SESAM:
A Semi -Synchronous, Energy Savvy, Application-Aware MAC

Joint work with Renato Lo Cigno and Matteo Nardelli, University of Trento, Italy

Published at IEEE/IFIP WONS 2009, 2-4 February 2009, Snowbird, Utah, USA

Soon: Networks and Distributed Systems group, Department of Informatics, University of Oslo

Michael Welzl http://www.welzl.at

DPS NSG Team http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

CAIA guest talk
Swinburne University
Melbourne AUS
12 February, 2008

Outline

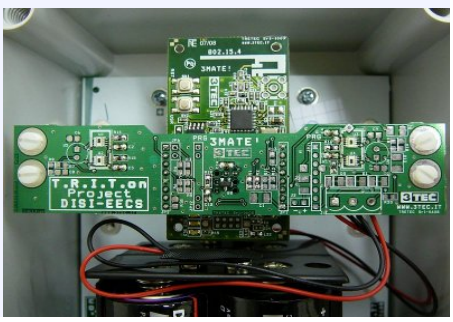
- Problem / background
  - History, background
  - Power-conserving MAC schemes for WSN
- SESAM
  - Results
- Ongoing work
- Conclusion

History

- I live in the Pathfinder motel, corner Cotham/Burke
  - Nice place, but WiFi is slooow
  - Last year, I started to wonder if that isn't a bigger problem than high-speed TCP (e.g. I told you about the Window Scaling option issue…)
- Initial idea: when it is known that some hosts will send for an extended duration, they can form a clique, to schedule transmissions (a bit like a dynamic temporary token ring) ⇒ avoid CSMA/CA collisions
  - I visited Trento to work with Renato Lo Cigno on this
  - … and found that similar stuff exists - in research, but not in the standard
  - Renato: "this makes more sense for WSNs, where traffic patterns are known"
  - I started to dig into the energy-conserving-MAC-for-WSN literature
  - Developed a rough idea with Renato, based on explicit dissemination of traffic patterns, to avoid overlaps; quite complex
  - Later, Renato visited me for two days, and told me about his new idea that makes this all much better and simpler, and we discussed it
    ⇒ **SESAM !**

Trento context: TRITon project

- Regional project on road (tunnels mainly) management
- Monitoring infrastructure through WSN
- Network availability for rescue and fire dept.
- Energy is a major concern when cabling is not available (most of the times for retrofitting)
- Tmote compliant hardware
- Tiny OS + specialized routing and application software
- Tiny OS implementation of B-MAC … until SESAM is ready to roll out!!



T.R.I.T.on
Trentino Research & Innovation for Tunnel Monitoring

Web Site: http://triton.disi.unitn.it/

What's an "energy-conserving MAC for WSN"?

- WSN: tiny devices, deployed in a way that can make plugging in a power adapter very inconvenient
  - Very, very inconvenient
  - E.g. thrown out of airplane across a cornfield, dumped in the ocean, …
- Starting point for wireless communication: IEEE 802.15.4 CSMA (/CA)
  - Not very energy efficient: e.g. nodes are *always* awake to listen
  - Hence, WSN hardware commonly implements only PHY + low level framing parts of IEEE 802.15.4 spec; MAC typically customized
  - CSMA(/CA) behavior typically used as fallback in WSN MAC schemes
- Energy-conserving MAC work usually starts with the question: "What if we sometimes put nodes to sleep on top of CSMA?"
  - Note: idle listening is usually the most energy consuming part

Energy-conserving MAC examples

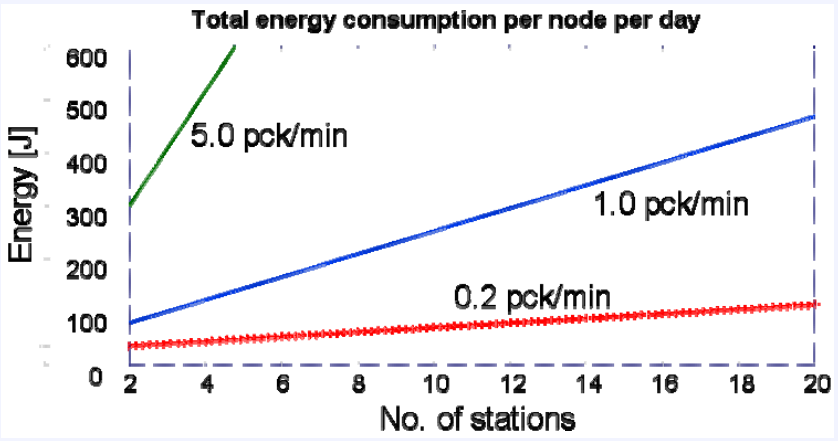
- S-MAC: Everyone sleeps most of the time. Occasionally, senders and receivers wake up together to contend for the channel
  - With several suggested improvements: e.g. immediately go back to sleep after hearing an RTS or CTS packet that is not for me
  - T-MAC: terminate time slots early when nodes don't have anything to send
- TDMA schemes / hybrids
  - TRAMA: alternate between…
    - random access phase: nodes tell each other transmission schedules
    - Scheduled phase: use the schedules to avoid contention
  - FLAMA: does not need schedule announcements, exploits knowledge of application-specific traffic patterns ("flows")
  - Z-MAC: schedules assumed to be more static – assigned in the beginning and upon major network configuration changes
  - Funneling-MAC: improves upon Z-MAC by exploiting typical tree-upstream data flow for schedule distribution

Energy-conserving MAC examples /2

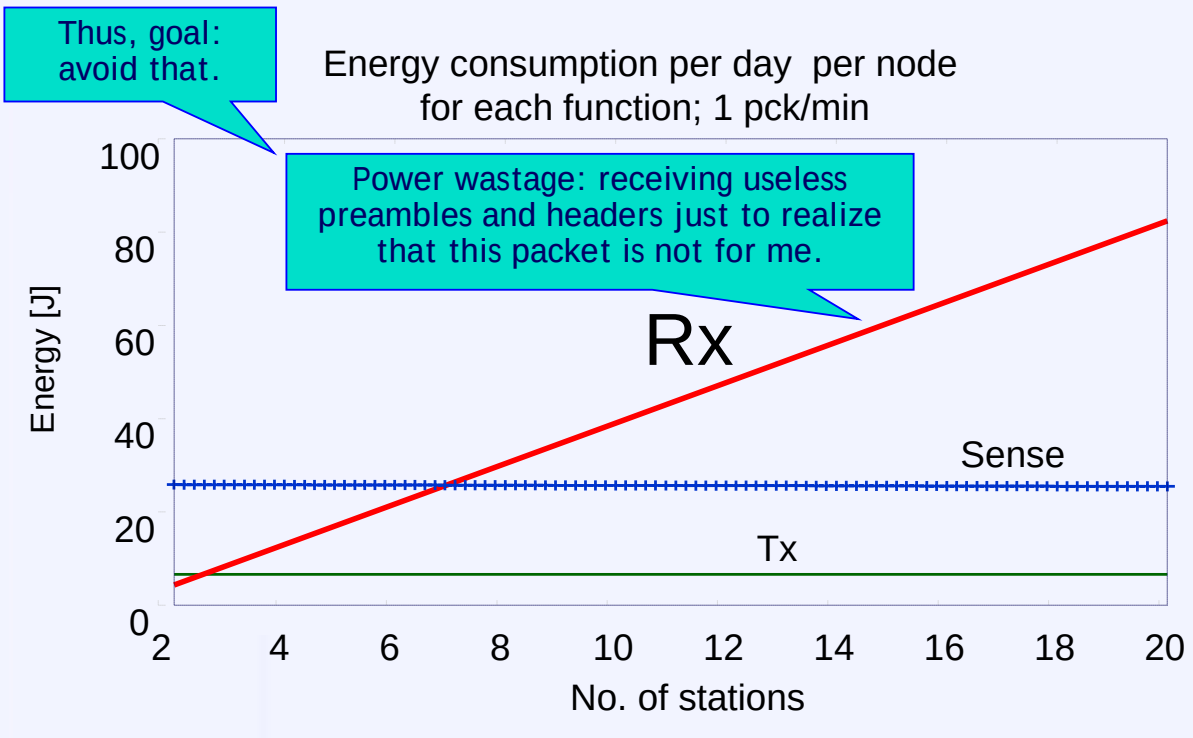
- Low Power Listening (LPL): sender transmits preamble before actual data, potential receivers regularly probe the channel
  - B-MAC: preamble length > receiver sleep period ⇒ receiver will not miss it
  - Problem: power wastage for sending long preamble
    ⇒ several mechanisms improve this: WiseMAC, X-MAC, SCP/MAC, ...

A simple model of B-MAC

Low level MAC	802.15.4
Idle power	1 µW
Sense power	30 mW
Rx power	60 mW
Tx power	25 to 50 mW



B-MAC energy consumption: detail per function

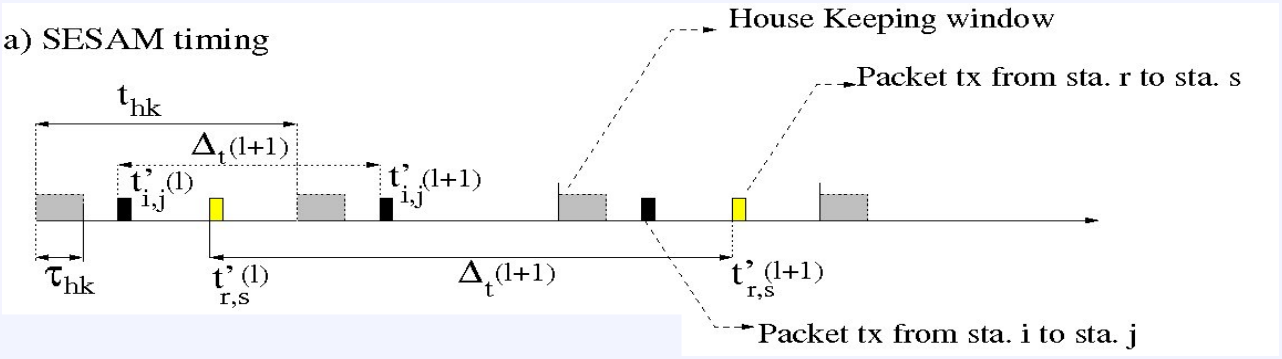


SESAM

- Idea: exploit special properties of WSNs
  - WSNs are not general purpose; they fulfill special tasks
  - Very low traffic on average
  - Mediated by a routing middle-layer
  - Stable and predictable traffic relations toward sinks
    - distributed data fusion is interesting, but rarely used and supported
  - Alarms must be supported, but response time of seconds is O.K.

- Design goals
  - Reduce needless sensing, minimize overhearing
  - Decouple sensing traffic from alarms and management
  - Minimize global coordination (no TDMA) and signaling
  - Use standard & available low level routines
  - Consider simple, pairwise (i,j) coordination
  - App. often knows next transmission time ⇒ let's use this for coordination

SESAM: global timing

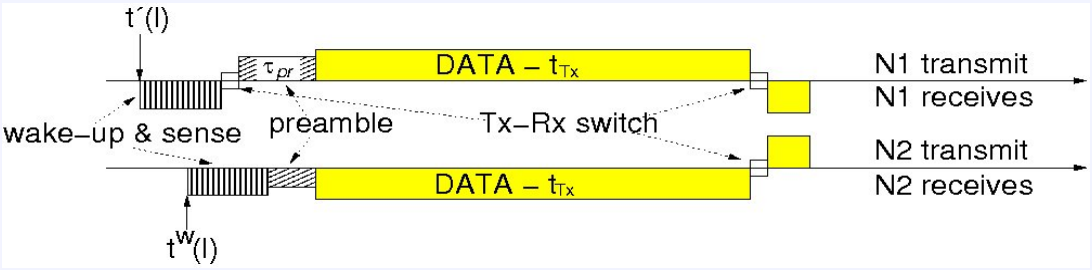


- With every packet, Tx station tells Rx station the next transmission time
  - Rx will then wake up again ⇒ (almost) no useless waking up and listening at all!
  - Traffic relations of Tx/Rx pairs are decoupled except for occasional contention

- Additionally, everyone wakes up in housekeeping periods
  - used for establishing new relations, broadcast, routing, management, bootstrap

SESAM: local timing and collision resolution

- Preambles compensate for clock drifts



- Collision:
  - Behave like a 0-persistent CSMA, i.e. wait for a while and try again
    - Normally bad: goal = de-correlate access in time with overloaded channel
    - Here: channel not overloaded, goal = de-correlate different traffic relations
  - If "for a while" is a constant, traffic relations A-B and C-D will collide again
    ⇒ use pseudo-random sequence, known to Tx and Rx host in a traffic relation
  - Repeat in case of another collision
  - Upon success: **maintain new timing!**

SESAM: collision resolution /2

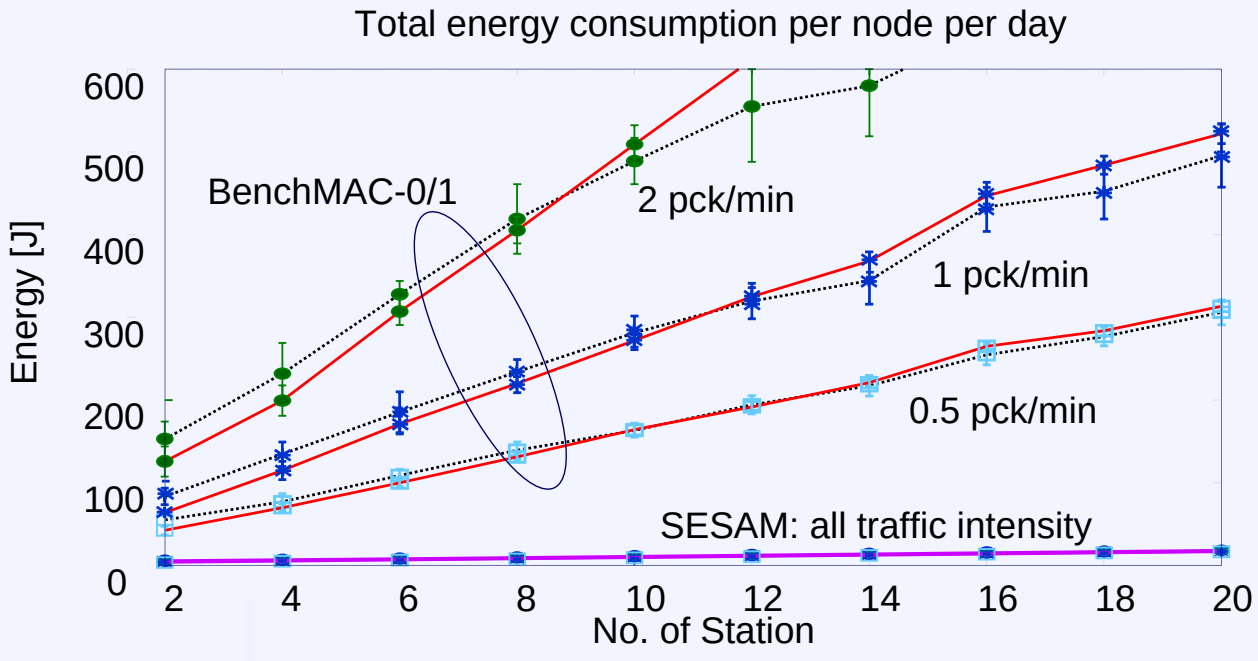
- In case of repeated collisions: how long can we shift the time ahead?
  - Limit: next (normally) scheduled sending time
  - If time shift exceeds it, retransmit packet at next scheduled time instead
    - But let it carry new displacement (MOD "normal" schedule duration)
  - upon failure: abort traffic relation, discard packet, establish again

- What if an ACK is lost?
  - Sender and receiver are desynchronized
    - Sender moves ahead in pseudo-random sequence, receiver does not
  - Solved by retransmission of packet in next "normal" time slot
    - This is when the receiver expects a packet and wakes up
    - The packet will carry the next displacement, so sender and receiver become synchronized again

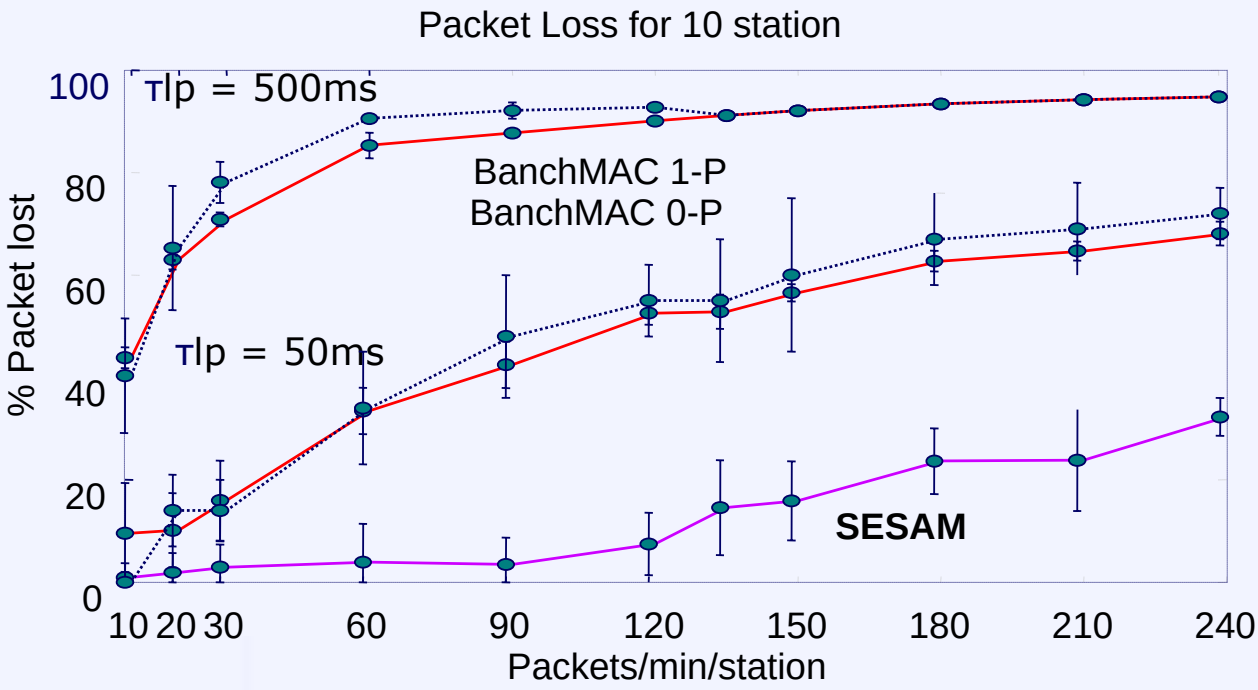
Results

- Comparison with two models of B-MAC
  - BenchMAC-0: CDMA + LPL 0-persistent on contention
  - BenchMAC-1: CDMA + LPL 1-persistent on contention
- Custom made simulations based on PeerSim with the actual consumptions measured on our nodes per each function
- Single collision and housekeeping domain
- Realistic default protocol parameters – packet sizes, durations, transmission power… all given in the paper, not shown here to keep you from falling asleep

Results: Total energy consumption

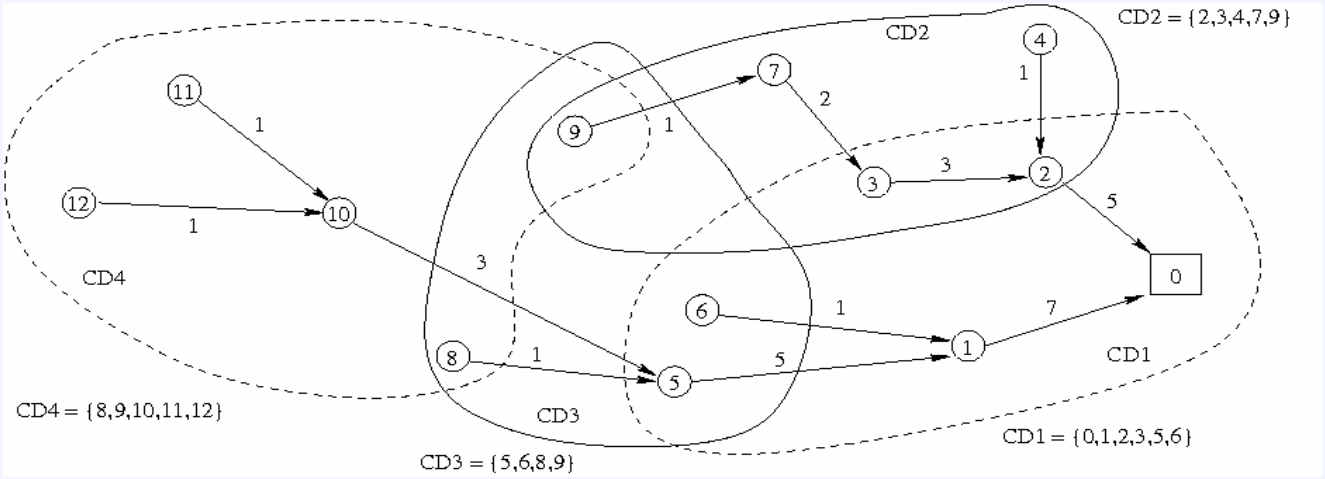


Results: losses for traffic bursts



Ongoing work @ Trento

- Multi-housekeeping management for support of multiple collision and housekeeping domains



- Implementation in TinyOS is planned in the next months

Conclusion

- WSNs are specialized, not general purpose networks
- Energy consumption is critical, often the enabling factor: communication functions (not transmission) dominates
- SESAM can reduce energy consumption by 1-2 orders of magnitude
  - Makes energy a non-problem
  - Years of lifetime out of AA batteries (like your gate remote controller)
- Further investigation + implementation work is underway

Thank you!

Questions?