Cross-layer Enhancement to Multi-hop Routing in Sensor Networks: an Empirical Study[1)]

SHANG Zhi-Jun[1,2] WANG Jun[1,2] YU Hai-Bin[1]

[1](*Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang* 110016)

[2](*Graduate School of the Chinese Academy of Sciences, Beijing* 100039)

(E-mail: shangzj@sia.cn)

Abstract Lossy link is one of the unique characteristics in random-deployed sensor networks. We envision that robustness and reliability of routing cannot be ensured purely in network layer. Our idea is to enhance the performance of routing protocol by cross-layer interaction. We modified mint protocol, a routing protocol in TinyOS and proposed an enhanced version of mint called PA-mint. A transmission power control interface is added to network layer in PA-mint. When routing performance of the current network is not satisfied, PA-mint monotonically increases the transmission power via the interface we added. PA-mint is able to connect orphan nodes and robust to node mobility or key nodes failure. In the case that automatic request retransmission is employed, the number of retransmissions can be reduced by PA-mint. Results from experiments show that PA-mint increases the reliability and robustness of routing protocol by cross-layer interaction.

Key words Sensor networks, routing, cross-layer design

1 Introduction

Routing protocol is a fundamental building block for sensor networks. Both queries or management information dissemination and efficient data collection need its support. Routing protocols for sensor network should be energy efficient, robust and reliable because sensor network is usually deployed to an unknown or even hostile environment[1,2]. Short-hop routing is more advantageous than long-hop routing in theory. But, the performance of short-hop routing is worth while to be studied in depth[3].

In recent years, cross layer design was studied to increase the efficiency of protocol stack[4]. Two cross layer schemes ECPS and E2LA, which are a kind of interaction between network layer and MAC layer are proposed in [5]. In [6], the author proposed a self healing mechanism based on the cross layer interaction. Mica[7] and several its successors are widely used in sensor networks research community. TinyOS is an operation system used in this serial hardware platform[8]. Programming in TinyOS uses a dialect of the C programming language called NesC, which comes with the TinyOS release. The language directly supports the component based programming model of TinyOS. By holistically analyzing both the application and the system as one component graph, cross-layer optimization for efficiency and code sizes can be done more easily.

Mint is a routing protocol in TinyOS. We found that there is some space to improve the reliability of mint. Based on the idea of cross-layer interaction, we propose an enhanced version of mint called PA-mint. PA-mint does not try to change the framework of mint. However, a power control interface is added to mint. According to link quality estimation by network layer, PA-mint can adjust transmission power of the transceiver adaptively. When routing performance of the current network is not satisfied, PA-mint monotonically increases the transmission power via the interface we added. Therefore, PA-mint is able to connect orphan nodes and robust to node mobility or key nodes failure. If the retransmission is allowed, PA-mint can reduce retransmission packets as well.

PA-mint is based on experimental observation that radio model is very irregular instead of a circle-disk. Although TinyOS simulator TOSSIM[9] is also built on the radio model obtained from experiments, power control is not supported well by TOSSIM. Therefore, we validate the performance of PA-mint empirically.

The remainder of this paper is organized as follows: We firstly introduce the original mint in Section 2. Then we describe PA-mint in Section 3. In Section 4, we illustrate the experiments and analyze the results. Finally, we conclude and present future work in Section 5.

1) Supported by National Natural Science Foundation of P. R. China (60374072, 60434030)
Received October 29, 2005; in revised form March 31, 2006

2 Original mint protocol

Mint is a tree topology based route protocol. It is suitable for data gathering application which employs many-to-one communication pattern. The goal of mint is to construct a stable routing topology upon unreliable lossy links in dense sensor network. There are three main steps in mint.

Setp 1. Discover and characterize the radio connectivity. In addition to application packets, every node locally broadcasts route update messages periodically. Both packets include a sequence field in routing packet head. Packet loss events are inferred based on gaps in link sequence number. The in-bound link estimation (E_{rx}), *i.e.* the quality a node receiving packet from its neighbor nodes, is given by

$$E_{rx}^{(i)} = w_1 \cdot E_{rx}^{(i-1)} + w_2 \cdot \frac{N_{succ}}{N_{succ} + N_{miss}}, \quad i = 2, 3, 4, \cdots, n \tag{1}$$

where w_1 and w_2 denote different weights, $w_2 > w_1$. N_{succ} denotes the number of packets received successfully and N_{miss} the number of packets lost. In fact, moving window exponentially weighted moving average (MWEWMA) method is used to smooth E_{rx} in (1).

Step 2. Neighbor management. Every node builds a neighbor table based on the packet it heard. Neighbor table is updated dynamically and the size is limited. Only the "good" node is kept in the table.

Step 3. Select good routes. Route selection is based on a abstract value of routing cost. Mint also considers the out-bound link estimation (E_{tx}), *i.e.* the receiving quality of neighbor nodes. E_{tx} is extracted from routing update packet, which piggybacks some out-bound link estimation. In fact, E_{rx} and E_{tx} indicate in and out link qualities respectively. They are combined to a link cost, which is given by

$$C_{link} = \frac{1}{E_{rx} \cdot E_{tx}} \tag{2}$$

Total routing cost is give by

$$C_{total} = f(C_{link}, C_{parent}) \tag{3}$$

Link cost (C_{link}) indicates the quality of local link, while parent cost (C_{parent}) indicates the parent node′s reachability to sink node or base station. Mint uses a simple $f(\cdot)$, which sums up C_{link} and C_{parent}.

3 PA-mint protocol

Link quality is estimated in mint passively. Instead, PA-mint attempts to change link quality by adjusting transmission power, which is controllable by node itself.

3.1 Transmission power adjustment policy

For networks such as wired networks and infrastructure supported wireless networks, reliable link (compared with that of sensor network) is assumed by routing protocol. Routing protocols focus on how to choose the routing path efficiently. For sensor networks, however, wireless communication is noisy and time varying because of low-cost and low-power CMOS based RF transceiver. Results from real deployment networks show radio connectivity is probabilistic rather than a boolean event[10]. That is, connectivity is not a clear cut concept between connected and not connected. This empirical observation breaks the typical assumption of the circular-disc connectivity model.

Link quality depends on uncontrollable factors such as node mobility, weather, interference, noise, as well as controllable parameters, such as transmission power and antenna. If uniform transmission power is specified by applications, two results maybe occur. If transmission power is set too low, there is a danger of very bad link quality, even network partitioning. If transmission power is set too large, the pro is higher signal to noise ratio (SNR) or lower bit error rate (BER), the con is the limited spatial reuse.

Variable transmission power scheme can increase the capacity of the multihop network[11]. Furthermore, transmission power control results in extending battery life, which is a crucial factor for sensor networks. Considering mint protocol, when E_{rx} and E_{tx} is small, C_{link} will be very high. Thus, there will be no node that can be chosen as a parent. Therefore, PA-mint adjusts transmission power based on the link estimation in network layer, *i.e.* in the estimator component of mint. The goal is to make the local link quality better.

We add the interface of CC1000Control to mint, as shown in Fig. 1. A transmission power adjustment algorithm called PowerControl is implemented as shown in Fig. 2. At initialization phase, every node uses the same power level. In routing estimation update period, E_{tx} in neighbor table is scanned. If there is no E_{tx} higher than the threshold, current power level is incremented by one; if the neighbor table is null, current power level is also incremented. To keep the stability of routing topology, the algorithm sets a high limit of power level. The increased transmission power may change the routing topology, but this change is very limited. If the threshold of E_{tx} is set appropriately, a domino like topology change will not be triggered across the entire network.

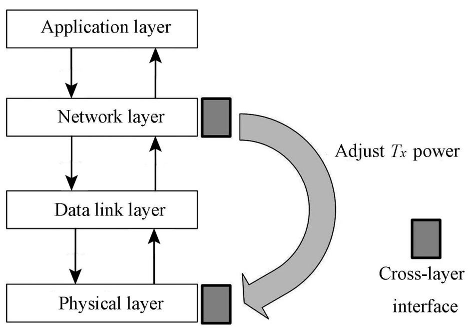

Fig. 1 The idea of cross-layer interaction in PA-mint

```
Input: neighbor table T, the threshold (for E_tx) E, high
 limit of Tx power level P, current Tx power p_tx
Output: None
POWERCONTOL (T, E, P)
bAdjust = TRUE
if T != Null
    foreach entry e in T
        if E_tx of e > E
            bAdjust = FALSE
            break
if bAdjust == TRUE
    p_tx = p_tx +1
    if p_tx <= P
        CC1000Control.SetRFPower( p_tx )
```

Fig. 2 Power control algorithm

3.2 Modification to routing cost estimator

If local link quality is increased, C_{total} of many neighbor nodes are reduced. To keep the fairness, we modify routing cost estimator in mint as following:

$$C_{new} = f(C_{link}, C_{parent}) + \alpha C(n_{child}) \tag{4}$$

where n_{child} is number of child nodes, α is a tunable parameter. The node with fewer children is chosen as a parent in case two nodes own same C_{total} in (3).

3.3 Power adjustment vs. retransmission

If we assume that transmitting one packet one hop successfully is an independent process with the probability p_i, the probability of deliver one packet end to end is

$$P = \prod_{i=0}^{\max\_hop} p_i \tag{5}$$

Therefore, the reception rate decreases with the increase of hop counts. In real world, adjacent environment and the hardware of different nodes vary, not all p_i are the same. There is the case $p \ll 1$, *i.e.*, any drop in any part of multi hop chain will cause the packet lost.

Limited retransmission is a simple solution to the above problem. But the retransmission also has some drawback obviously. Take Mica2 mote as an example. CC1000 is a half duplex transceiver. To receive ACK packet, CC1000 has to be switched to receive state after a packet is transmitted. This behavior will cost nearly 250 microseconds. In addition, ACK packet is transmitted and received along with each upper layer packet. In total, the retransmission adds about 6 milliseconds latency along each hop. It makes contribution to packet collision as well.

We can regard PA-mint as an alternative solution. If the retransmission is not supported, PA-mint can deliver more packets to base station. Otherwise, PA-mint can reduce the number of retransmissions as well.

4 Experiments

4.1 Experiment setup

We run Surge application, which is provided as an example in TinyOS. The network collects environmental parameters periodically and delivers data to a base station. The layout is shown in

Fig. 3. The grid distance in Fig. 3 is 50cm. The grey rectangle is the position of base station. The other grey circles are positions of other nodes. The white circles are the positions where node 2 was moved two times. In our experiment, power level of PA-mint is set to 2∼9 (-20dBm∼-10dBm). mint applies a fixed power level 3 (-19dBm). Each node sends five hundreds application layer packets to the selected parent node. These packets are delivered by multi-hops communication and recorded by a PC *via* serial port.

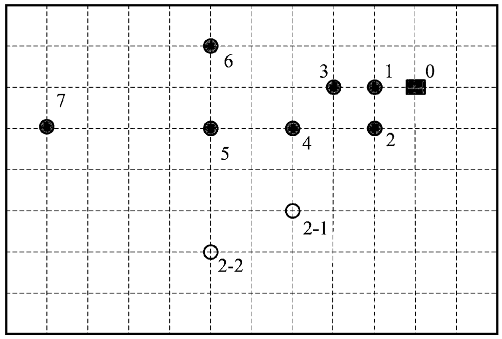

Fig. 3 The layout of experimental network

4.2 Experiment results and analysis

1) PA-mint *vs* mint without ACK

We observed the hop count increased with the distance between nodes and base station, which is denoted as d in the following. Running PA-mint, the maximum hop count is four at the node which labeled seven. Running mint, the maximum hop count is three, because node seven can not connected to the network due to the low power level. For PA-mint, the final power level of different nodes is shown in Fig. 4. Reception rates running two protocols respectively are shown in Fig. 5. Reception rate decreases when hop count increase. But, the reception rate in PA-mint is higher than that of mint due the power level adjustment policy. Especially to node seven, it can not deliver any packet to base station running mint.

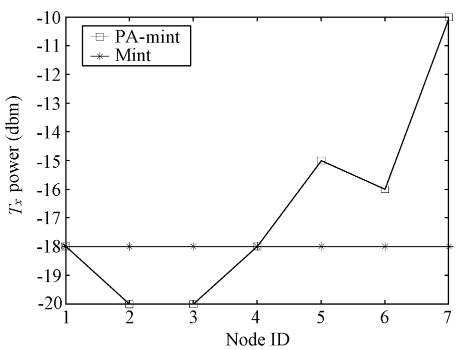

Fig. 4 Transmission power of PA-mint and mint

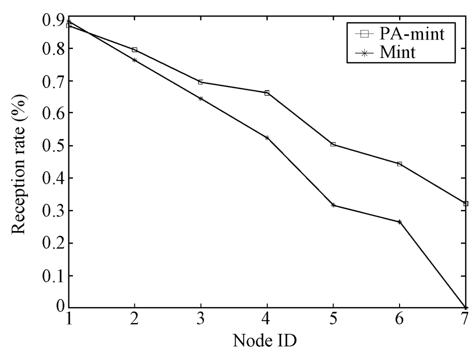

Fig. 5 Reception rate of PA-mint and mint (w/o ACK)

2) PA-mint *vs* mint with ACK

Results running both protocols with ACK are shown in Fig. 6. It is obvious that both PA-mint and mint have a higher reception rate. But the percentage of retransmissions in PA-mint is less than that of mint.

3) Power adjustment in PA-mint

We moved node 2 two times in the experiment. The positions of node 2 are shown in Fig. 3. The mean number of packets per minute delivered to base station is shown in Fig. 7. We also draw the power level in the same figure. But the value is zoomed large five times just for clear showing. In Fig. 7, reception rate for node 2 decreases dramatically when node 2 is moved to position 2-1 (at the point of two minutes). Appling PowerControl algorithm, power level of node 2 is increased dynamically. When node 2 is moved to the position of 2-2 about seven minutes later, the power level reaches to the highest limit.

Fig. 8 shows the power level change in node 7. At the beginning of the experiment, node 7 can not connect to the network due the distance from sink node. The neighbor table in node 7 is empty. Node 7 then increments its power level in the following routing update periods until it reaches the highest limit, and chooses node 5 as its parent.

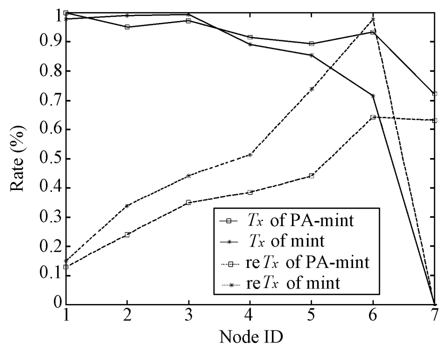

Fig. 6 Reception rate of PA-mint and mint with ACK

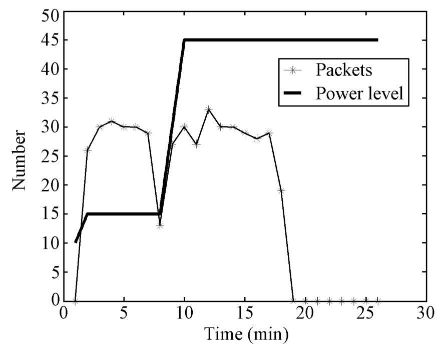

Fig. 7 Power adjustment of node 2

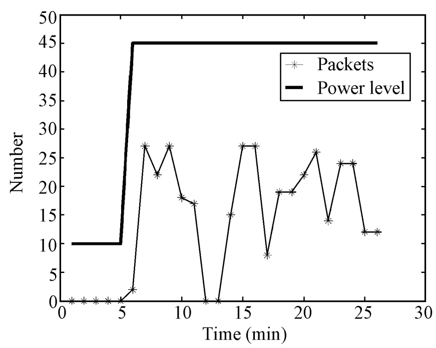

Fig. 8 Power adjustment of node 7

5 Conclusions and future work

We modified mint protocol in TinyOS and proposed a cross-layer enhanced version, PA-mint. PA-mint increments the transmission power based on neighbors′ estimation on link quality. We empirically evaluated the performance of PA-mint. The results are two fold. First, PA-mint can increase the robustness and reliability of original mint. Second, if retransmission is allowed, PA-mint can reduce retransmission packets.

However, this work is still in the early stage. We therefore discuss some related questions and present future work.

1) Energy conservation

In the sensor network, the nodes use battery in most cases, so it is very important to save power in order for the battery to last as long as possible. However, PA-mint only increments the transmission power if the wireless link is not reliable enough, until the upper power limit is arrived. It seems that PA-mint should decrement transmission power under some conditions (*e.g.* if the wireless link is reliable enough). But the question is when the local wireless link will be reliable enough? Since sensor network is almost static, generally, wireless links become worse during the lifetime of sensor networks. In fact, the goal of PA-mint is to find the worse, but the best available link and make the quality better. However, how to reverse from the direction of increasing transmission power is still an interesting question, *i.e.*, in case that environmental noise or external interference become weaken.

2) Interference with other nodes

The interference with other nodes is not considered enough when the transmission power of a certain node is incremented in section 3. Intuitively, the interference in a certain local area will be strong if the transmission power is increased. But, we do not think it is a serious problem for low traffic applications such as Surge. However, an in-depth theoretical analysis is better for more general scenarios.

3) Scale of the evaluation network

In an ordinary sensor network, there will be a lot of nodes. So the performance of PA-mint in a large-scale sensor network should be tested either through experiment or simulation. We choose the former method as we explained in section 1. However, there are only seven nodes in the experimental network. It is difficult to test PA-mint in a large real world network. We have planned to evaluate the performance of PA-mint in TOSSIM. But, TOSSIM is not convenient to simulate the power control. We will try other simulation tools, such as OMNet++ or Prowler in the future.

4) Jointly design

Transmission power is just one of the factors that affect link quality. If we integrate the retransmission policy, power adjustment and multi-path routing, we envision that the reliability and robustness of PA-mint can still be increased.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments.

References

1 Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2005, **3**(3): 325∼349
2 Al-Karaki J N, Kamal A E. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 2004, **11**(6): 6∼28
3 Haenggi M, Puccinelli D. Routing in *ad hoc* networks: A case for long hops. *IEEE Communications Magazine*, 2005, **43**(Oct.): 93∼101
4 Raisinghani V T, Iyer S. Cross-layer design optimizations in wireless protocol stacks. *Computer Communications*, 2004, **27**(8): 720∼724
5 Safwat A, Hassanein H, Mouftah H. Optimal cross-layer designs for energy-efficient wireless ad hoc and sensor networks. In: Proceedings of 2003 IEEE International Performance, Computing, and Communications. Piscataway, NJ, USA: IEEE Press, 2003. 123∼128
6 Sadler C, Kant L, Chen W. Cross-layer self-healing mechanisms in wireless networks. In: Proceedings of the World Wireless Congress. Cupertino, CA, USA: Delson Group Inc., 2005. 249∼254
7 Hill J, Culler D. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 2002, **22**(6): 12∼24
8 Gay D, Levis P, Culler D. Software design patterns for TinyOS. In: Proceedings of the 2005 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems. Chicago, Illinois, USA: ACM Press, 2005. 40∼49
9 Levis P, Lee N, Welsh M, Luller D. TOSSIM: Accurate and scalable simulation of entire tinyOS applications. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. Los Angeles, California, USA: ACM Press, 2003. 126∼137
10 Woo A, Tong T, Culler D. Taming the underlying challenges of reliable multihop routing in sensor networks. In: Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys′03). Los Angles, California, USA: ACM Press, 2003. 14∼27
11 Gomez J, Campbell A T. A case for variable-range transmission power control in wireless multihop networks, In: Proceedings of the Conference on Computer Communications (Infocom). Piscataway, NJ, USA: IEEE Press, 2004. Series 1-4, 1425∼1436

SHANG Zhi-Jun Research assistant at Shenyang Institute of Automation, Chinese Academy of Sciences (CAS). He is also a Ph. D. candidate in Shenyang Institute of Automation, CAS and Graduate School of the Chinese Academy of Sciences. His research interests include embedded system, network modeling, and protocol design.

WANG Jun Ph. D. candidate at Shenyang Institute of Automation, CAS and Graduate School of the Chinese Academy of Sciences. His research interests include routing and information processing in wireless sensor network.

YU Hai-Bin Professor at Shenyang Institute of Automation, CAS. He received Ph. D. degree in control theory and engineering from Northeast University in 1997. His research interests include real time scheduling theory in industrial communication, wireless sensor network, and distribute artificial intelligence.