

RPL 的 ETX 路由度量研究

曹祥仪 曾 碧 何翠红

(广东工业大学计算机学院 广州 510006)

摘 要 ETX(Expected Transmission Count)是现有的低功耗有损网络路由协议(Routing Protocol for Low power and Lossy Networks,RPL)采用的一种路由度量。ETX 最小化了两个节点间成功传输一个数据包所需发送的次数,但在数据传输的实时性方面的考虑有所欠缺。在 ETX 的基础上最小化数据传输延时,保证数据传输的高成功率和低延时,最后在 contikiOS 中实现并与标准的 ETX 进行了比较。

关键词 无线传感网,ETX,RPL,路由度量

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.020

Research for ETX Routing Metric of RPL

CAO Xiang-yi ZENG Bi HE Cui-hong

(College of Computer Science,Guangdong University of Technology,Guangzhou 510006,China)

Abstract ETX (Expected Transmission Count) is routing metric of existing RPL(Routing Protocol for Low power and Lossy Networks). ETX minimizes required total number of delivering a packet between two nodes successfully. However,it does not take the real-time data transmission into account. This paper minimized the latency of data transmission based on ETX,and guaranteed high success rate and low latency of data transmission. Finally, the improved algorithm was implemented in contiki OS and compared with the standard ETX.

Keywords WSN,ETX,RPL.,Routing metric

1 引言

当前 WSN(Wireless Sensor Network)已经在向着 IPV6 的方向发展,低功耗有损网络路由协议(Routing Protocol for Low power and Lossy Networks,RPL)^[3]是 IETF 为低功耗无线传感网制定的 IPV6 路由协议。在一个采用 RPL 的 WSN 中,一个节点作为根节点(DAG root),与 WSN 中其他节点建立基于目标的有向无环图(DODAG),该 DODAG 即为 WSN 的最优网络拓扑^[2]。DODAG 中任意两点间的路径即为最优路径,RPL 根据目标函数(OF)^[4]来计算两个节点间的最优路径。目前在 contikiOS 中实现的 OF 有基于跳数、基于节点能量(node-energy)和基于预计发送次数(ETX)^[5],基于跳数和节点能量的 OF 在这里不做讨论。根据 ETX 的定义,一个节点成功发送一个数据包到目的节点需要发送该数据包的次数最少,ETX 是由节点在一段时间内发送数据包的成功率(d_f)和接收数据包的成功率(d_r)计算得出,并未考虑数据包传输所需要的时间。

本文将改进 contikiOS 实现的标准 ETX 算法,改进的算法能够在保证数据传输成功率的基础上使得传输延时最小,最后将改进的算法在 cooja^[6]模拟器上实现并进行评估。

本文第 2 节对 ETX 路由度量算法进行概述并提出延时的计算方法;第 3 节对 ETX 改进算法进行详述;第 4 节给出

评估结果;最后总结全文。

2 RPL 的路由度量概述

RPL 的路由度量是 WSN 中的节点建立 DODAG 的依据,由节点的 OF 决定。到目前为止,ROLL 工作小组已经制定了 OF0(Objective Function Zero)^[7]和 MRHOF(Minimum Rank with Hysteresis Objective Function)^[8]。OF0 只以最小跳数作为建立路由的依据;MRHOF 则以路由度量作为建立路由的依据,RFC6551^[4]定义的路由度量有多种,但目前没有任何一个路由度量的标准实现,比较权威的有 contikiOS 中的 node-energy 和 ETX,这里只对 ETX 进行研究。

2.1 ETX 路由度量的实现

一条链路的 ETX 即是成功传输一个数据包需要传输的次数,其中传输是双向的;一条路由的 ETX 则是每条链路的 ETX 的总和。例如,一条数据传输成功率为 100% 的 3 跳路由,其 ETX 值为 3;一条数据传输成功率为 50% 的单跳路由,其 ETX 值为 2。

一条链路的 ETX 可以用式(1)计算出来:

$$ETX = \frac{1}{d_f * d_r} \tag{1}$$

其中, d_f 表示发送传输率, d_r 表示接收传输率。

假设有一条链路 link(A,B),如图 1 所示。

到稿日期:2015-03-18 返修日期:2015-07-12

曹祥仪 硕士生,主要研究方向为无线传感网,E-mail:975305728@qq.com;曾 碧 博士,教授,主要研究方向为软计算和嵌入式系统;何翠红 硕士,副教授,主要研究方向为嵌入式系统。

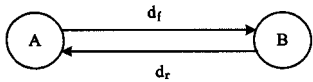


图1 A和B之间的链路

其中, d_f 可表示 A 发送给 B 的数据传输成功率, d_r 表示 A 接收 B 的数据传输成功率。计算 d_f 和 d_r , 需要专用的探测数据包, 每个节点需要周期性地广播探测数据包, 通过式(2)计算出单向数据传输成功率, 用 $r(t)$ 表示。

$$r(t) = \frac{x}{\text{count}(A)} \quad (2)$$

其中, x 表示实际传输的数据包, $\text{count}(A)$ 表示应该传输的数据包。

计算一条链路的 ETX 需要用到 d_f 和 d_r 。根据图 1, B 节点发出的每个探测包中包含了 B 节点从各邻居节点实际接收到的探测数据包的个数, 这样 A 节点就可以通过 B 广播的探测数据包知道 A 成功发送的探测数据包的数量, 即式(2)中的 x , $\text{count}(A)$ 是 A 本来就知道的, 所以根据式(2)可以计算出 d_f ; 由于广播探测数据包的周期时间是已知的, 因此 A 可以知道在一段时间内应该收到 B 的探测数据包的数量, 则根据自身统计的实际接收 B 的探测数据包的数量, 可计算出 d_r 。根据式(1), A 节点就可以计算出 $\text{link}(A, B)$ 的 ETX, 同理 B 也可以计算出 $\text{link}(A, B)$ 的 ETX。

2.2 延时计算

在 WSN 中, 节点通常都是由电池供电, 所以节点需要采用休眠机制, 以节约电池电量, 延长节点寿命, 节点会周期性唤醒并且只有在唤醒的状态才能进行数据的收发, 由于各节点的唤醒周期不一定相同, 因此存在一个节点需要一直发送数据包直到另一个节点被唤醒接收, 此时就会产生延时。

数据传输产生的延时如图 2 所示^[9]。

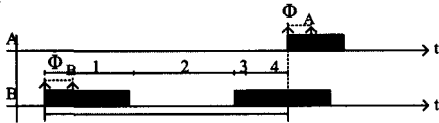


图2 B节点发送数据包给A节点的时间图

图 2 表示 B 给 A 发送数据包的过程, 图中标有数字的 4 个部分表示 B 成功发送数据包到 A 节点产生的延时, 其中 1 表示 B 节点准备数据包的时间; 2 表示 B 节点等待时间; 3 表示抢占信道所花的时间; 4 表示等待 C 节点唤醒的时间。理想状态下, 1, 3, 4 是必要的时间, 为发送数据包所需要的最短时间, 将这些时间间隔的总和称为最小转发时间 (Minimum Forwarding Time, MFT), 则转发延时 D 的计算分为以下两种。

(1) A 与 B 的唤醒周期 C_T 相同。这种情况下, $C_{T,A} = C_{T,B}$, A 和 B 同时唤醒, 则 B 的唤醒时段和 A 的唤醒时段的差值是一个常数, 因此 D 可以根据式(3)计算出来。

$$D = \begin{cases} \Phi_A - \Phi_B, & \Phi_A - \Phi_B > MFT \\ \Phi_A - \Phi_B + C_{T,A}, & \Phi_A - \Phi_B \leq MFT \end{cases} \quad (3)$$

其中, Φ_B 为 B 节点的唤醒时段, Φ_A 为 A 节点的唤醒时段。

(2) A 与 B 的唤醒周期 C_T 不相同。这种情况下, $C_{T,B} \neq C_{T,C}$, 如图 3 所示, 只有 B 在时间段区域 τ 内唤醒发送数据包才可以被 A 接收, B 开始发送数据包的时刻在 τ 内是一个随机值, 即 $\Phi_A - \Phi_B$ 不是恒定的常数, 因此只能计算 A 与 B 之

间的平均延时, 此时的延时 D 就由式(4)计算。

$$D = ((C_{T,A} + MFT) + MFT) / 2 = C_{T,A} / 2 + MFT \quad (4)$$

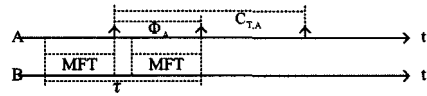


图3 A和B对应的唤醒时段, B只有在 τ 内开始发送数据包才能保证 A 能接收到数据包

3 算法改进

第 2 节给出了 ETX 路由度量实现方案和延时的计算, 这里有以下两点需要讨论。

(1) ETX 路由度量未考虑到延时因素。2.1 节中, ETX 是根据节点周期性发送和接收的广播探测数据包计算得出, 其周期时间是固定的, 每个节点在与其邻居节点进行交互时会协商发送探测数据包的时间和周期。

(2) 在目前采用 RPL 的 WSN 中, 任意两节点间的数据传输都需要先单播给 DODAG 根节点, 然后根节点再转发到目的节点。根据 2.2 节, 若节点间的唤醒周期不同, 则数据包转发就会产生延时。

综上所述, ETX 的测量是建立在节点同步的情况下, 节点的唤醒周期和唤醒时段都是相同的, 然而实际的 WSN 中的节点都是异步工作的可能性更大, 此时节点的唤醒周期和唤醒时段就会不同, 数据传输过程会存在延时。为了使各节点之间的数据传输更加实时可靠, 可以使节点在选择邻居节点作父亲节点时, 将延时因素加入到 ETX 来保证数据传输的高成功率和低延时。

3.1 改进的单条链路延时

在 ETX 中加入延时的考虑后, 根据图 2, 改进的 ETX 算法就是计算发送数据包前 ($ETX-1$) 次未成功发送的延时和第 ETX 次成功发送的延时, 那么此时计算 A 和 B 节点间延时 D 的情况分为以下两种。

(1) A 与 B 的唤醒周期 C_T 相同。这种情况下, 假设 $ETX_{A,B}$ 为 A 和 B 之间的数据包预计发送次数, $C_{T,A} = C_{T,B}$, $\Phi_A - \Phi_B$ 是一个常数, 因此 D 可以根据式(5)计算出来。

$$D = \begin{cases} \Phi_A - \Phi_B + (ETX_{A,B} - 1) * C_{T,A}, & \Phi_A - \Phi_B > MFT \\ \Phi_A - \Phi_B + C_{T,A} * ETX_{A,B}, & \Phi_A - \Phi_B \leq MFT \end{cases} \quad (5)$$

(2) A 与 B 的唤醒周期 C_T 不相同。在这种情况下, 假设 $ETX_{A,B}$ 为 A 和 B 之间的数据包预计发送次数, $C_{T,A} \neq C_{T,B}$, $\Phi_A - \Phi_B$ 不是一个常数, 那么在 B 成功传输一个数据包给 A 前会传输 ($ETX_{A,B} - 1$) 次, 所用时间为 ($ETX_{A,B} - 1$) * $C_{T,A}$, 此时 D 就由式(6)计算。

$$D = C_{T,A} / 2 + MFT + C_{T,A} * (ETX_{A,B} - 1) \quad (6)$$

3.2 改进的路由度量

在 3.1 节中, 计算了单条链路上的路由度量, 在一个 DODAG 中, 每个节点通过路由度量与根节点建立路由, 则其他节点到根节点的总路由度量 (这里定义为 ETX_D) 可用式(7)计算得出。

$$ETX_D = \begin{cases} 0, & \text{如果节点是根节点} \\ ETX_D_p + D_p, & ETX_D_p + D_p < D_{\max} \\ D_{\max}, & ETX_D_p + D_p \geq D_{\max} \end{cases} \quad (7)$$

其中, ETX_D_p 由父亲节点通告, D_{\max} 为阈值上限。ETX 的计算可通过 2.1 节的方法实现, 唤醒周期 C_T 和唤醒时段 Φ

可通过特定的单播探测数据包从邻居节点处获取,当节点唤醒周期相同时,可用式(5)计算出 D_p , 周期不同则采用式(6)进行计算。

3.3 改进路由度量分析

与标准的 ETX 相比,改进的路由度量在保证了标准 ETX 的特点之外,还进一步增加了对延时的考虑。

举例说明,链路理想的环境中数据包传输成功率为 100%,有 A、B、C 3 个节点,B 从 A、C 之间选择父亲节点,A、B、C 的唤醒周期分别为 1s、2s、3s,则测量 ETX 采用的广播探测数据包的周期可以采用最小公倍数法计算,即每 6s 广播一次。根据标准的 ETX 算法,AB 和 BC 之间的 ETX 都为 1,那么 B 会从 A、C 随机选择父亲节点。但是根据改进的算法,若 A、B、C 同时启动,B 会在 2s、4s、6s 时唤醒发送数据包,A 则会在 1s、2s、3s、4s、5s、6s 时唤醒接收数据包,C 则是 3s、6s 时唤醒接收数据包,那么,B 在 2s 时发送数据包,忽略数据包传输时间,A 可以马上接收到,但是 C 不会,此时 B 需要等待 1s 的时间才能保证 C 接收到数据包,所以此时 B 选择 A 作为父亲节点更为合适,而且根据式(6),B 确定会选择 A 作为父亲节点。但是若采用标准的 ETX 算法,B 也会选择 C 作为父亲节点,此时就会产生较大的延时。

4 仿真实现及其结果

本文将改进的 ETX 算法在 contiki2.7 上实现并且在 cooja 仿真器上进行模拟,并与标准的 ETX 和普通延时进行了对比。测试的节点有 4 种,1 号节点作为 root 节点一直保持唤醒状态,2-9 号节点的唤醒周期为 0.1s,11-13 号节点唤醒周期为 0.2s,14-17 号节点唤醒周期为 0.3s,节点位置信息如图 4 所示,两节点间的连线表示这两节点可以建立连接。

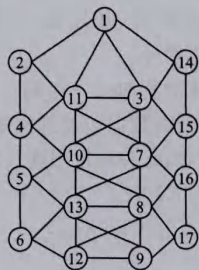


图 4 设计的节点位置

在无损链路状态下,cooja 仿真位置如图 5 所示。

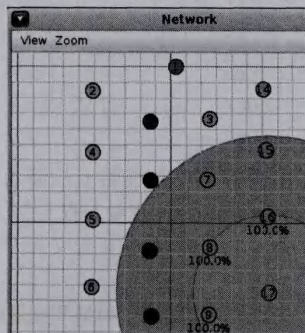


图 5 无损链路状态下的节点位置

无损状态下延时 DELAY、ETX 和改进算法 ETX_D 所构建的网络拓扑分别如图 6-图 8 所示。

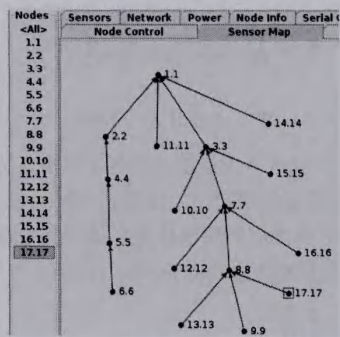


图 6 延时 DELAY 在无损链路状态下形成的拓扑

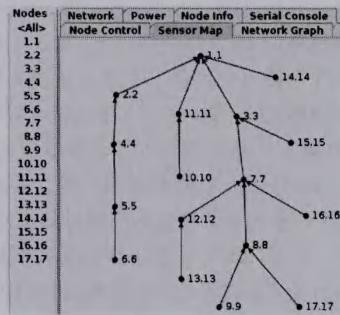


图 7 ETX 在无损链路状态下形成的拓扑

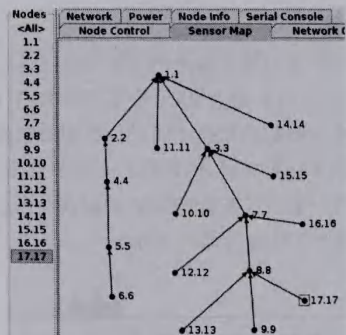


图 8 改进算法 ETX_D 在无损链路状态下形成的拓扑

以图 6 为例,左边为布置节点位置的窗口,其中以 17 号节点为圆心的内圆表示该节点的传输距离,右图是以 1 号节点作为 root 节点,与其他节点通过采用延时路由度量形成网络拓扑结构,其中 1.1 表示 1 号节点,其他节点表示方法类似。节点采用不同的路由度量计算方法,最后形成的网络拓扑结构也会不同。

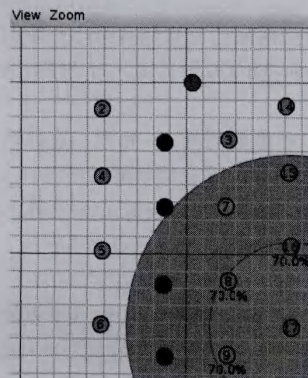
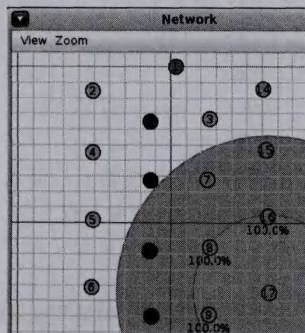


图 9 有损链路状态下的节点位置

在有损链路状态下,节点位置如图 9 所示,采用延时 DE-

LAY、ETX 和改进算法 ETX_D 所构建的网络拓扑分别如图 10—图 12 所示。

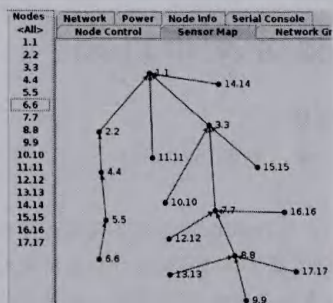


图 10 延时 DELAY 在有损链路状态下形成的拓扑

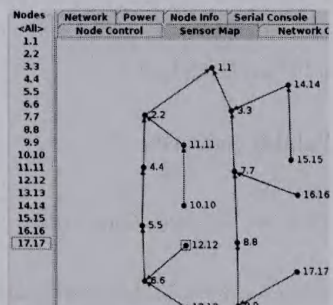


图 11 ETX 在有损链路状态下形成的拓扑

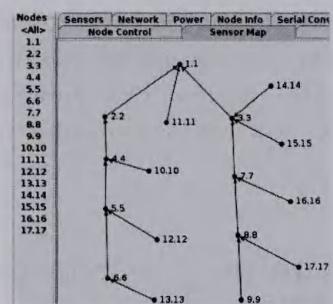


图 12 ETX_D 在有损链路状态下形成的拓扑

4.1 拓扑分析

在无损链路的情况下,数据的收发成功率为 100%,即 $ETX=1$,不存在丢包情况。根据式(5)和式(6),改进的 ETX 算法(这里命名为 ETX_D)退化为计算延时,所以 ETX_D 和延时 DELAY 形成的拓扑结构相同,所以图 6 采用延时作为路由度量形成的路由拓扑与图 8 采用 ETX_D 形成的路由度量形成的拓扑结构相同。根据延时路由度量计算,节点会寻找根节点耗时最少的路径,根据式(7),2、11、3、14 号节点会选择 1 号根节点作为父亲节点,因为 1 号节点通告的 ETX_{Dp} 值为 0,则它们自身到根节点的延时为 D_p ,由式(5)或式(6)进行计算。因为 3 号节点的唤醒周期比 10 号节点的小,由式(7)也可算出 10 号节点选择 3 号节点作为父亲节点,结果如图 6 或图 10 所示。图 7 中,在采用 ETX 作为路由度量的情况下,每个节点都是根据周期性处理探测数据包来计算链路的 ETX 值,由于 3 号节点需要处理包括 10 号节点在内的多个子节点的数据包,而 11 号节点只需要处理 10 号节点的数据包,因此 11 号节点处理速率比 3 号节点快,故 10 号节点会先选择 11 号节点作为父亲节点。以上也说明采用 ETX 作为路由度量的 WSN 会避开拥塞的密集网络,所以图

7 中,10 号节点选择 11 号节点作为父亲节点,13 号节点选择了 12 号节点作为父亲节点。

在有损链路的情况下,数据的收发成功率不为 100%,即 $ETX \geq 2$,存在丢包的情况。可以看出,图 6 和图 10 中根据延时建立起来的网络拓扑并没有变化,因为以延时 delay 作为路由度量,是以数据收发成功率为 100%作为前提的,并且只用获取邻居节点的唤醒周期 C_T 和唤醒时段 Φ ,唤醒周期 C_T 和唤醒时段 Φ 都是恒定值, MFT 是设置的常量。以上的节点参数都是恒定值,节点为了获取这些计算延时所需要的参数,只要能够接收到邻居节点广播的一个探测包就可以获取邻居节点的参数,从而计算出自身与该邻居节点之间的延时度量,再选择父亲节点。所以延时度量在每种链路情况下得出的拓扑都是一样的。根据 ETX,除了前面所说的数据包处理速率问题外,在有损网络中还会产生丢包问题,多个子节点向同一个父亲节点传送数据包会产生碰撞,从而出现丢包问题,根据 2.1 节,网络密集时计算出来的 ETX 会比稀疏网络中的 ETX 大,因此采用 ETX 作为路由度量的 WSN 形成的网络拓扑会比较稀疏,因此在图 11 中,11 号节点选择 2 号节点作为父亲节点,与图 11 的情况不同的图 10 中,11 号节点只考虑延时,不考虑网络密度对数据传输成功率的影响,因此 11 号节点选择 1 号作为父亲节点,在图 10 和图 11 中的 14 号节点选择父亲节点的方法与 11 号节点相同。根据 ETX_D,从式(5)和式(6)可以看出,ETX 是计算 ETX_D 的一个影响因素,根据前面所述的 ETX 的特点,网络拓扑会比较稀疏,这样在其他节点成功发送一个数据包给 root 节点时,发送该数据包的次数会比较少,但是不一定传输次数少,其他节点成功发送数据包到 root 节点的延时就低,在图 12 中,11 号节点没有同图 11 一样为了避开网络拥塞去选择 2 号节点作为父亲节点,而是选择了 1 号节点作为父亲节点,同理 15 号节点选择了 3 号节点作为父亲节点。以 11 号节点为例,根据式(6)和式(7),假设 11 号节点与 1 号节点间的 ETX 为 4,计算出来的 ETX_D 为 MFT ,11 号节点与 2 号节点的 ETX 为 2,2 号节点与 1 号节点的 ETX 为 1,计算出来的 ETX_D 为 $2MFT + 3/2 * C_{T,2}$,其中 $C_{T,2}$ 表示 2 号节点的唤醒周期,因此很显然 11 号节点会选择 1 号节点作为父亲节点。

以上是结合理论对形成的拓扑进行的说明,后面的内容将通过已形成拓扑,统计其他节点成功发送数据包给 root 节点的延时,对改进算法加以验证。

4.2 延时比较

为了计算其他节点到 1 号 root 节点的平均延时,本文使每个其他节点向 root 节点成功发送 1000 个数据包,并记录每个数据包的延时后算出总的延时。图 13 中给出了图 6—图 8 的网络拓扑中的每个非 root 节点向 1 号 root 节点成功发送数据包的平均延时,在图 14 中给出了图 10—图 12 下的平均延时。

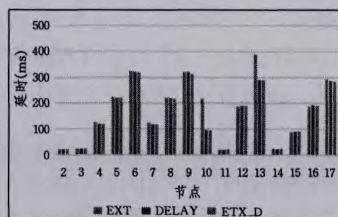


图 13 理想链路状态下的延时比较

(下转第 133 页)

古天龙,何仲春,常亮,等.基于符号 ADD 和线性多分支程序的分类型安全评估[J].电子学报,2014,42(5):940-947

[20] Goldreich O. Foundations of cryptography [M]. Beijing: Publishing house of electronics industry, 2003(in Chinese)
Goldreich O. 密码学基础:英文版[M].北京:电子工业出版社, 2003

[21] Canetti R. Studies in secure multiparty computation and applications[D]. The Weizmann Institute of Science, 1996

[22] Rabin M O. How To Exchange Secrets with Oblivious Transfer [J]. IACR Cryptology ePrint Archive, 2005: 187

[23] Paillier P. Public-key cryptosystems based on composite degree

residuosity classes [C] // Advances in Cryptology—EURO-CRYPT'99. Springer Berlin Heidelberg, 1999: 223-238

[24] Lai Y T, Pedram M, Vrudhula S B K. EVBDD-based algorithms for integer linear programming, spectral transformation, and function decomposition [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1994, 13(8): 959-975

[25] Gu Tian-long, Xu Zhou-bo. Ordered binary decision diagram and its application [M]. Beijing: Science Press Ltd., 2009: 17-57 (in Chinese)
古天龙,徐周波.有序二叉树决策图及应用[M].北京:科学出版社, 2009: 17-57

(上接第 105 页)

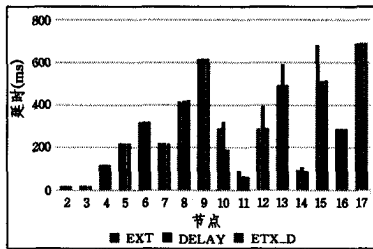


图 14 有损链路状态下的延时

从图 13 中可以看出,理想链路状态下,由于 ETX 并没有考虑延时的原因,并且形成的网络拓扑是比较稀疏的,图 7 中 10 号节点选择了唤醒周期较长的 11 号节点作为父亲节点, 13 号节点选择了 12 号节点作为父亲节点,并没有同图 6 和图 8 中的拓扑一样去选择唤醒周期较小的父亲节点: 3 号和 8 号节点,因此在采用 ETX 作为路由度量形成的网络拓扑中, 10 号节点成功发送数据包给 1 号 root 节点会比采用延时和改进算法 ETX_D(见图 6、图 8)作为路由度量时所用的时间长。

有损链路的状态下,对于 13 号节点,由于延时路由度量并没有考虑数据包收发成功率的因素,因此图 10 和图 6 中, 13 号节点都选择 8 号节点作为父亲节点,而对于考虑到了数据包收发成功率因素(即最少发送次数)的 ETX 和 ETX_D, 图 11 和图 12 中, 13 号节点则选择了 6 号节点作为父亲节点。在有损链路中,由于存在丢包问题,因此成功发送一个数据包与发送该数据包的次数有关,较少的发送次数也会降低延时,故在采用 ETX 和 ETX_D 作为路由度量形成的网络拓扑(见图 11、图 12)中, 13 号节点成功发送数据包给 1 号 root 节点会产生较小的延时,如图 14 中给出的 13 号节点的延时比较所示。对于 10 号节点,比较延时路由度量(见图 10)和 ETX_D 路由度量(见图 12),由于网络拥塞和丢包会影响延时,因此采用 ETX_D 的 10 号节点并没有选择 3 号节点作为父亲节点,而是选择子节点较少的 4 号节点作为父亲节点,比较 ETX 路由度量(见图 11)和 ETX_D 路由度量(见图 12),由于 10 号节点到 11 号节点的延时比 10 号节点到 4 号节点的延时长,并且 3 个节点间的 ETX 值也并不相同,因此图 12 中的 10 号节点会选择 4 号节点作为父亲节点,如图 14 中给出的 10 号节点在各情况下的延时比较,在采用 ETX_D 作为路由度量形成的网络拓扑中, 10 号节点成功发送数据包给 1 号 root 节点所产生的延时最小。

结束语 本文介绍了一种改进的 ETX 算法,同时在 cooja 模拟器上给出了仿真的结果,并与普通延时和标准的 ETX 进行

了比较。实验结果证明,改进的 ETX 算法更具有实际应用意义,在有损链路并且具有不同唤醒周期节点的 WSN 中,该算法可以使成功传输数据包所产生的延时最小。但也应该看到,本文给出的路由度量的计算比较简单,提出的影响延时的因素也比较少,更为详细和完善的设计方案还有待研究。

参考文献

[1] Dunkels A, Gronvall B, Voigt T. Contiki-a Lightweight and Flexible Operating System for Tiny Networked Sensors[C]// Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks. 2004

[2] Hu Qin-yan, Yin Chang-chuan. Research RPL routing protocol for wireless sensor networks[J]. Things of Technology, 2014 (1): 57-62(in Chinese)
胡芹艳,尹长川.无线传感网络中的 RPL 路由协议研[J].物联网技术, 2014(1): 57-62

[3] Winter T, Thubert P, Brandt A, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks[S/OL]. RFC 6550 (Proposed Standard), Mar. 2012. <http://www.ietf.org/rfc/rfc6550.txt>

[4] Vasseur J P, Kim M, Pister K, et al. Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks[OL]. <http://www.ietf.org/rfc/rfc6551.txt>

[5] De Couto D S J, Aguayo D, Bicket J, et al. A highthroughput path metric for multi-hop wireless routing[C]// Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom'03). San Diego, California, September 2003

[6] Osterlind F, Dunkels A, Eriksson J, et al. Crosslevel sensor network simulation with cooja[C]// Proceedings 2006 31st IEEE Conference on Local Computer Networks. Nov. 2006: 641-648

[7] Thubert P. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL) [S/OL]. RFC 6552 (Proposed Standard), Internet Engineering Task Force, Mar. 2012. <http://www.ietf.org/rfc/rfc6552.txt>

[8] Gnawali O, Levis P. The Minimum Rank with Hysteresis Objective Function[S/OL]RFC 6719 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. <http://www.ietf.org/rfc/rfc6719.txt>

[9] Gonizzi P, Monica R, Ferrari G. Design and Evaluation of a Delay-Efficient RPL Routing Metric[C]// Department of Information Engineering University of Parma Parco Area delle Scienze 181/A Parma. Italy, July 2013: 1573-1577