复制路由、基于效用的单复制路由等。像 epidemic forwarding、spray and wait/focus、ORWAR 这些多复制路由策略都是通过增加网络中消息的复本数来提高传输成功率。Epidemic forwarding 使用同步数据库复制技术,当两个节点相遇,交换彼此的消息矢量,从对方获得自己没有的消息,但是它没有充分考虑到节点的价值,盲目地复制而导致网络资源浪费。Spray and wait 通过消息拷贝值 L 限制了每个消息在网络中的复本数。该算法分为两个阶段,首先是喷洒阶段,每个消息创建时,消息的拷贝值为 L ,当携带该消息的节点在遇到其他没携带该消息的节点时,将消息传给相遇节点并将 L 设置为 $L/2$,自己保留 $L/2$ 的拷贝值,直到消息的拷贝值 $L=1$,进入等待阶段,直到遇到目的节点,才将消息递交给目的节点。Spray and focus 是对 spray and wait 的第二个阶段进行改进,将消息中继到投递概率更高的节点,自己删除该消息,因此网络中的复本数仍然是 L 。Spray and wait/focus 限制了网络中的复本数,减小了开销,但同时也加大了传输延时。ORWAR 算法是利用二分喷洒的方式,它给出三个等级的消息效用值(低、中和高)来区分消息的优先级,效用值在消息创建时随机产生,并在传递过程中保持不变,在选择要转发或复制的消息时,首先将消息按照每比特效用值从大到小的顺序排列;再根据携带消息的节点与相遇节点之间的碰撞窗口估计所能传输的消息大小。从队首选择每比特效用值最大且消息大小小于通过碰撞窗口估计所能传输的大小的消息进行转发或复制。ORWAR 算法可以部分避免因为网络带宽有限而造成数据包在传输过程中断,尽量减少包的碎片,但是在现实网络中,消息的效用值应该随时间、消息所处的位置而变化,并且两个相遇节点之间的碰撞窗口大小也可能因为节点的运动而估计不准确。PropHet (probabilistic routing protocol using history of encounters and transitivity)^[9] 是一种基于历史信息 and 可传递概率的路由协议,它利用节点相遇的历史信息和概率的传递性来估计每个节点与其他节点相遇的概率,将消息向到达目的节点概率高的节点传输,这种算法减小了网络开销,但算法在传输期望值的计算和拥塞处理等方面还存在不足。

在以上路由策略中, epidemic forwarding、spray and wait、ORWAR 均没有对下一跳节点进行选择,PropHet 通过概率选择下一跳节点,但在拥塞处理等方面存在不足,因此具有一定

的盲目性和局限性。

本文利用二分喷洒方法,结合消息动态改变的效用值来选择优先级更高的消息进行复制,同时,根据节点价值来选择下一跳节点。通过与几种经典算法的性能比较可知,本文提出的基于节点价值的效用路由算法有效地增加了传输成功率,减小了传输延时,同时使每个消息被分配资源的机会变得均衡,从而减小了消息在缓存区停留的时间。

2 基于节点价值的效用路由算法

2.1 动态改变效用值的消息选择机制

与 ORWAR 算法一样,VoN 算法中每个消息都有一个效用值的属性用来标志消息的优先级,为了更准确地反映消息的效用值,本文改进 ORWAR 算法中的效用值,提出了消息效用值函数(式(1))。消息效用值与消息剩余 TTL、当前节点到目的节点的距离均成反比。剩余 TTL 越小,消息越有可能因过期被删除,消息转发的紧急程度越高;消息距离目的节点越近,在节点移动过程中越有可能遇到目的节点,因此被赋予较高的效用值。消息在队列中的顺序仍然采用 ORWAR 中每比特效用值由大到小排列的思想。消息效用值动态改变使得每个消息被分配到网络资源的机会变得均衡。

$$U_i = \frac{1}{TTL_i \times D_i} \quad (1)$$

$$D_i = \sqrt{(x_c - x_d)^2 + (y_c - y_d)^2} \quad (2)$$

其中: x_c, y_c 分别表示消息所在的当前节点的 x, y 坐标, x_d, y_d 分别表示消息目的节点的 x, y 坐标; TTL_i, D_i, U_i 分别表示消息 i 的剩余生存期、与目的节点的距离、效用值。消息的效用值越大,优先级越高。随着时间的推移,消息所在位置的变化,消息的效用值动态更新,消息队列也随着效用值的变化而变化。消息队列如图1所示。



图1 消息优先级队列

在进行消息复制或转发时,每次都选择队首的消息进行复制或转发;接收消息时,如果要接收的消息的每比特效用值小于消息队列中队尾的消息效用值,则不接收;若大于消息队列中队尾的消息,则先删除队尾的消息,再接收。

2.2 基于节点价值的消息复制机制

定义1 节点价值。它反映了节点转发消息的能力,受节点速度、可用缓存区的影响。节点移动速度越快,遇到其他节点的次数就越多;节点剩余缓存越多,接收消息的能力就越强,越不容易因为节点级的拥塞造成丢包。

由此,可得节点价值计算公式:

$$val_i = V_i \times S_{ir} \quad (3)$$

其中: val_i, V_i, S_{ir} 分别表示节点价值、节点速度、节点剩余缓存大小。节点价值与节点速度、剩余缓存成正比。当节点的剩余缓存 S_{ir} 为 0,即使节点速度大,也不应选择此节点作为消息复制的下一跳,以尽量避免因节点拥塞而丢包。根据剩余缓存选择下一跳节点使节点的负载相对均衡。

本算法采用 spray and wait 的二分喷洒方式转发消息。当

前有两个或多个可用连接时,根据节点价值选择最优节点进行转发。算法流程如下:

a) 所选择的消息的目的节点是多个连接的节点中的一个,则直接将消息传给目的节点。

b) 当前节点 n_i 只遇到一个节点 n_j , 并且 n_j 不是目的节点, 通过式(3)计算 n_i 和 n_j 的节点价值, 如果 n_i 的节点价值大于 n_j , 不进行消息的复制, 否则, 如果 $L > 1$, n_i 将消息拷贝给 n_j , 并将两个节点中的消息拷贝值 L 设置为 $L/2$; 如果 $L = 1$, n_i 将消息拷贝给 n_j , 并且不删除 n_i 中的消息。

c) 所选择的消息的下一跳节点不是目的节点, 且遇到了多个节点, 通过式(3)计算各节点的节点价值, 选择价值最大的节点并执行步骤 b)。

算法中选择性喷洒使得每个节点的负载变得均衡, 同时, 在算法的第二个阶段采用选择性的洪泛, 增加了消息被传输的机会。算法的流程如图2所示。其中, 图2中的变量 C 表示可用连接另一端的所有节点。

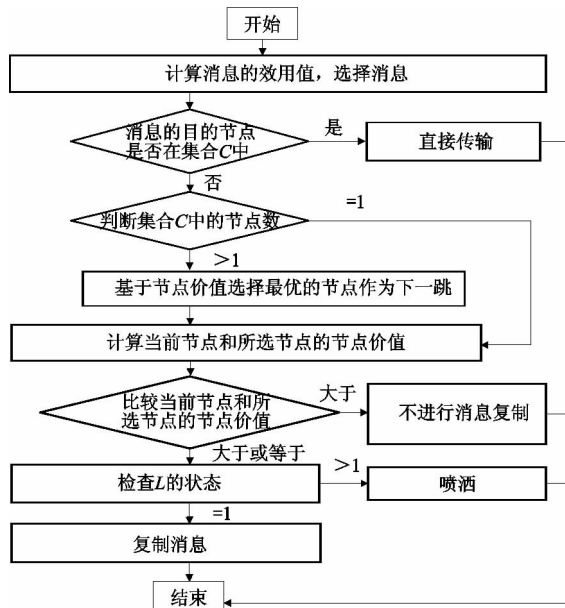


图2 算法流程

2.3 缓存区管理

在文献[10~12]中有 drop oldest (DO)、drop youngest (DY)、drop last (DL)、drop front (DF)、N-Drop 等拥塞控制策略, 当节点发生拥塞时, 以上每种策略在特定的场景下都能起到优化节点缓存区的效果。本文中采用的删包机制分为以下三个方面:

a) 删除 TTL 过期的消息。

b) 删除已经传输成功的消息以及及时释放缓存区。在每个节点的路由器中均保存一份传输成功的消息表 T_{ddm} , T_{ddm} 采用哈希表的存储结构。其中, 消息的 ID 为表的键, 消息的剩余 TTL 为值, 使用 TTL 属性来删除 T_{ddm} 中 TTL 过期的消息, 避免了存储整个消息造成 T_{ddm} 过大, 浪费网络资源。每个节点定时广播自己的 T_{ddm} 表, 并设置定时器 T 。在 T 时间间隔内, 节点获取邻居节点的 T_{ddm} 表, 并与自身的表求并集, 更新自身的 T_{ddm} ; 最后将 T_{ddm} 中传输成功的消息从队列中删除。

c) 如果在进行了上述 a) b) 的删包机制之后要接收的消息的大小仍然大于缓存的大小, 则按每比特效用值大小遍历删除

队尾消息, 如果要删除的消息正在传输或拷贝值 L 大于 1 (当 L 大于 1 时, 说明消息还只经过极少次的复制), 则不删除, 继续从队尾遍历删除每比特效用值第二小的消息。

3 仿真结果及性能分析

3.1 仿真环境及参数设置

本文选用 ONE^[13] (opportunistic network environment) 仿真平台对基于节点价值的效用路由算法 (VoN)、传染病路由 (epidemic)、spray and wait (SNW)、ORWAR、PropHet 这五种算法的性能进行比较。仿真场景采用芬兰的 Helsinki 城市交通地图。场景区域大小为 $4500 \text{ m} \times 3500 \text{ m}$, 节点数为 100 个, 其中, 75 个行人、25 辆小汽车, 行人具有 5 MB 的缓存大小、通信半径为 10 m、移动速度为 $[1.8, 5.4] \text{ km/h}$, 汽车具有 50 MB 的缓存大小、通信半径为 10 m、移动速度为 $[10, 50] \text{ km/h}$ 。节点的通信接口均采用蓝牙接口 250 kbps (2 Mbps), 消息的生存期为 1 h, 消息产生的时间间隔为 $[25 \text{ s}, 35 \text{ s}]$, 消息的大小为 $[500 \text{ k}, 1 \text{ M}]$, 消息拷贝值参数 L 为 6。

3.2 性能分析

本文分别对 VoN、epidemic、SNW、ORWAR、PropHet 这五种算法在传输成功率、平均传输延迟、网络开销率和消息在缓存区停留的平均时间这四个方面的性能进行比较分析。仿真的时长分别为 0.5 h、1 h、1.5 h、2 h、2.5 h、3 h 六种情况。

消息传输成功率 (delivered_prop): 成功递交给目的节点的数据包个数 / 产生的数据包总数。

平均传输延迟 (latency_avg): 所有投递成功的数据包从源端到目的端的时间之和 / 总的数据包个数。

网络开销率 (overhead_radio): (网络中被中继的数据包 - 成功递交的数据包) / 成功递交的数据包。

消息在缓存区停留的平均时间 (buffertime_avg) 表示消息停留在每个节点缓存区的平均时间。

1) 消息的传输成功率

从图3可以看出, 在网络刚启动不久, VoN 算法和 epidemic 就具有很高的传输成功率, 但在 1.5 h 之后 VoN 算法比 epidemic 优, 这是因为每次都会选择最好的节点喷洒或洪泛而不是盲目地复制, 同时采用了一定的删包机制以对节点的资源进行管理。从图中可以看到, ORWAR 算法一直也具有较高的传输成功率, 因为它每次根据碰撞时间窗口选择相对小的消息进行转发, 但是在本文中, 消息的大小设置为 $[500 \text{ k}, 1 \text{ M}]$, 小的消息相对较少, 这样会影响到算法的性能。VoN 的传输成功率在 1.5 h 之前一直是 SNW 的两倍, 运行 3 h 后比 SNW 高出 19%, VoN 的增幅小于 SNW 是因为它在第二个阶段采用了选择性的洪泛。

2) 平均传输延迟

从图4可以看出, 从 1 h 之后, VoN 算法的平均延迟是五种算法中最小的, 到达 3 h 后, VoN 算法的延迟比 SNW 的延迟少 800 s (13 min) 左右。这是因为在经过一定的时间, VoN 算法网络中的复本数要比 SNW 算法多, 复本数的增加虽然会加大系统开销, 但同时减少了消息从源节点传输到目的节点之间的时间, 从而降低了传输延迟。因为在 VoN 算法中一直选择速度快的节点喷洒或洪泛, 所以 VoN 算法比 epidemic 在传输延迟上一直

要优。

3) 网络开销率

从图 5 中可以看出,VoN 算法的开销比 SNW、PropHet、ORWAR 的开销要大。因为第二阶段选择的是洪泛算法,网络中副本数多,占用的缓存、消耗的能量也多,但该算法正是有效地利用节点的剩余缓存,以牺牲缓存空间为代价来换取高的传输成功率和短的传输延时。由于 VoN 算法采用了选择性的复制与删包机制,避免了数据包的盲目洪泛,因此 VoN 算法相比 epidemic 极大程度上降低了网络开销,约为 epidemic 的 1/3。

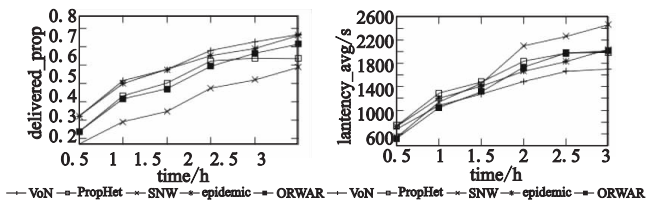


图3 消息的传输成功率
随时间的变化

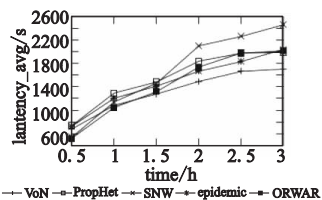


图4 消息的平均传输延时
随时间的变化

4) 消息在缓存区停留的平均时间

从图 6 中可以看出,VoN 算法中消息在缓存区停留的平均时间是最短的,说明整个网络的转发性能好。由于采用动态改变的效用值选择消息,使得每个消息被选择转发的机会变得相对均衡,消息被分配到资源的机会变得均衡,从而减小了消息在缓存区停留的平均时间。

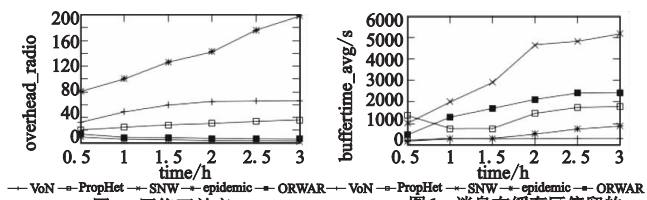


图5 网络开销率
随时间的变化

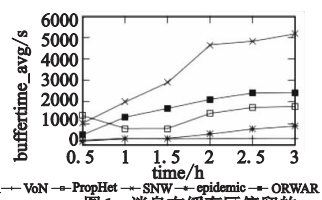


图6 消息在缓存区停留的
平均时间随时间的变化

4 结束语

本文结合消息动态改变的效用值和节点价值,并采用一定的缓存区管理机制,提出了一种性能更优的效用路由算法。通过与其他经典路由算法比较,本文算法提高了消息传输成功率、降低了消息传输延时并减小了消息在缓存区的停留时间。从仿真结果可以看到,算法在网络开效率上相对于传染病算法优化了很多,但是与其他几种算法相比较,减小算法的网络开销还是有提升的空间。未来的工作将在保证其他性能较优的情况下进一步减小算法的网络开销。

参考文献:

[1] PERKINS C E. Ad hoc networking[M]. Boston: Addison-Wesley Longman Publishing Co., Inc, 2001.

[2] 熊永平, 孙利民, 牛建伟, 等. 机会网络[J]. 软件学报, 2009, 20(1): 124-137.

[3] SPYROPOULOS T, PSOUNIS K, RAGHAVENDRA C S. Efficient routing in intermittently connected mobile networks: the multiple-copy case[J]. IEEE/ACM Trans on Networking, 2008, 16(1): 77-90.

[4] VAHDAT A, BECKER D. Epidemic routing for partially connected ad hoc networks, CS-2000-06[R]. Durham, NC: Duke University, 2000.

[5] SPYROPOULOS T, PSOUNIS K, RAGHAVENDRA C S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks[C]//Proc of ACM SIGCOMM Workshop on Delay-Tolerant Networking. New York: ACM Press, 2005: 252-259.

[6] SPYROPOULOS T, PSOUNIS K, RAGHAVENDRA C S. Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility[C]//Proc of the 5th Annual IEEE International Conference on Pervasive Computing and Communications. Washington DC: IEEE Computer Society, 2007: 79-85.

[7] SPYROPOULOS T, PSOUNIS K, RAGHAVENDRA C S. Single-copy routing in intermittently connected mobile networks[C]//Proc of the 1st IEEE Communications Society Conference on Sensor and Ad hoc Communications and Networks. 2004: 235 - 244.

[8] SANDULESCU G, NADJM-TEHRANI S. Opportunistic DTN routing with window-aware adaptive replication[C]//Proc of the 4th Asian Conference on Internet Engineering. New York: ACM Press, 2008: 103-112.

[9] LINDGREN A, DORIA A, SCHELÉN O. Probabilistic routing in intermittently connected networks[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003, 7(3): 19-20.

[10] ZHANG Xiao-lan, NEGLIA G, KUROSE J, et al. Performance modeling of epidemic routing[J]. Computer Network: the International Journal of Computer and Telecommunications Networking, 2007, 51(10): 2867-2891.

[11] KRIFA A, BARAKA C, SPYROPOULOS T, et al. Optimal buffer management policies for delay tolerant networks[C]//Proc of the 5th IEEE Communications Society Conference on sensor and Ad hoc Communications and Networks. San Francisco: [s. n.], 2008: 260-268.

[12] LI Yun, ZHAO Ying, LIU Zhan-jun, et al. N-Drop: congestion control strategy under epidemic routing in DTN[C]//Proc of 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. New York: ACM Press, 2009: 457-460.

[13] KERÄNEN A, OTT J, KÄRKÄINEN T. The ONE simulator for DTN protocol evaluation[C]//Proc of the 2nd International Conference on Simulation Tools and Techniques. Belgium: ICST, 2009: 55.