

EasiCC :一种保证带宽公平性的传感器网络拥塞控制机制

鞠海玲^{1,2} 崔 莉¹ 黄长城¹

¹(中国科学院计算技术研究所 北京 100080)

²(中国科学院研究生院 北京 100049)

(lcui@ict.ac.cn)

EasiCC :A Congestion Control Mechanism for WSN

Ju Hailing^{1,2}, Cui Li¹, and Huang Changcheng¹

¹(*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080*)

²(*Graduate University of Chinese Academy of Sciences, Beijing 100049*)

Abstract It's necessary for wireless sensor network (WSN) applications to deal with network congestion, because the channel bandwidth of WSN is usually narrow. However, existing congestion control mechanisms don't run well on WSN platforms. A practical congestion control mechanism for WSN should not only have good network performance but also have little control cost, the proposed EasiCC (EasiNet congestion control mechanism) is such a congestion control mechanism. In EasiCC, the source nodes prorate data packets into several transporting priorities, all the network nodes adjust in-phase packet filtering threshold to adapt network congestion status. With the help of filtering threshold and packet priority, network bandwidth is fairly allotted among data streams. EasiCC uses a stepwise and exponential method to adjust network traffic, so as to reduce control messages between network nodes. EasiCC uses network access waiting time and packet queue overflowing to detect network congestion, uses network admittance and queue dropping simultaneously to ensure the integral network performance. EasiCC has been implemented in wireless sensor network test-bed and costs little on the network bandwidth, communication energy and node memory. Simulation and experimental results indicate that EasiCC can limit transmission delay effectively, reduce packet loss ratio remarkably, and provide bandwidth fairness between data streams at the same time.

Key words wireless sensor networks; congestion control; bandwidth fairness; packet priority; network admittance

摘 要 实用的传感器网络拥塞控制方案不仅需要满足多项网络性能指标,而且必须控制开销很小,提出了一种满足上述要求的拥塞控制机制 EasiCC (EasiNet congestion control mechanism)。在 EasiCC 中,数据流源节点将数据报文按比例划分到各优先等级中,各网络节点根据网络拥塞程度动态地、同步地调整报文过滤标准,结合报文过滤标准和报文优先级来调节网络流量,保证了无线信道带宽分配上的公平性,将网络准入控制和队列丢包手段相结合来调整网络流量,保证了网络综合性能指标。EasiCC 控制开销很少,已在实际传感器网络平台中实现。模拟验证和实验测试结果显示, EasiCC 能够公平地为各数据流分配发报速度和网络带宽,并且在报文传输成功率、传输延迟等性能指标上均有良好的表现。

关键词 无线传感器网络;拥塞控制;带宽公平性;报文优先级;网络准入

中图法分类号 TP393

1 引言

1.1 应用场景

本文提出的拥塞控制机制 EasiCC (EasiNet congestion control mechanism) 可应用于环境监测传感器网络。环境监测传感器网络是使用最广泛、也最典型的一类传感器网络 (WSN)。这类网络由位置固定的网络节点组成, 以自组织方式进行动态组网, 路由变化速度也比较缓慢; 各节点周期性地将传感数据报文通过无线网络向基站发送, 网络流量主要来自周期性的数据流。在以下情况下可能出现网络拥塞现象: 1) 当所观测环境发生重要变化时, 有很多传感器节点将适当加快采样数据和发送数据报文的频率; 2) 传感器节点通信模块所支持的带宽通常较窄, 相邻节点之间需要竞争使用无线共享信道, 因而无线网络增大到一定规模时更容易出现网络拥塞现象。在发生上述情况时进行拥塞控制, 减少网络丢包率对于环境监测传感器网络的实用性至关重要。

本文研究目标是提出一种适合 WSN 的拥塞控制机制及其实现方法 (EasiCC)。下面首先介绍其评价指标、相关研究工作和本文研究贡献。

1.2 评价指标

WSN 拥塞控制技术有多个评价指标: 1) 在数据报文传输成功率方面, 为了避免浪费通信能量, 希望减少传输途中丢包数目; 2) 在控制报文开销方面, 为了节省通信能量和带宽资源要限制控制报文的用量, 特别是减少周期性控制报文的用量; 3) 在带宽公平性方面, 重要性相同、但是产生于不同节点的数据流, 它们每秒钟应该有相同或者相近数目的报文成功到达基站; 4) 在传输延迟方面, 从排队理论可知当网络信道过于拥挤时报文传输延迟也过大, 拥塞控制方案应该既保证较高信道利用率又保证较小报文传输延迟。

1.3 相关研究

由于针对的应用场景不同, 不同 WSN 拥塞控制方案侧重的指标不同, 采用的拥塞发现和抑制方法也有所差异。

在 CODA^[1] 拥塞控制方案中, 每个数据源节点动态跟踪测量无线信道忙闲情况, 发现信道拥塞后会降低发包速度; 在网络带宽利用率超过某个规定阈值^[1] 时, 各发送节点如果不能收到接收节点周期性反馈的确认消息报文也会认为发生拥塞而自动降

低发包速度。CODA 的优点是能根据信道状态及时调节数据流速度, 缺点是连续监听信道以及发送周期性反馈报文都需要较大通信能量开销。

ESRT^[2] 拥塞控制算法寻找一个最佳工作点, 在保证一定频度的传感器数据报文顺利到达基站的前提下, 能尽量减少源节点发包速度、节省通信能量。但是, ESRT 关心的是与一定事件相关的数据报文到达基站的总体速度以及节省通信能量, 而不关心节点间的带宽公平性。

文献 [3] 和 TinyOS^[4] 的应用程序 Surge 中都使用了分布式算法来控制各节点产生和转发报文的速度。在文献 [3-4] 方法中, 各节点仅使用节点本地信息来进行拥塞发现和流量抑制, 其好处是不需要反馈拥塞控制报文, 缺点是模拟和实验数据显示很大比例的数据报文在转发途中因拥塞被丢掉, 浪费了通信能量和带宽资源。模拟数据还显示, 不同数据流的报文成功率不同, 数据流间带宽分配不公平。

文献 [5] 中的拥塞控制方案所针对的是 Many-to-one 数据流, 并且将公平性作为重要评价标准。每个网络节点动态统计自己当前可用信道带宽和子孙节点数目, 据此计算出各数据流速度上限, 并将速度上限值通过消息报文通知子孙节点; 每个节点由各级祖/父节点和自己计算的上限值中取出最小的一个作为今后数据流速度。在文献 [5] 的模拟实验和实际平台测试中都取得了很好带宽公平性, 但是其数据报文的平均重传率超过 2, 说明其拥塞控制算法还需要改进。

Internet 上针对 TCP 流量的拥塞控制研究成果层出不穷, 不断有文章从业务流特点、网络环境特点、公平性等方面出发对 TCP 拥塞控制提出改进措施^[6-8]。TCP 数据流通常是双向的, TCP 拥塞控制解决方案普遍借助反向流量将网络拥塞状态源源不断地反馈给数据流源节点, 源节点则根据各种策略动态调整数据流发送速度。自从文献 [9] 发表以来, 研究界把网络流量管理/拥塞控制问题抽象为线性优化问题, 并使用分布式控制理论来分配网络带宽资源, 这方面近年来出现了很多研究成果^[10-12]。但是, Internet 上 TCP 拥塞控制的成熟解决方案很难照搬到 WSN 中, 一方面因为 WSN 中主要业务流是从各传感器节点通往基站的单向流量, 不能像 TCP 协议那样利用反向流量频繁地向数据流源节点反馈网络拥塞状态; 另一方面因为 WSN 中网络节点的内存和处理能力都比 Internet 网络节点小很多, 不能执行过于复杂的算法。

门限值为多少,每个数据流都有一定比例的报文被传输到达基站,从而保证了数据流间的公平性。

节点门限值的维护方法是:1)节点在初始化时门限值被标定为0;2)当发现网络拥塞时就逐渐增加各节点的门限值,节点门限值每增加1,网络流量就减少一半,由于门限值的过滤,各节点发送的报文数目将逐渐减少,网络拥塞将得到缓解,直至拥塞完全解除后各节点门限值才稳定下来;3)当网络信道利用率大幅度下降时,全网节点可以同步降低门限值(最低降到0),节点门限值每减少1,网络流量就增加一倍,从而间接地将信道利用率恢复较高水平。

EasiCC通过控制节点门限值的调节速度,实现了快减慢增的流量调整效果。在EasiCC中,每当发现拥塞时节点门限值能快速增加到拥塞消失为止;当没有拥塞时,每经过较长一段时间(如数百个报文发送周期)各节点才各自尝试将门限值减1。如果减少门限值后没有引发拥塞,则各节点就稳定在新门限值上;如果减少门限值后网络出现拥塞,则会很快通过拥塞调控机制恢复原先的门限值。

EasiCC中每个节点需要维护本节点门限值以及下一个本地报文的序列号,而不需要为流经自己的各数据流维护状态信息。因此EasiCC的内存开销与网络中节点数目、数据流数目无关,网络规模的可扩展性非常好。

2.2.3 数据流带宽权值的标定

在EasiCC中,每个数据流都有一个带宽权值(weight)的属性,其取值范围是 $[1, 128]$ 区间的整数,缺省值为1。定义带宽权值为1数据流的报文周期作为一个标准周期,则一个标准周期内带宽权值为 W 的数据流将产生 W 个数据报文。在EasiCC中,各节点比较标准周期(而不是自己实际的报文周期)与发送队列等待时间来判断拥塞情况:如果一个传感器节点产生的数据流带宽权值为 W ,则该节点中允许发送队列积压的本地报文数目最大为 W ;每当节点产生一个新的本地报文时,如果发送队列中积压的报文数目已经达到允许的最大值则说明队列等待时间超过了标准周期,因此认为发生了一次拥塞并丢弃新产生的报文。环境监测网络经过一段时间的运行后,位置相邻各节点的门限值将趋于一致;如果两个数据流的带宽权值相差 W 倍,则这两个数据流向网络注入报文的速率将相差约 W 倍;而带宽权值相同的数据流向网络注入报文的速率也就趋于一致。

EasiCC给应用层程序提供了一个配置接口,允

许根据具体应用的特点来动态调整数据流的带宽权值。比如,在森林防火应用中,那些低温地理区域内产生的数据流带宽权值可以设为1,而那些高温、火警区域内产生的数据流带宽权值可以设为5;当一个地理区域内由高温逐渐转为低温时,该区域内数据流带宽权值也就自动地由5逐渐降为1。

2.2.4 控制消息报文的使用

在无线网络中,节点间共享无线信道,一个节点如果过快地发送报文将会导致其邻近节点都感觉到信道拥塞,因此公平的方法是让相互邻近节点间的门限值接近、占用信道的程度接近。EasiCC中,各节点每次调整自己的门限值时,都会使用广播消息报文将自己最新门限值向所有邻居节点广播。每个节点收到一个相邻节点发来的消息报文后,将相应调整自己的门限值,向邻居节点的最高门限值看齐。

节点门限值的取值空间越大,那么为了让所有节点同步到同一个门限值所需要控制消息量就越多。因此,EasiCC为了减少节点间控制消息报文的使用次数,对流量进行的是粗粒度阶梯式调节,并将门限值取值范围限制得很小,缺省情况下定义为0~8的整数。

2.3 EasiCC的适用性

EasiCC最适合于数据流量和网络拓扑变化不频繁的无线网络应用场景。EasiCC利用了网络流量稳定性,根据历史流量情况来预测未来网络流量、分配网络带宽,从而在保证网络性能指标的同时减少控制开销。本文引言中介绍的环境监测传感器网络符合上述特点,所以是EasiCC的典型应用场景。

3 模拟验证

3.1 模拟环境

本文第3.2~3.4节对3种能够运行于WSN节点的拥塞控制算法进行了对比测试。这3种算法是:1)EasiCC算法,综合使用队列等待时间和队列溢出两种方法监测拥塞情况,并且通过控制消息报文协调各节点的操作;2)TinyOS算法,综合使用队列等待时间和队列溢出方法监测拥塞情况,各节点分别独立地调节流量;3)DropTail算法,仅仅使用队列溢出作为拥塞发现和流量调节措施。对这3种算法进行对比测试时,让每个网络节点只维护通向0号节点(基站)的路由,这样更逼近于WSN应用场景。

本文第3.5节对EasiCC和Internet上TCP拥塞控制算法做了对比测试,TCP算法中需要向各个

数据流源节点反馈消息报文,所以 EasiCC 和 TCP 对比测试时,每个网络节点仍然维护到全网节点的路由。

表 1 示出使用了 ns2 网络模拟平台的主要配置参数。如表 2 所示,模拟中用到 2 种规模的网络拓扑。为每种规模的拓扑产生 3 种随机的节点布局和 8 种初始流速度(应用层产生报文的速率),下面我们(拓扑规模编号,节点布局编号,初始流速度)的三元组来简称某个特定场景。除去节点 0 本身,每个节点均产生一个流向节点 0 的数据流。当所有数据流的初始流量超过网络总容量,数据流的速度受拥塞控制算法作用而自适应下降。在获取每组测试数据时,为了区分路由原因和拥塞原因对网络传输性能的影响,等模拟运行 100s 后路由由状态基本稳定,所有节点才开始产生和发送报文。TCP 算法要求数据报文采用 TCP 协议格式,EasiCC,TinyOS,DropTail 算法则要求数据报文采用 UDP 报文格式。

Table 1 Configuration of Simulation Parameters

表 1 模拟参数配置

Parameter	Value
Packet Load Size (B)	50 (application layer load)
Mac Frame Size (B)	122 (for UDP), 144 (for TCP)
Duration of Streams (ms)	900
Network Protocols	802.11 DCF, DSDV, CBR, UDP/TCP, IP, ARP
Packet Queue Size (packet)	50
Channel Bandwidth (Mbps)	1

Table 2 Configuration of Simulation Scenes

表 2 模拟场景

Topology No.	Scene Size (m ²)	Node Positions	Node Sum
Random_50	1000 × 1000	Randomly Placed	50
Random_200	2000 × 2000	Randomly Placed	200

3.2 模拟验证结果 1: 报文成功率和控制报文开销

图 2 展示了不同拥塞控制算法下的端对端报文传输成功率(End-to-end PSR)情况。2 个子图分别对应 2 种网络规模下的测试结果。图 2 中每个数据点代表了在某特定初始流速度、特定网络规模、特定拥塞控制算法下由式(1)综合 3 种不同节点布局下模拟数据计算出的统计值。因为不同规模网络能够容纳的数据流最大速度(临界流速度)不同,所以测试不同规模网络时采用了不同的数据流初始速度集。

$$PSR = \frac{N_{arrived}}{N_{originated}}, \quad (1)$$

式(1)中, $N_{arrived}$ 表示到达目的节点的报文总数, $N_{originated}$ 表示源发的报文总数。

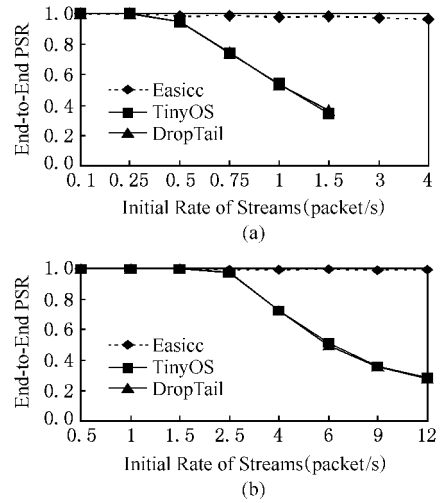


Fig. 2 End-to-end PSRs for congestion control algorithms. (a) Simulation results in scene random_200 and (b) Results in scene random_50.

图 2 不同拥塞控制算法作用下的报文传输成功率。(a)场景 Random_200 下的结果 (b)场景 Random_50 下的结果

从图 2 看出,在初始流速度较小时(各子图中横坐标前 3 个取值),3 种拥塞控制算法都能保证很高的端对端报文成功率;当初始流速度逐步升高时,EasiCC 的拥塞控制算法一直能保证 97% 以上的报文成功率,其他两种拥塞控制算法则性能快速下滑,报文成功率降到了 50% 以下。当网络规模和初始流速度均很大时,TinyOS 和 DropTail 两种拥塞控制算法不能有效抑制数据流量,严重降低了路由报文的传输成功率,并导致路由协议崩溃、测试过程终止,因此图 1(a)中没有绘制两种算法在两个最大初始流速度下的测试结果。

EasiCC 算法需要通过控制消息报文协调各节点的操作,表 3 是根据模拟结果和式(2)计算出的 EasiCC 控制报文开销(control expending)。

$$Expending = \frac{N_{control}}{N_{arrived}}, \quad (2)$$

式(2)中, $N_{control}$ 表示发送的控制报文总数; $N_{arrived}$ 表示到达目的节点的报文总数。

从表 3 看出,在初始流速度较小时没有控制报文开销;当初始流速度逐步升高时控制报文开销逐步增加;在所有测试场景中,控制开销几乎都在 1% 以内。与报文传输成功率方面的收益相比,EasiCC 的微小控制开销是可以忽略的。

Table 3 Control Packet Expending of EasiCC
表 3 EasiCC 的控制报文开销

Random_200		Random_50	
Initial Rate (packet/s)	Control Expending	Initial Rate (packet/s)	Control Expending
0.10	0	0.50	0
0.25	0	1.00	0
0.50	0.000906	1.50	0
0.75	0.003015	2.50	0.000742
1.00	0.005315	4.00	0.000571
1.50	0.005934	6.00	0.001503
3.0	0.008936	9.00	0.001656
4.0	0.010188	12.00	0.002161

3.3 模拟验证结果 2 :数据流间带宽公平性

图 3 展示了不同拥塞控制算法下各数据流(根据源节点划分数据流)成功传输报文数目(success packet per stream)情况. 图中, 均值和均方差分别用柱形和“T”形的高度表示.

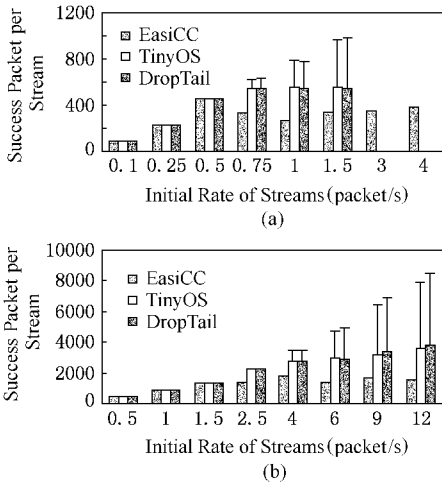


Fig. 3 Average and deviation of success packets per stream. (a) Simulation results in scene (Random_200, 1, *) and (b) Results in (Random_50, 1, *)

图 3 各数据流成功报文数目的均值及均方差. (a)场景 (Random_200, 1, *) 的结果 (b)场景 (Random_50, 1, *) 的结果

从图 3 看出, 在初始流速度较小时信道比较空闲, 所有拥塞控制算法各数据流成功报文数目的均方差都非常小. 当初始流速度逐步升高时, EasiCC 算法数据流成功报文数目的均方差一直很低, 而 TinyOS 和 DropTail 两种算法均方差很大.

图 4 绘制了各数据流成功到达目的节点的报文数目, 是在 50 个节点网络规模、1 号节点布局、

12packet/s 初始流速度的模拟场景下统计的. 从图 4 看出, EasiCC 算法可以保证每个数据流都能有相近数目的成功报文、带宽分配公平, TinyOS 算法各数据流成功报文的数目差别非常大.

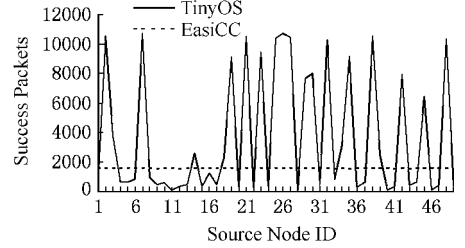


Fig. 4 Success packets for streams in scene (Random_50, 1, 12).

图 4 场景 (Random_50, 1, 12) 的各数据流成功报文数目

图 5 是各个控制算法成功到达目的节点数据报文的平均路由跳数, 是在 50 个节点网络规模、1 号节点布局、8 种初始流速度场景下根据式 (3) 统计出来的. 从图 5 看出, 当初始流速度逐步升高时, EasiCC 算法成功报文的平均路由跳数变化不大, 而 TinyOS 和 DropTail 两种算法成功报文的平均路由跳数则逐步变小. 结合 3~5 可以推断, 当初始流速度逐步升高时, TinyOS 和 DropTail 两种算法中那些路由距离短的报文在成功报文中的比例逐步上升, 换句话说, 那些路由距离长的报文在带宽分配上受到了歧视.

$$\overline{hop} = \frac{\sum_{i \in \Omega} hop(i)}{size(\Omega)}, \quad (3)$$

式 (3) 中, Ω 表示到达目的节点的报文集合; $hop(i)$ 表示报文 i 的路由跳数; $size(\Omega)$ 表示集合 Ω 所含元素个数.

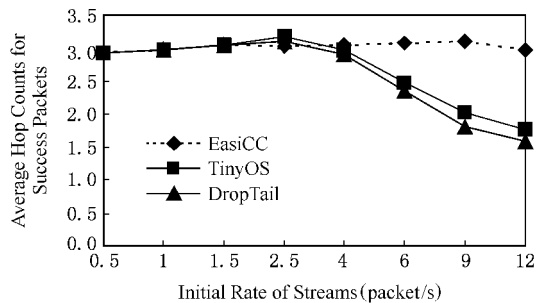


Fig. 5 Average hop counts for success packets in scene (Random_50, 1, *)

图 5 场景 (Random_50, 1, *) 成功报文的平均路由跳数

EasiCC 算法在各数据流总速度方面不是最优,

它是通过折损适量数据流总速度为代价来保证各项网络性能指标的总体平衡. 多项性能指标同时达到较高水平是对实用算法的基本要求. 特别是在 WSN 中, 为了节省通信能量必须提高报文传输成功率, 为了不漏报环境监测中的重要事件必须保证数据流间带宽分配的公平性, 因此 EasiCC 在多项指标间平衡折中的做法非常有必要.

3.4 模拟验证结果 3: 报文传输延迟

图 6 绘制了模拟过程中的报文传输延迟. 图 6 中列出的是数据报文单跳延迟的统计平均值, 而不是端对端延迟的平均值, 单跳延迟是队列等待时间和 MAC 等待时间之和. 图 6 中每个数据点代表了在某特定初始流速度、特定网络规模、特定拥塞控制算法、由式(4)综合 3 种不同节点布局模拟结果数据计算出的统计值. 从图 6 看出, 在初始流速度较小时, 所有拥塞控制算法下报文的平均单跳传输延迟都非常小; 当初始流速度逐步升高时, EasiCC 算法下单跳传输延迟变化不大, 而 TinyOS 和 DropTail 两种算法下单跳传输延迟则大幅度增大. 出现这种现象的原因一方面因为 TinyOS 和 DropTail 两种算法不能有效降低信道拥塞程度致使每个报文 mac 等待延迟增加, 另一方面是因为这两种算法不能有效降低发送队列长度导致队列等待延迟增加.

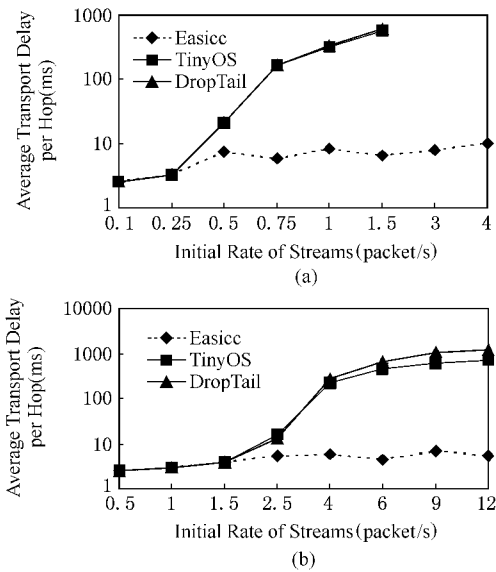


Fig. 6 Transport delay per hop for congestion control algorithms. (a) Results in scene random_200 and (b) Results in scene random_50.

图 6 不同拥塞控制算法作用下的单跳报文传输延迟. (a) 场景 Random_200 的结果 (b) 场景 Random_50 的结果

$$\overline{\text{delay}} = \frac{\sum_{i \in \Omega} \text{delay}(i)}{\sum_{i \in \Omega} \text{hop}(i)}, \quad (4)$$

式(4)中, Ω 表示到达目的节点的报文集合; $\text{delay}(i)$ 表示报文 i 的端到端传输延迟; $\text{hop}(i)$ 表示报文 i 的路由跳数.

3.5 模拟验证结果 4: EasiCC 与 TCP 性能对比

以上我们测试了 EasiCC 和 TCP 拥塞控制算法在各种场景下的性能, 下面逐一介绍它们在各性能指标方面的表现.

1) 报文成功率: EasiCC 和 TCP 算法均表现良好, 在多种模拟场景下端对端报文传输成功率都在 97% 以上.

2) 控制报文开销: EasiCC 算法显著小于 TCP 算法. EasiCC 算法中控制报文主要用于同步各网络节点门限值, TCP 算法中控制报文主要传送端对端数据报文确认信息. 在各种场景的模拟结果中, TCP 算法的控制报文使用量很大, 与数据报文传输量处于相同量级, EasiCC 算法的控制报文使用量很小, 小于数据报文传输量的 1%.

3) 带宽公平性: EasiCC 算法明显优于 TCP 算法. 图 7 中列出了这两种算法在带宽公平性方面的模拟结果, 图中每个数据点表示某特定初始流速度下各个数据流成功到达目的节点报文数目, 均值和均方差分别用柱形和“T”形图表示. 从图 7 看出, 当初始流速度逐步升高时, EasiCC 算法数据流成功报文数目的均方差一直很低, 而 TCP 算法均方差很大, 说明 EasiCC 算法在带宽公平性方面明显优于 TCP 算法.

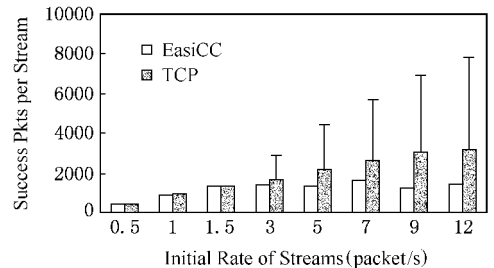


Fig. 7 Bandwidth fairness for EasiCC and TCP algorithms in scene (Random_50, 1, *).

图 7 EasiCC 与 TCP 算法在场景 (Random_50, 1, *) 下的带宽公平性

4) 单跳传输延迟: EasiCC 算法优于 TCP 算法. 图 8 绘出了这两种算法分别作用下的数据报文单跳传输延迟, 对应着 50 个节点网络规模、1 号节点布局场景的模拟结果. 对照图 8 和图 6 可以看出, 在

传输延迟方面 ,TCP 算法劣于 EasiCC 算法 ,但是优于 TinyOS 算法.

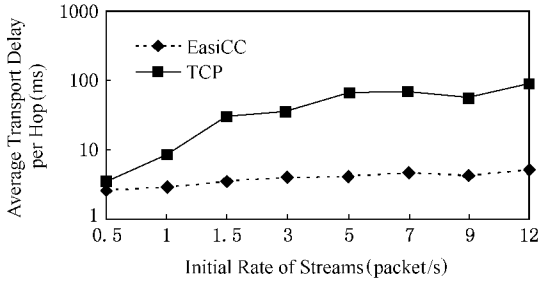


Fig. 8 Transport delay for EasiCC and TCP algorithms in scene (Random_50 ,1 , *).

图 8 EasiCC 与 TCP 算法在场景(Random_50 ,1 , *)下的传输延迟

总结前面模拟结果看出 ,在本文的模拟场景中 EasiCC 的综合性能指标优于 TCP 算法. 出现这种结果是因为两种算法针对的应用环境不同 ,TCP 协议是为 Internet 设计的 ,而 EasiCC 是针对 WSN 的无线共享信道特点设计的.

4 实验测试

4.1 实验环境

在实验测试中 ,使用了 16 个传感器节点和一个基站. 传感器节点硬件平台为 EasiNode^[13] ,无线通信芯片为 CC1000 ,节点操作系统为 TinyOS 1.1.7. 实验中对 TinyOS 1.1.7 中 MAC 层实现代码进行了如下调整 :1)采用最多退后 3 次的指数退后信道探测算法 ;2)开启报文逐跳应答功能 ,发现丢包问题时最多重发报文 5 次 ;3)TinyOS 中所有报文使用相同的固定应答码 ,而我们改用动态应答码.

我们使用了两种实验场景 :1 号场景在办公楼一个走廊中 ,使用图 9(a)的固定路由拓扑结构 ;2 号场景在办公楼 3 个走廊中 ,使用图 9(b)的固定路由拓扑结构. 在所有场景中 ,每个传感器节点不仅产生和向父节点发送本地报文 ,还将收到的数据报文通过父节点向基站转发. 数据报文长度为 17 字节.

在实际平台上对比测试了 EasiCC 和 TinyOS 两种能够运行于传感器网络节点上的拥塞控制算法. 实验测试中 ,很难精确统计报文传输延迟等性能指标 ,所以我们仅统计出节点间带宽公平性和带宽权值方面的测试结果. 每一轮实验保持相同节点摆放方式和固定路由 ,尝试 6 种不同的节点初始发包速度. 测试 EasiCC 时 ,每种初始速度下的实验细分

两个阶段 ,每阶段实验时间均为 100s ,第 1 阶段(收敛期)中各数据流的发报速度由初始值逐渐过渡到稳态值 ,第 2 阶段(稳定期)中各节点按照稳态速度发送报文. TinyOS 拥塞控制算法中不需要维护节点门限值 ,所以它的实验过程中也没有收敛期和稳定期的区别.

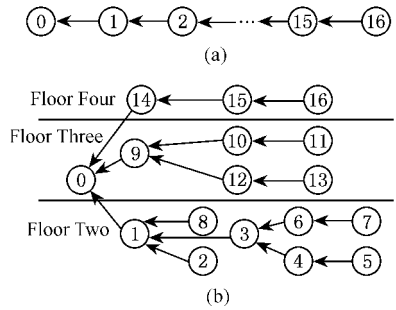
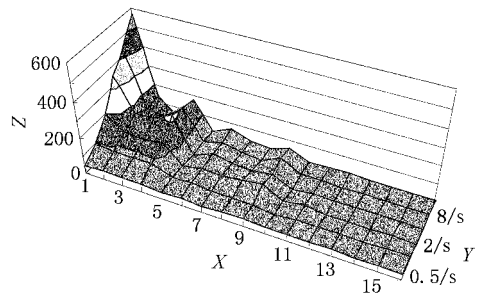


Fig. 9 Fixed topologies for field experiments. (a) Topology for scene No.1 and (b) Topology for scene No.2.

图 9 实验场景 1 2 所用的固定网络拓扑结构. (a)1 号场景结构图 (b)2 号场景结构图

4.2 实验测试结果 1 :节点间带宽公平性

图 10 是 TinyOS 拥塞控制算法在场景 1 的实验结果 ,每排 16 个样本点分别对应 16 个源节点产生并成功发送到基站的报文数目(不含转发报文数目). 从图 10 看出 ,当初始速度较低(1 个/2s)时 ,各个节点产生并成功到达基站的报文数目相近. 随着初始发包速度的增加 ,各源节点成功到达基站报文数目差异变大. 当初始速度调整为 16 个/s 时 ,不同节点间到达基站的报文数目相差达十几倍.

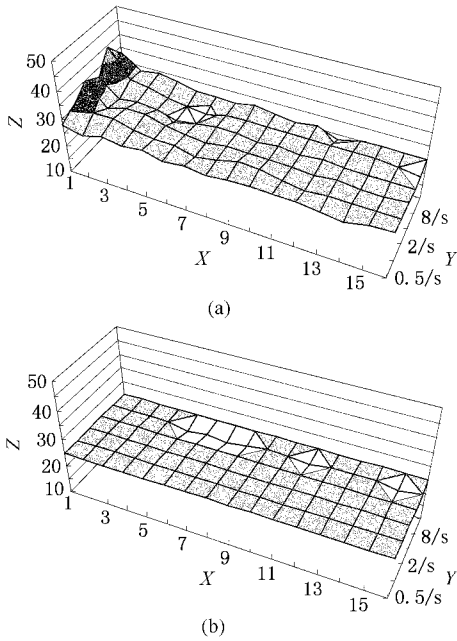


X :Source node ID ,Y :Initial packet generation rate , Z :Number of packets received by Basestation , in scene No.1
Fig. 10 Packet success ratio using TinyOS algorithm.

图 10 TinyOS 拥塞抑制算法下的报文到达率

图 11 是 EasiCC 在场景 1 的实验结果 ,图 11(a)对应实验收敛期 ,图 11(b)对应实验稳定期. 从图中看出 ,收敛期各个节点成功到达基站的报文数目差别都明显比图 10 小 ,进入平稳期后各节点都稳定在

同一发包速度(本实验配置下为 1 包/4s),各节点到达基站的报文数目几乎相等.



X :Source node ID ,Y :Initial packet generation rate , Z :Number of packets received by Basestation

Fig. 11 Packet success ratio using EasiCC algorithms.(a) Results of converging period in scene No.1 and (b)Results of stable period in scene No.1.

图 11 EasiCC 拥塞抑制方法下的报文到达率.(a)1 号场景下收敛期的模拟结果 (b)1 号场景下稳定期的模拟结果

从图 10、图 11 可以得出结论 :当各节点产生报文的速率较低(如 1 个/2 秒)时 ,TinyOS 和 EasiCC 的拥塞控制技术都能达到较好的节点间公平性. 当各节点产生报文的初始速度较高(如 16 个/秒),远远超过了网络容量时 ,TinyOS 算法不能保证节点间的公平性 ,而 EasiCC 通过拥塞控制报文在节点间协调发包速度 ,能够保证节点间的公平性.

4.3 实验测试结果 2 :数据流带宽权值

图 12 给出了场景 2 中 EasiCC 拥塞抑制方法在支持数据流带宽权值方面的实验结果. 设置节点 11 ,12 所产生数据流的带宽权值为 16 ;其他节点所产生数据流的带宽权值为 1. 从图 12 看出 ,不管初始发包速度设为多少 ,进入稳定态以后各节点都能达到合理的稳态速度 ,并且节点 11 和节点 12 到达基站的数据报文数目接近其他节点的 16 倍. 从图 12 看出 ,EasiCC 能很好地区别对待带宽权值不同的数据流 ,既给带宽权值高的数据流分配了较高网络带宽 ,同时又能保证带宽权值低的数据流也有一定比例的数据报文到达基站.

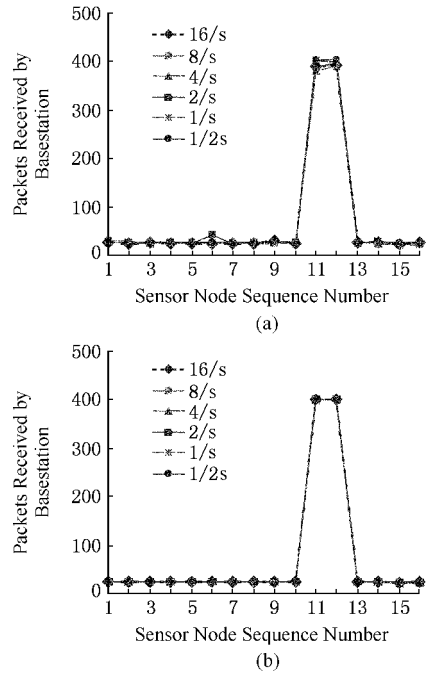


Fig. 12 Important streams get higher bandwidth than others in EasiCC. (a) Results of converging period in scene No.2 and (b) Results of stable period in scene No.2.

图 12 EasiCC 拥塞抑制方法对数据流带宽权值的支持.(a)2 号场景下收敛期的模拟结果 (b)2 号场景下稳定期的模拟结果

4 结 论

本文提出了一种实用的 WSN 拥塞控制算法 ,其实用性体现在以下 3 个方面 :1)网络节点按照报文优先级和信道拥塞状况动态调节流量 ,不需要为每个数据流维护单独的流状态信息 ,因此对节点内存和计算资源消耗很小 ,已经在实际 WSN 平台上实现并获得验证 ;2)通过事件触发的控制报文来同步节点间的控制状态 ,由于没有使用频繁的周期性控制报文 ,所以控制报文开销不大 ;3)EasiCC 设计中不追求各数据流总速度的最大化 ,而是追求各项网络性能指标的总体平衡. 从模拟和实验结果看出 ,EasiCC 算法在报文传输成功率、控制报文开销、数据流间带宽公平性、报文传输延迟等性能指标上均有良好的表现.

传感器网络的特色之一就是系统性能要求随着具体应用的不同而差别很大. 我们今后将研究在应用系统中动态统计网络特征(如网络流量大小和抖动度、网络拓扑和稳定度)的方法 ,从而更有效地调整控制参数来适应应用场景要求.

参 考 文 献

- [1] Chieh-Yih Wan, Shane B Eisenman, Andrew T Campbell. CODA: Congestion detection and avoidance in sensor networks [C]. ACM SENSYS '03, Los Angeles, USA, 2003
- [2] Yogesh Sankarasubramaniam, O B Akan, Ian F Akyidiz. ESRT: Event-to-sink reliable transport in wireless sensor networks [C]. ACM MOBIHOC '03, Annapolis, 2003
- [3] Alec Woo, David E Culler. A transmission control scheme for media access in sensor networks [C]. The ACM MOBICOM, Rome, Italy, 2001
- [4] Tinyos, an open-source operation system designed for wireless embedded sensor networks [OL]. <http://www.tinyos.net/>, 2004
- [5] Cheng Tien Ee, Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks [C]. ACM SENSYS '04, Baltimore, Maryland, 2004
- [6] Cheng Jing, Shen Yongjian, Zhang Dafang, *et al.* TCP-Shape: Study on a renovation method of network congestion control algorithms [J]. Acta Electronica Sinica, 2006, 34(9): 1621-1625 (in Chinese)
(程京, 沈永坚, 张大方, 等. TCP-Shape: 一种改进的网络拥塞控制算法研究 [J]. 电子学报, 2006, 34(9): 1621-1625)
- [7] K Tan, F Jiang, Q Zhang, X Shen. Congestion control in multi-hop wireless networks [J]. IEEE Trans on Vehicular Technology, 2007, 56(2): 863-873
- [8] He Yanxiang, Xiong Naixue, Yang Yan. An enhanced TCP congestion control algorithm [J]. Journal of Computer Research and Development, 2005, 42(12): 2070-2076 (in Chinese)
(何炎祥, 熊乃学, 杨燕. 一种改进的 TCP 拥塞控制算法 [J]. 计算机研究与发展, 2005, 42(12): 2070-2076)
- [9] F P Kelly, A Maulloo, D K H Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability [J]. Journal of the Operational Research Society, 1998, 49(3): 237-252
- [10] Guohua Zhang, Yiyu Wu, Yonghe Liu. Stability and sensitivity for congestion control in wireless networks with time varying link capacities [C]. The 13th IEEE Int'l Conf on Network Protocols (ICNP '05), Boston, Massachusetts, 2005

- [11] Bozidar Radunovic, Jean-Yves Le Boudec. Rate performance objectives of multihop wireless networks [J]. IEEE Trans on Mobile Computing, 2004, 3(4): 334-349
- [12] Xiaojun Lin, Ness B Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks [J]. IEEE INFOCOM 2005, Miami, Florida, 2005
- [13] EasiNet, WSN Products [OL]. <http://www.easinet.cn/products.htm>, 2005



Ju Hailing, born in 1972. Ph. D. candidate in wireless sensor network. Engineer. Her main research interests include wireless network and network traffic management.

鞠海玲, 1972 年生, 博士研究生, 工程师, 主要研究方向为无线网络及网络流量管理。



Cui Li, born in 1962. Ph. D., professor, and Ph. D. supervisor. Director of the Sensor Network Laboratory in ICT, CAS. Senior member of China Computer Federation. Her main research interests

include sensor technology and wireless sensor network.

崔莉, 1962 年生, 博士, 研究员, 博士生导师, 中科院计算所传感器网络实验室主任, 中国计算机学会高级会员, 主要研究方向为传感器技术、无线传感器网络。



Huang Changcheng, born in 1963. Ph. D. and associate Professor. Director of the Optical Network Lab, Department of Systems and Computer Engineering, Carleton University, Canada. Visiting

professor of ICT, CAS. His main research interests include optical and computer networking.

黄长城, 1963 年生, 博士, 加拿大卡尔顿大学系统及计算机工程系副教授, 光网络研究室主任, 中科院计算所客座研究员, 主要研究方向为光通信和计算机网络通信技术。

Research Background

Congestion control is one of the fundamental and practical research issues. It's challenging to provide congestion control in the wireless sensor networks which are composed of resource-limited sensor nodes. The algorithm proposed must be effective and enable an easy implementation at the same time. The congestion control mechanism EasiCC proposed in this paper shows good performance and potential usability. We plan to carry out more experiments to test the algorithm. This work is partially supported by the NSFC Projects and the National Grand Fundamental Research 973 Program of China under grant No. 2006CB303000.