Title
Non Intrusive Analysis of Sensor Network MAC Protocols

Permalink
https://escholarship.org/uc/item/4wt0v46m

Authors
Tyler McHenry
John Heidemann

Publication Date
2005
Non-Intrusive Analysis of Sensor Network MAC Protocols

Tyler McHenry, John Heidemann

ISI Laboratory for Embedded Networked Sensor Experimentation - http://www.isi.edu/ilense

Introduction: Some experiments require MAC-layer analysis of live sensor networks

- High density networks cannot be simulated reliably
  - Potential traffic increases, the actual collision rate must be measured rather than modeled statistically to discover realistic MAC performance. In real networks, packets do not collide or corrupt at a known rate.
  - Low-level MAC layer timings need testing in light of real collision rates.

- Designed from the bottom up to be as portable as possible
  - New Reporting and analysis software are addons to Ethereal
  - Packet-level processing emulates a Linux network interface
  - Data collection based on snooper motes (packet format definitions) are easily created for new MAC protocols or higher-level protocols

Problem Description: MAC-Level analysis is hindered by the addition of reporting code

Key Idea:

- Need to completely avoid modifying all mote-side software

Proposed Solution: A snooper mote can collect data to be processed later with Ethereal

The “Radio Traffic Analysis” (RTA) Suite

- Data collection based on snooper motes
  - Constantly listens using the same physical layer as the MAC being analyzed, but need not implement the MAC protocol itself.
  - Echoes every byte it overhears to a Linux host over some backchannel (most simply, a serial cable). Other motes do not even know it exists.
  - Can also monitor for “total loss” packet collisions by detecting changes in RSSI.

- Packet-level processing emulates a Linux network interface
  - The moteradio driver accepts bytes from a snooper mote and presents them as whole packets on a network interface with a microsecond timestamp added.
  - Any program can read this interface as it would read any other network card.

- Reporting and analysis software are addons to Ethereal
  - A familiar and widely-used network traffic analysis package
  - Provides a portable framework and familiar GUI that can capture packets from any network interface and feed them to custom processing code depending on the type of packet captured.

- New dissectors (packet format definitions) are easily created for new MAC protocols or higher-level protocols
  - Packets progress through a tree of analysis code from physical layer to MAC layer to higher layers (if applicable).
  - Adding a new dissector for a new MAC or higher layer protocol is straightforward.
  - Dissectors currently exist for B-MAC, S-MAC, SCP-MAC, the SMACTest application, and Linkstate data.

- Designed from the bottom up to be as portable as possible
  - Ethereal is already portable to Linux, Windows, and more.
  - Porting to a different version of Linux or a new platform entirely necessitates modifying or replacing the moteradio driver only.

The Ethereal GUI with RTA Modifications

Goal: Automatic Adaptability in MAC Analysis

- Each protocol with a dissector may provide more complex data analysis
  - For S-MAC, RTA provides: point-to-point conversation tracking, enumeration of schedules and schedule tracking, and node activity graphs.
  - Each of these requires intensive post-processing of captured packets.

- Analysis provided by the dissector should not make assumptions
  - A key use of Radio Traffic Analysis is testing the effectiveness of modifications to protocol parameters.
  - Dissectors should detect parameters whenever possible.

- Example: S-MAC sleep/listen cycle length (period) and schedules
  - The S-MAC dissector detects the period in use within some margin of error and is able to determine that 3 nodes are on the same schedule.
  - It can then constrain the timing of sleeps in that schedule within a +/- 5ms window for prediction of future sleeps times.

Current Approach:

- Add code to motes to send debug info