Thoughts about a network layer for sensor networks

Gilman Tolle

February 22, 2005

1 Naming

Moving data to a point in the network is easier if you can address the point.

Consider the following classes of addresses:

1. The address of a single node
2. The address of a group of nodes
3. The address that implicitly means "all nodes"

Standard IP addresses fit the first class. IP multicast groups fit the second class. IP has no analogue for the third class.

All of these addresses are explicit. Much of the current practice in sensor networks leaves the destination address implicit. It’s "the root of the tree I’m in", or "all the nodes within transitive radio connectivity".

A node could belong to a group address, or many group addresses. What A node could join a group address because of a predicate, itself sent to the "all nodes" address. With group addresses, we can separate data movement from group formation by interposing this membership step.

With explicit naming of nodes, groups, and "all nodes", we can also move towards unifying our separate network layers of collection and dissemination.

2 Mechanisms

It’s always easy to send a message to one of these three addresses classes: flood it out to everyone. It’s just not always efficient.

- Efficiently sending a message to a single address is the goal of collection routing. (e.g. MintRoute, ReliableRoute, Drain)
- Efficiently sending a message to a single address is also the goal of geographic routing. (e.g. Beacon Vector Routing)
- Efficiently sending a message to the "all nodes" address is the goal of dissemination. (e.g. Trickle, Drip)

Efficiently sending a message to a group of nodes is the goal of IP multicast, but I know of no analogue in the sensor network space yet. Building a spanning tree, like IP multicast does, would be an interesting exercise. We see that sending a message to a group of nodes may require forwarding by nodes not in the group, but those nodes are still required to make the group reachable. They should be auto-discovered by such a protocol, and may change all the time if the protocol uses the adaptivity lessons learned in MintRoute.

This model does not preclude in-network processing of packet data, but many current applications don’t need in-network processing. Simple message replacement or denial along existing paths could be done with "Intercept" interfaces, and in-network processing that needs its own special path formation could always exist as an alternate network layer that uses the different link layers directly.
3 Link Layers

If we call this a network layer, it should support multiple link interfaces. All of our nodes have 2 interfaces now: a wireless connection and a wired connection. Nodes in the future may have more: multiple radios, for example. Selecting which links will be used to reach a given network layer address requires the equivalent of "ifconfig" for motes; a binding between network addresses and links.

Consider a sensor network node in which each link had its own address, and its own conception of which addresses lie on the other side. As in PPP, a node can discover the address opposite its point-to-point link. As in Ethernet, a node can discover the addresses opposite its radio link, or that node can leave them undiscovered and rely on the neighbors to do their own filtering. Right now, we give the wireless link first-class status, calling its link-layer address the network-layer address of the node. The wired link doesn’t get its own name or its own link-layer header, and is understood to have a single specific network-layer address on the other side.

Now, if our dissemination layer and our collection layer could understand multiple links in a common addressing mode, we can think about a dissemination that crosses interfaces to reach node patches not physically connected, or a collection tree that involves Internet links in the middle of the tree and not just at the root.

4 SP?

Many of these ideas use the IP mindset of treating all links as equals, and treating all addresses within one framework. Further efficiency gains may result from taking advantage of specific link technologies. For example, we could trade memory for energy by maintaining a mapping between network addresses and radio link addresses, so the receivers’ filtering hardware can reject packets without waking up the microcontroller, or manage the preamble length automatically.