Poster Abstract: Route-back delivery protocol for Collection Tree Protocol-based applications

Victor Rosello[1], David Boyle[2,3], Jorge Portilla[1], Brendan O'Flynn[3], Teresa Riesgo[1]

[1]Centro de Electronica Industrial, Universidad Politécnica de Madrid
victor.rosello@upm.es
[2]Dept. of Electrical and Electronic Engineering, Imperial College London
david.boyle@imperial.ac.uk
[3]Tyndall National Institute, University College Cork

Abstract. We propose the Route-back Delivery (RBD) protocol; a routing mechanism to create reverse routes exploiting the Collection Tree Protocol to allow unicast data dissemination from the sink. The main goal of this work is to provide a mechanism to enable bi-directional communications among the root(s) and specific sensor nodes in data gathering applications that does not use broadcast only mechanisms. The main objective of the root-to-remote-nodes route creation is to disseminate short messages to change application parameters in a unicast fashion. This facilitates remote configurability in heterogeneous WSN deployments.

1 Introduction

Application parameter updating (e.g. sampling rate, alarm thresholds, reporting periodicity, etc.) has high dependability on the hardware associated with the nodes (i.e. in heterogeneous deployment scenarios), and the spatial resolution of the devices deployed. For example, nodes with different sensors have different application level operating parameters. Nodes with different deployment field characteristics (e.g. indoor versus outdoor) can have different detection thresholds for the same type of sensors and event. For this reason we propose a mechanism that can select the destination node for the application update against the traditional reprogramming techniques for wireless sensor networks, which usually do not provide node selection capabilities [1]. Typical data collection applications tend to use unidirectional communication (e.g. from remote nodes to a root/gateway). Many protocols for this kind of applications are proposed in the literature, the most popular of which is Collection Tree Protocol (CTP) [2] that is integrated in TinyOS.

2 The Collection Tree Protocol

The CTP protocol provides reliable, multi-hop delivery of packets to the root of a tree for relatively low traffic rates. CTP does not send a packet to a particular root;

adfa, p. 1, 2011.

instead, it implicitly chooses a root by choosing a next hop. Nodes generate routes to roots using a routing gradient. CTP uses expected transmissions (ETX) as it routing gradient. The ETX of a node is the ETX of its parent plus the ETX of its link to its parent. Given a choice of valid route, CTP chooses the one with the lowest ETX value. The implementation of CTP consists of three different components: Link Estimator, Routing Engine and Forwarding Engine, and uses two different types of messages: CTP Data (shown in Fig. 1) and CTP Routing. CTP Routing messages are transmitted as periodic beacons, and are used to create the route to the root with the lowest ETX dynamically. CTP Data messages are used to transmit the information to the root. In this work, we use the information provided by the CTP Forwarding engine when CTP Data messages are received to create reverse routes from the root to the origin node of a data message.

3 Route-back delivery (RBD)

Most pro-active routing protocols are designed to create routes between any nodes in the network, and typically require the use of broadcast transmissions (e.g. AODV, DYMO, or its TinyOS version TYMO) [3]. This kind of transmission has a significant impact on channel availability, and is ultimately energy inefficient. The primary motivation behind our work is to allow remote, node-specific (i.e. unicast), and configurability on-the-fly for heterogeneous deployments of WSNs. These operations tend to be sporadic, and routes must connect the root to the specific node (not any/all nodes in the network). Therefore, it is desirable to reduce the effort required to create routes, in addition to reducing the impact of the routing mechanism on the channel availability node resource usage.

3.1 Reverse path creation

To simplify the route creation mechanism, regular data messages are exploited to build reverse routes. This relies on CTP in the creation of reliable routes. In this way, we seek to develop a method where no extra messages (beyond the regular data messages) are needed to create the reverse path, and thus reduce the transmission cost overhead. When a node receives a CTP data message, the origin address, sender address and the time has lived (THL), are used to create a record of the route in the route-back table. This table contains the information needed to reach the origin node of the CTP data message.

3.2 Route-back Table

This table contains the available routes. The significant fields of the route-back table are: *Destination:* Destination address of the route. *Neighbor:* Address of the next hop in the path. *Hop Count:* Remaining hops to reach the destination. *Routed messages:* Number of times that the route has been used.

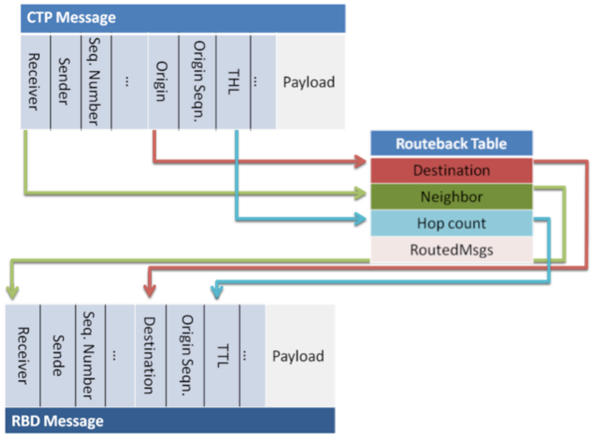

Fig. 1. CTP Data Message, Route-back Table and the RDB Message

3.3 Route-back Delivery

Currently, only the root node can begin a route back delivery process, and only the *Destination* nodes in the route-back table can be set as destination addresses in an RBD message. On receipt of an RBD message, the node searches for the destination address in the route-back table. If the route exists, the message is sent to next hop in the path using the *Neighbor* address in the route-back table, and the time to live (TTL) field in the RBD message is decremented. If TTL is 0 or a node in the route is not available an error report is sent to the root node. These error reports are sent using CTP as a regular data message. In Fig. 1, the relationship between the CTP Data message, the route-back table, and the RDB message is illustrated.

4 Implementation and Evaluation

The RBD mechanism has been implemented to work together with the CTP version included in the TinyOS-2.1.2 distribution. In Fig. 2, we depict an application that combines CTP and RBD. Initial tests were conducted on a test-bed of 4 TelosB devices, (1 root and 3 nodes), under various topologies: 3 branches of 1 hop distance, 2 branches (one of 2 hops and one of 1 hop) and 1 branch with 3 hops were constructed and tested. Initial tests at scale are simulated using Cooja.

4.1 Simulation scenario

CTP data are sent at a rate of once per second. The RBD send rate is 1-3 seconds. 1 root and 20 randomly placed nodes were used. The simulation runs for 180 seconds. A 90 per cent success rate was observed for each of transmitting and receiving packets.

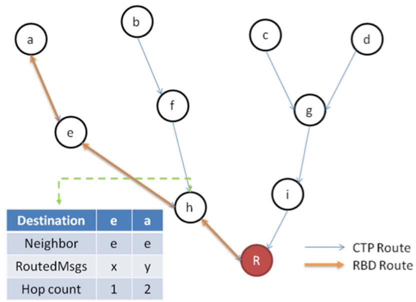

Fig. 2. CTP+RDB Network topology

4.2 Results

100% of nodes receive RBD data. RBD error happens in 12 of 20 nodes. Recovery times and delivery errors are shown in Table 1 and Table 2, respectively. The memory footprint for an application with CTP and RBD with a 10-element route-back table is shown in Table 3.

Table 1: Time to RDB delivery after an error

Max	Min	Mean
111s	54ms	1.1s

Table 2: Number of errors before RDB delivery

Max	Min	Mean
6	1	2

Table 3: RBD Memory Footprint

Memory	CTP	CTP + RBD	Δ
RAM (bytes)	2072	2808	736 (35%)
ROM (bytes)	16458	18206	1648 (10%)

5 Conclusion

We demonstrate the feasibility of simplified, low-overhead, selective message delivery in WSNs exploiting CTP. We plan to refine and deploy RBD in a large test-bed to fully characterize and validate the approach; then contribute it to the community.

References

[1] Q. Wang, Y. Zhu and L. Cheng, “Reprogramming wireless sensor networks: challenges and approaches”. IEEE Network, vol.20, no.3, pp. 48- 55, May-June 2006.

[2] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis and A. Woo, “The Collection Tree Protocol (CTP)”. TinyOS documentation. Available: http://www.tinyos.net/tinyos-2.x/doc/txt/tep123.txt.

[3] M. M. Chandane, S. G. Bhirud and S.V. Bonde. “Analysis of AODV, DSR and DYMO Protocols for Wireless Sensor Network”. Journal of Computing, vol. 4, no 1, January 2012.