Fuzzy Inferno and Nostalgic Cow: Two Practical Applications for Ad-Hoc Networks

Wlodek Olesinski
Olsonet Communications Corporation
51 Wycliffe Street
Ottawa, Canada K2G 5L9
e-mail: wlodek@olsonet.com

Pawel Gburzynski & Ioanis Nikolaidis
University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
e-mail: [pawel,yannis]@cs.ualberta.ca

We present two example applications of wireless ad-hoc networks: *Fuzzy Inferno* and *Nostalgic Cow*. We then describe two general families of applications these two examples belong to. Finally, we mention our routing protocol, operating system and hardware that we used to implement and demonstrate these applications.

In *Fuzzy Inferno*, the ad-hoc network helps firefighters combat a forest fire. Imagine airplanes (water bombers) dropping hundreds of cheap nodes (called *tags*) on the burning areas of a forest. The tags, equipped with sensors, automatically form an ad-hoc network, collect temperature readings and other data, and relay them to Masters. The Masters in this application can be laptops or PDAs in fire trucks. They derive tag locations from incoming multi-hopping packets. Firemen, equipped with this live data, can fight the fire more efficiently.

This application is one practical instantiation of a blueprint that we call *RTags* (*Routing Tags*). This application is characterized by the presence of an "elevated" node type called Master. Any node can become Master at any time, either self proclaimed or elected by other nodes. Usually, the network is partitioned among the Masters that interface it for external (human or computer) operators. In a typical operation, Masters send messages to other nodes (tags) to solicit replies, or to trigger some actions. This does not preclude other nodes (any nodes) exchanging messages: the traffic originating or terminating at Masters is merely "highlighted," which means that some routing parameters are optimized for its presence.

In the *Fuzzy Inferno* instantiation, ordinary nodes (tags) are dropped over the burning forest, while the Master node, operated by a fireman, displays the data collected by the tags.

Another application, *Nostalgic Cow*, is deployed on a large cattle farm. Imagine cows in a herd being tagged with small wireless devices (*tags*). The tags periodically broadcast unique identifiers and other data, which are relayed through the ad-hoc network with the cooperation of other tags. These data are collected via stationary nodes called *pegs*, and relayed further to a human operator or a computer system for processing.

This application is an instantiation of another generic application called *T&P* (*Tags & Pegs*). In *T&P*, the network consists of two types of nodes. The pegs are intentionally immobile, at least compared to tags. Some pegs can play the role of external gateways. Their primary purpose is to provide a kind of semi-fixed infrastructure for tracking the location of tags. Depending on the requirements, the assortment of tools facilitating this tracking may include specialized sensors deployed at tags (e.g., accelerometers) reporting their status to pegs. We do not consider GPS as the primary source of location information, because of the cost and environmental limitations. A significant degree of accuracy can be achieved by measuring and correlating the received signal level at multiple pegs perceiving the same tag. The class of applications covered by *T&P* deals with mobile objects (assets, luggage, people, etc.), whose mobility patterns may have to be classified by event-triggering predicates (mutual exclusion, avoidance of certain spots, time restrictions, and so on).

We implemented the above applications on two low-cost hardware platforms: (1) MSP430F148 microprocessor (from Texas Instruments) driving TR1000 transceiver (from RF Monolithics), and (2) eCOG-1 microprocessor (from Cyan Technology) driving CC1000 transceiver (from Chipcon). Each application can execute on either platform.

The devices run *PicOS* [2] – our highly compact, small footprint, multithreaded operating system, which supports lightweight processes and thread scheduling, and *Tiny Ad-hoc Routing Protocol* (*TARP*) [1, 3] – our unorthodox ad-hoc forwarding scheme. The latter is driven by rules, which provide for controllable and automatically tunable fuzziness (redundancy of paths) with smooth route recovery.

Summarizing, we propose two interesting applications, outline their general features that could be used in many other applications, and mention the software and hardware platforms used to implement them. With all this in hand, we can rapidly build a variety of other useful applications [4].

References

[1] P. Gburzynski, B. Kaminska and W. Olesinski. "A Tiny and Efficient Wireless Ad-hoc Protocol for Low-cost Sensor Networks," DATE'07, Nice, France, April 2007, *to appear*.

[2] E. Akhmetshina, P. Gburzynski, and F. Vizeacoumar. "PicOS: A Tiny Operating System for Extremely Small Embedded Platforms." *Proceedings of ESA'03*, Las Vegas, June 23-26, 2003, pp. 116-122.

[3] W. Olesinski, A. Rahman, P. Gburzynski. "TARP: A Tiny Ad-hoc Routing Protocol for Wireless Networks." *Proceedings of Australian Telecommunications Networks and Applications Conference ATNAC*, Melbourne, Australia, December 8-10, 2003.

[4] Olsonet Communications Corporation, www.olsonet.com