

Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks

Venkatesh RAJENDRAN, Katia
OBRACZKA and J.J. GARCIA-LUNA-
ACEVES

Presented by Rabin Karki

Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- Schedule Exchange Protocol
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



Introduction

- Sensor networks
 - Consist of a set of interconnected sensor nodes
 - Each node is equipped with one or more sensors and is normally battery operated
 - Nodes communicate with each other via wireless connection



Introduction

- The deployment of sensor network usually done in ad-hoc manner
 - Self-organize into a multi-hop wireless network
- Nodes may be difficult to recharge
- Nodes recharging may not be cost effective
- Major challenge
 - Self adaptive to changes in traffic, node state
 - Prolong the battery life



Some Applications



Habitat Monitoring

Great duck island, Maine

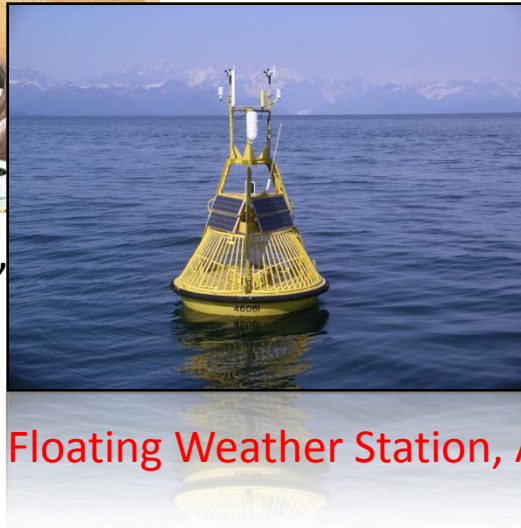


Some Applications



Habitat Monitoring

Great duck island,



Environment Observation and Forecasting System

Floating Weather Station, Alaska



Some Applications



Habitat Monitoring



Great duck island,



Environment Observation and Forecasting System



Weather Station, Alaska

Automated Local Evaluation
in Real Time (ALERT)

Weather Station, Pima, AZ



Some Applications



Habitat Monitoring



Great duck island,

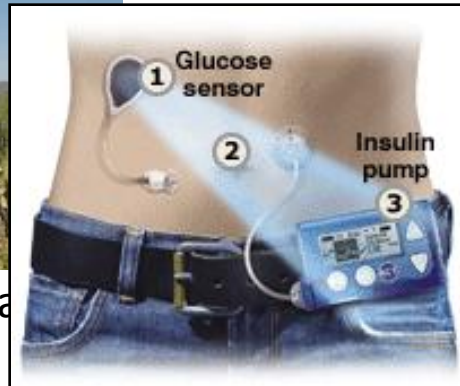


Environment Observation and Forecasting System



Weather Station, Pima

Weather Station, Alaska



Local Evaluation (ALERT)

Glucose Level Monitoring



Some Applications



Habitat Monitoring



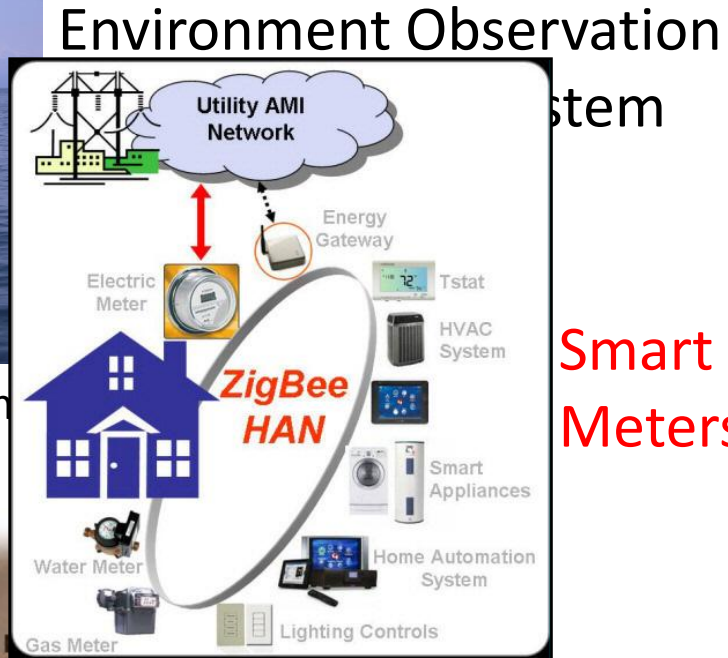
Great duck island,



Weather Station



Weather Station, Pima



Smart Energy Meters



Glucose Level Monitoring



Some Applications

Habitat Monitoring



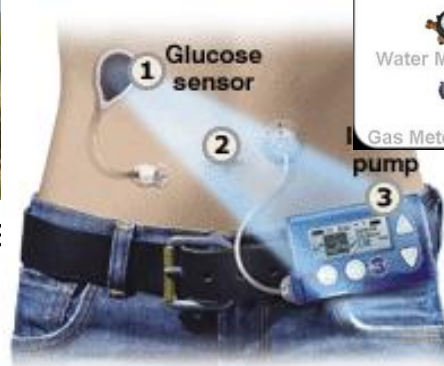
Great duck island,



Weather Station

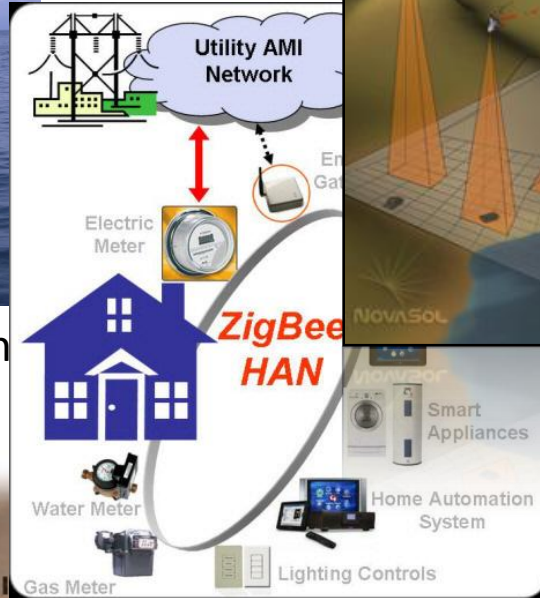


Weather Station, Pima

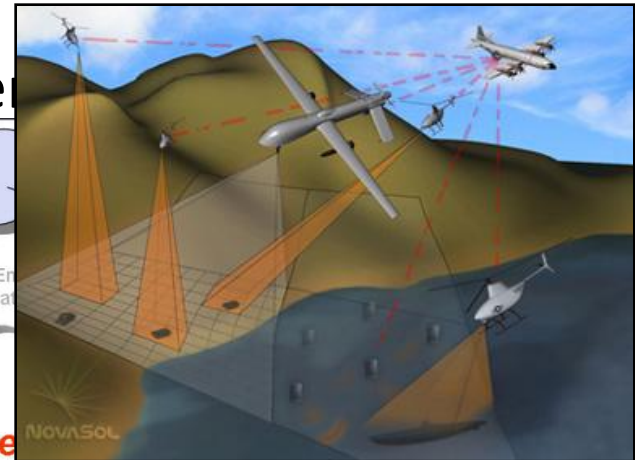


Glucose Level Monitoring

Environment



Military



Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- Schedule Exchange Protocol
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



Background

Categorization of MAC Protocols

- One approach (be nice – *share*)
 - Avoid interference by scheduling nodes on sub-channels
 - **TDMA** (Time-Division Multiple Access)
 - **FDMA** (Frequency-Division Multiple Access)
 - **CDMA** (Code-Division Multiple Access)
- Another approach (Compete/*contend*)
 - Don't pre-allocate transmissions, compete
 - ALOHA (Transmit, collision? Yes => retransmit later)
 - Carrier Sense (802.11)



Background: Power aware contention-based protocols

PAMAS (Power aware multi-access protocol)

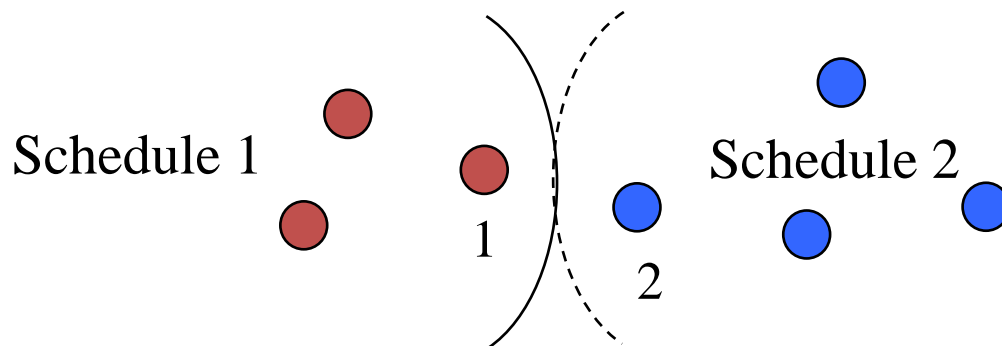
- Designed for multi-hop wireless networks
- Saves energy by avoiding overhearing
- Uses out-of-band signaling i.e. RTS-CTS message exchange takes place over a signaling channel that is separate from the channel used for packet transmissions.
- Every node in the system makes the decision to power off independently.
- A node knows if a neighbor is transmitting because it can hear the transmission (over the data channel).
- Likewise, a node (with a non-empty transmit queue) knows if one or more of its neighbors are receiving because the receivers transmit a busy tone when they begin receiving a packet (and in response to RTS transmissions).



Background: Power aware contention-based protocols

Sensor-MAC

- Like PAMAS, S-MAC also avoids overhearing
- But uses in-channel signaling
- Neighbors synchronize sleep schedules
 - Nodes periodically broadcast schedules
 - New node tries to follow an existing schedule



- Nodes on border of two schedules follow both



Background: Collision free protocols

- But increased load increases the probability of collisions of control and data packets in any contention-based schemes
- Resulting in degraded channel utilization and reduced battery life
- This motivates towards the development of schedule based transmission schemes

NAMA (Node Activation Multiple Access)

- For each time slot, only one transmitter per two-hop neighborhood is selected
- But does not address energy conservation (non-transmitting nodes switch to receiver mode)



Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- Schedule Exchange Protocol
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



TRAMA: Overview

- Energy efficient, collision free transmission attained by
 - Transmission schedules that avoid collision of data packets
 - Having nodes switch to low power mode when there is no data packet destined to those nodes
- Adequate throughput & fairness achieved by
 - Transmitter election algorithm that is inherently fair and promotes channel reuse
- Supports for unicast, broadcast and multicast traffic



TRAMA: Overview

- Nodes exchange
 - Their two-hop neighborhood information
 - The transmission schedule specifying the intended receivers in chronological order
- Nodes that should transmit and receive during each time slot are then selected according to that information



TRAMA: Overview

Consists of three components

- **Neighbor Protocol (NP)**: to transmit two-hop neighbor information
- **Schedule Exchange Protocol (SEP)** : to exchange schedules
- **Adaptive Election Algorithm (AEA)** : to select the transmitters and receivers using neighborhood and schedule information. All other nodes can then sleep.



TRAMA: Overview

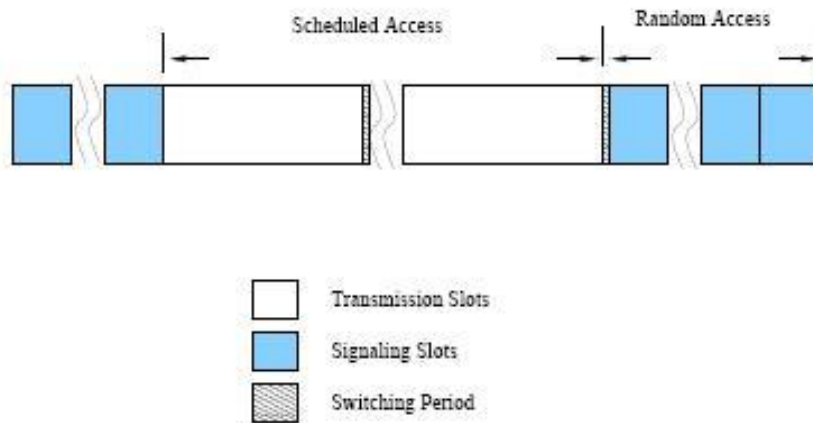


Figure 1: Time slot organization

- Uses single, time-slotted channel for both data and signaling transmissions.



TRAMA: Overview

- Time slot duration is larger than typical clock drifts
- Time slots of approx. 46ms are used in the paper, so drifts in the order of ms can be tolerated although they are typically in the order of μs
- So very simple and cheap-to-implement timestamp mechanisms can be used for node synchronization (*post-facto synchronization*, for example)
- In the paper, transmission slots are 7x longer than signaling slots



TRAMA: Neighbor Protocol

- NP propagates one-hop neighbor info during the *random access* period
- TRAMA starts in random access mode
 - Each node transmits by selecting a slot randomly
 - More dynamic networks require more frequent random access periods
 - All nodes have to be in Tx or Rx mode during this period
 - Node addition or deletion is done during this period



TRAMA: Neighbor Protocol



(a) Signal Header

- Packets carry incremental neighborhood updates
- If there have been no changes since the last update, signaling packets are sent as mere “keep-alive” beacons
- If the node doesn't hear back for a certain period of time, it times out and retransmits
- By the end of random access period, all nodes will have the information about two-hop neighbors with 0.99 probability



Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- **Schedule Exchange Protocol**
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



TRAMA: Schedule Exchange Protocol

- *Transmission slots* are used for collision free data transmission and schedule propagation
- A node has to announce its schedule using SEP before starting actual transmissions
- SEP maintains consistent schedule information across neighbors and periodically updates it



TRAMA: Schedule Exchange Protocol

Schedule Generation

- Each node computes *SCHEDULE_INTERVAL* based on the rate at which packets are produced by the higher layer
- The node then pre-computes the number of slots in the interval $[t, t + SCHEDULE_INTERVAL]$ for which it has the highest priority among its two-hop neighbors
- The priority of node u at time slot t is

26 $prio(u, t) = hash(u \oplus t)$



TRAMA: Schedule Exchange Protocol

- Slots thus obtained are called “*winning slots*”
- The node then announces the intended receivers for those slots
- If it doesn't have enough data to fill up all the “*winning slots*”, it announces that it has “*given up*” those slots (referred to as “*vacant slots*” later)
- Last winning slot is reserved for broadcasting node's schedule for next interval



TRAMA: Schedule Exchange Protocol

- Nodes announce their schedule via *schedule packets*
- Receiver address is not required. Bitmap of length equal to the number of one-hop neighbors is used instead
- The neighbors are ordered by their identities in the bitmap
- This decreases the payload and makes multicast/broadcast easier



TRAMA: Schedule Exchange Protocol

- For *vacant slots*, the node announces zero bitmap
- These slots with zero bitmap can be used by other nodes in the two-hop neighborhood
- The slot after which all the *winning slots* go unused is called *ChangeOver* slot.
- All nodes have to listen during the *ChangeOver* slot to synchronize their schedule



TRAMA: Schedule Exchange Protocol

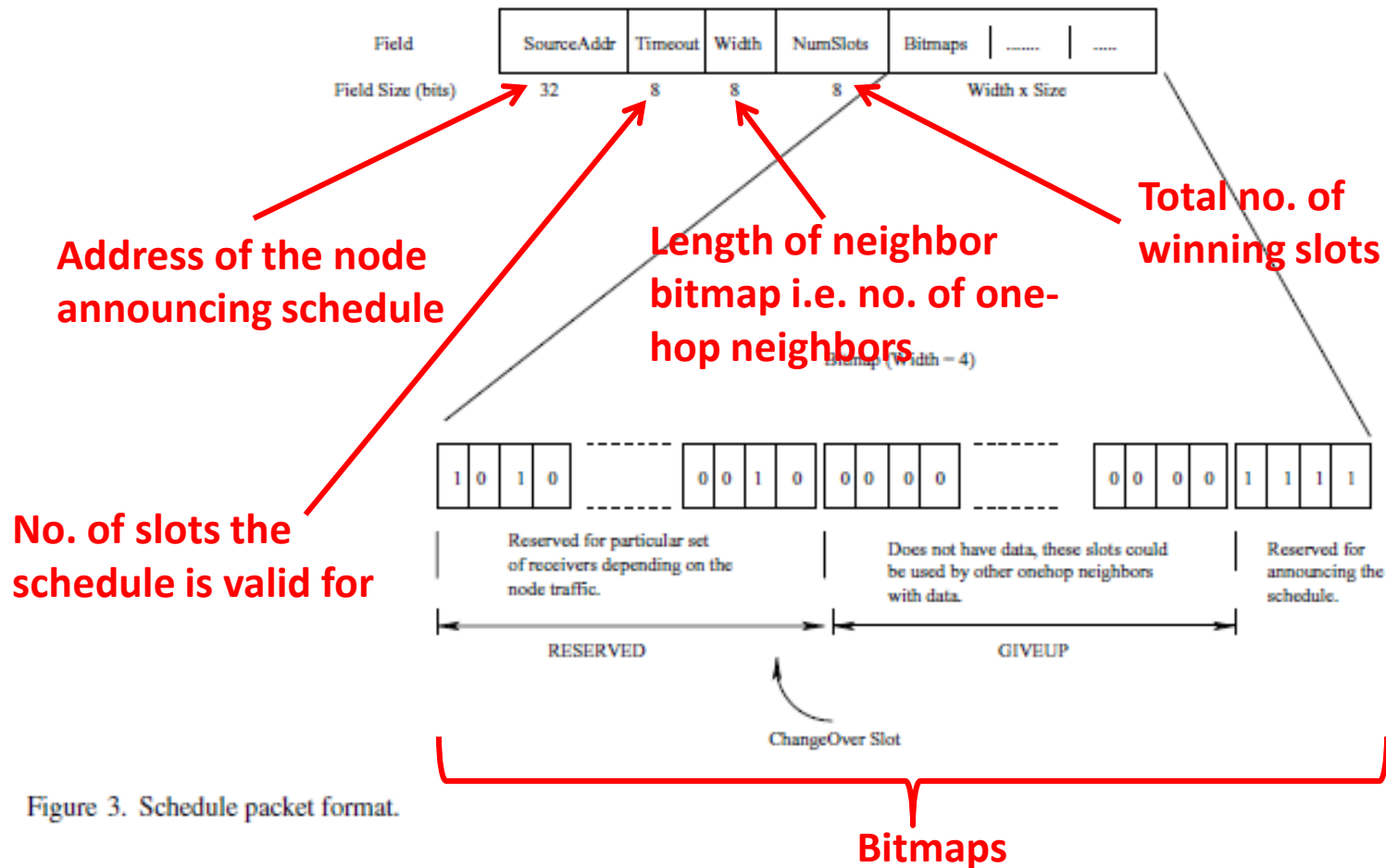


Figure 3. Schedule packet format.



TRAMA: Adaptive Election Algorithm

- At any given time slot t , the state of a given node u is determined by u 's two-hop neighborhood information and the schedules announced by u 's one hop neighbors
- At any given slot t , a node can be in one of three states *i)* transmit (TX) *ii)* receive (RX) or *iii)* sleep (SL)
 - A node u at t is in TX if *i)* u has the highest priority and *ii)* u has data to send
 - It is in RX if it is the intended receiver of the node which is in TX in the current slot.
 - Otherwise it can switch itself off to SL state



TRAMA: Adaptive Election Algorithm

Table 1
Notations and terminologies.

N2 (u)	Set of neighbors of node u which are two-hops away.
N1 (u)	Set of neighbors of node u which are one-hop away.
CS (u)	u 's <i>Contending Set</i> is the set of nodes in u 's two-hop neighborhood such that $\{u \cup \mathbf{N1}(u) \cup \mathbf{N2}(u)\}$.
$tx(u)$	<i>Absolute Winner</i> is the node with the highest priority in CS (u).
$atx(u)$	<i>Alternate Winner</i> is the node which has the highest priority among u 's one-hop neighbors, i.e., over the set $\{u \cup \mathbf{N1}(u)\}$.
PTX (u)	<i>Possible Transmitter Set</i> is the set of all nodes in $\{u \cup \mathbf{N1}(u) - atx(u)\}$ that satisfy the condition given in equation (2).
NEED (u)	<i>Need Contender Set</i> is the set of nodes in $\{\mathbf{PTX}(u) \cup u\}$ that are in need of additional transmission slots.
$ntx(u)$	<i>Need Transmitter</i> is the node with the highest priority among the set of nodes NEED (u) containing valid synchronized schedule.

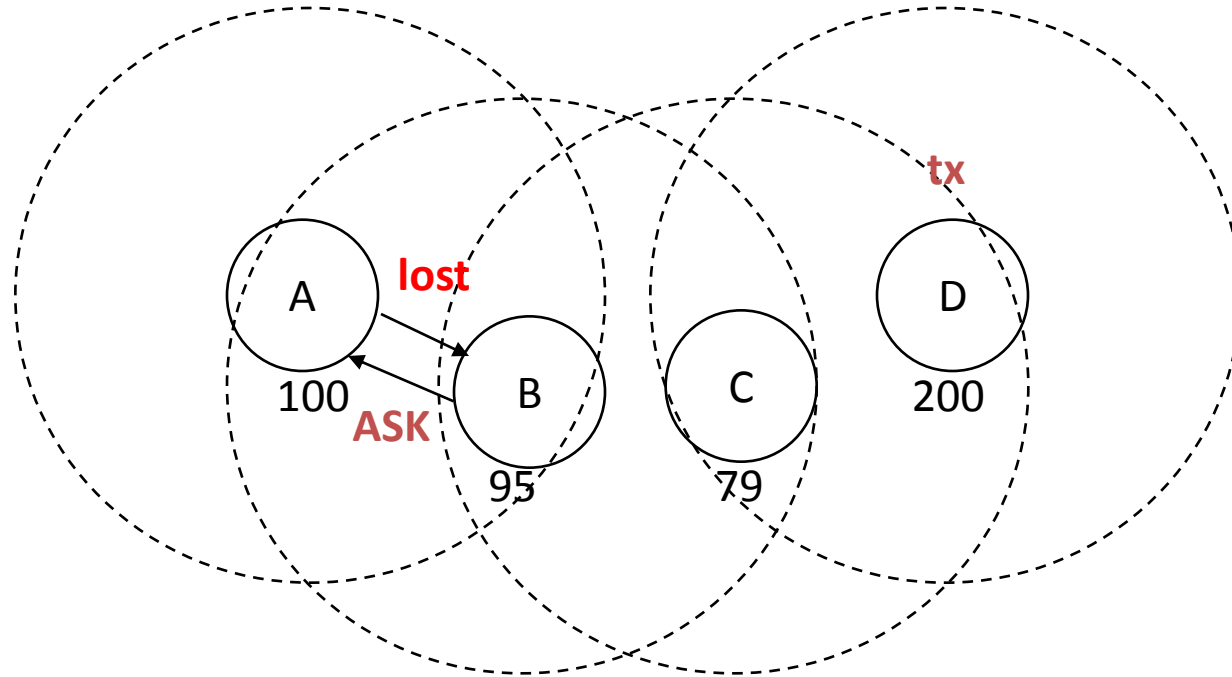


TRAMA: Adaptive Election Algorithm

- Whenever a node becomes *Absolute Winner* for a particular time slot, and has announced non-zero bitmap, no other node in its two-hop neighborhood will transmit in this slot – **for sure**.
- If it is not the *Absolute Winner*, it won't know who the actual transmitter for that particular slot is. This can lead to inconsistency.
- Let's look at an example:



TRAMA: Adaptive Election Algorithm



Inconsistency problem

Happens only when *Alternate Winner* is hidden from the *Absolute Winner* i.e., they are three hops away.



Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- Schedule Exchange Protocol
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



Experimental Setup

- Simulation platform
 - Qualnet
- Physical layer model
 - Based on TR1000
- 50 nodes are uniformly distributed over a 500m x 500m area
- 6 one-hop neighbors on average
- 17 two-hop neighbors on average
- 2 different types of traffic load
 - Synthetic data generation
 - Data gathering application



Performance Metrics

- **Average Packet Delivery Ratio**: ratio of number of packets received to the number of packets sent, averaged over all the nodes
- **Percentage Sleep Time**: ratio of number of sleeping slots to the total slots averaged over the entire network
- **Average Queuing Delay**: average delay for the packet to be delivered to the receiver
- **Average Sleep Interval**: average length of sleeping interval. Measure of no. of radio mode switchings.

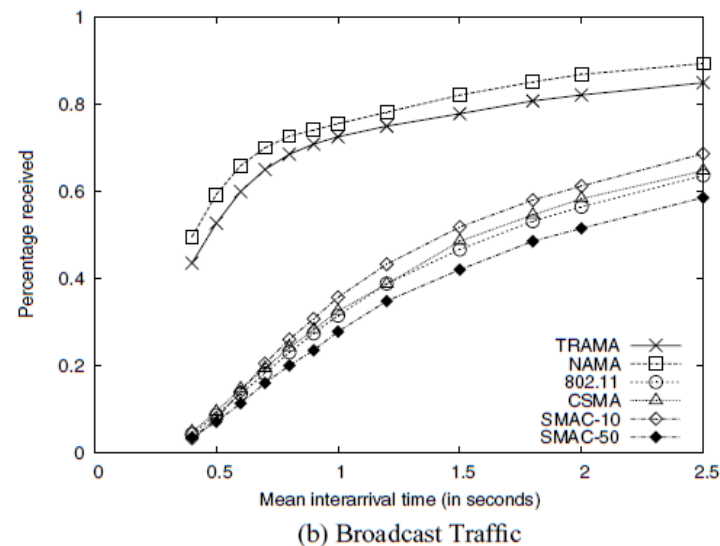
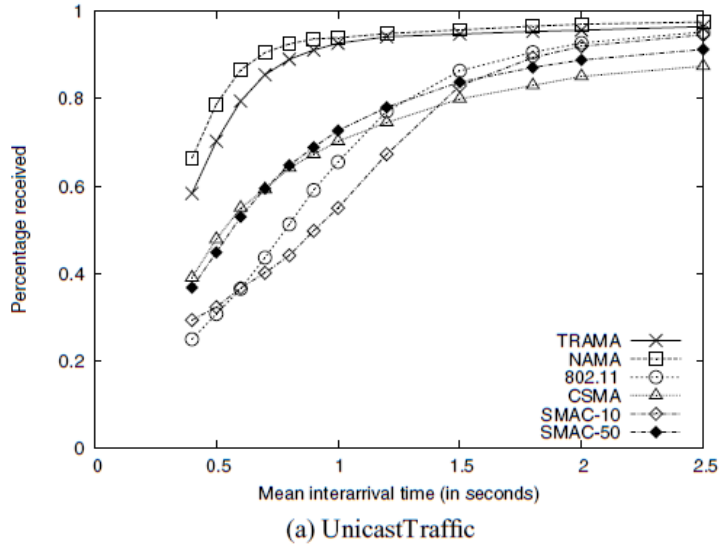


Synthetic Data Generation

- Data generated using exponential inter-arrival time varying rate from 0.5 to 2.5 secs
- All nodes in the network generate traffic based on that distribution
- A neighbor is randomly selected as the next hop
- Tested for unicast and broadcast traffic



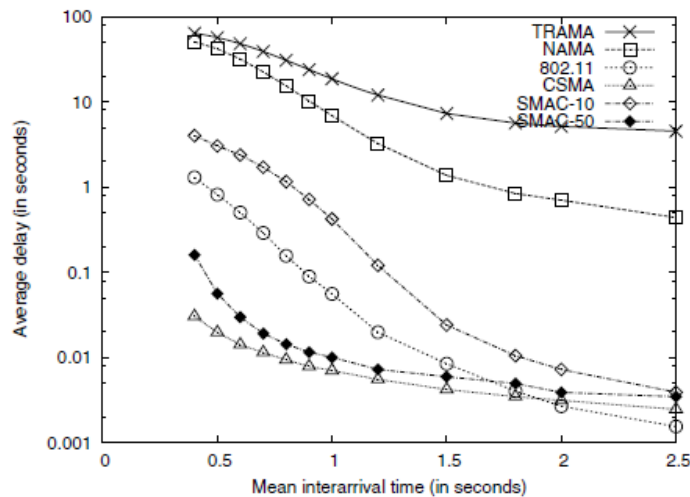
Simulation Results: Synthetic Traffic (Packet delivery ratio)



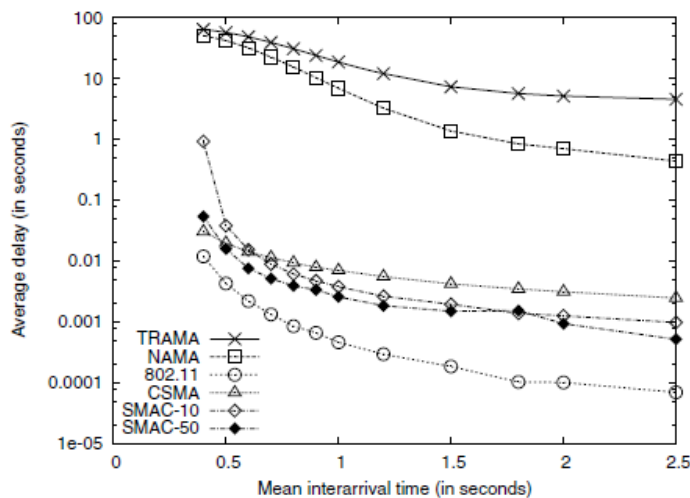
- Schedule based MACs achieve better delivery
- Broadcasting decreases the delivery ratio significantly in collision-based protocols (*kind-of obvious*).



Simulation Results: Synthetic Traffic (Average queuing delay)



(a) Unicast Traffic

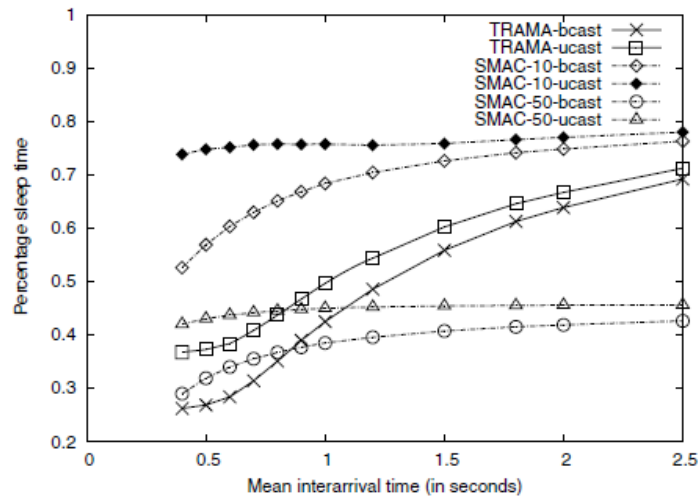


(b) Broadcast Traffic

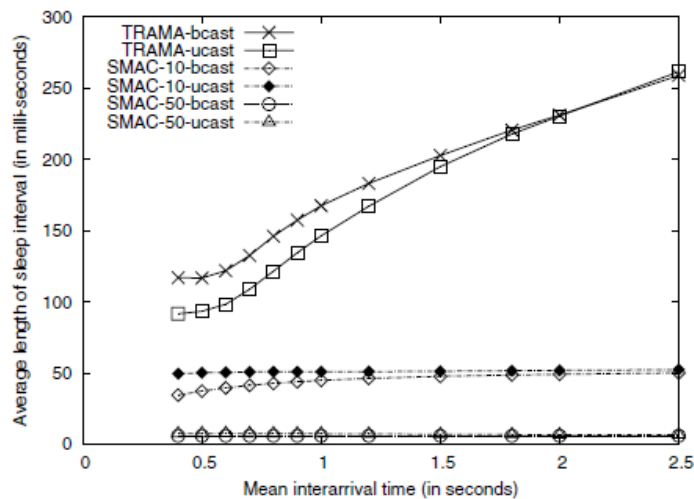
- Schedule based MACs incur higher queuing delay
- TRAMA has more queuing delay than NAMA because of the schedule info propagation overhead



Simulation Results: Synthetic Traffic (Energy savings & avg. sleep interval)



(a) Percentage Energy Savings



(b) Average Sleep Interval

- S-MAC with 10% duty cycle has higher percentage sleep time, but average length of sleep interval is much lower.
- So the overhead for mode switching is higher in S-MAC



Data Gathering Application

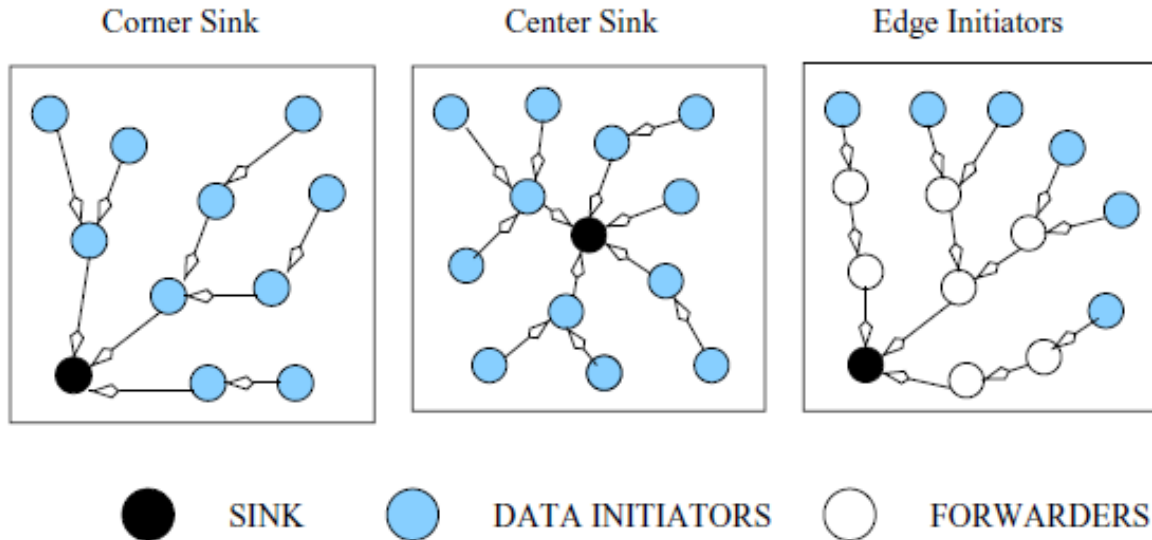
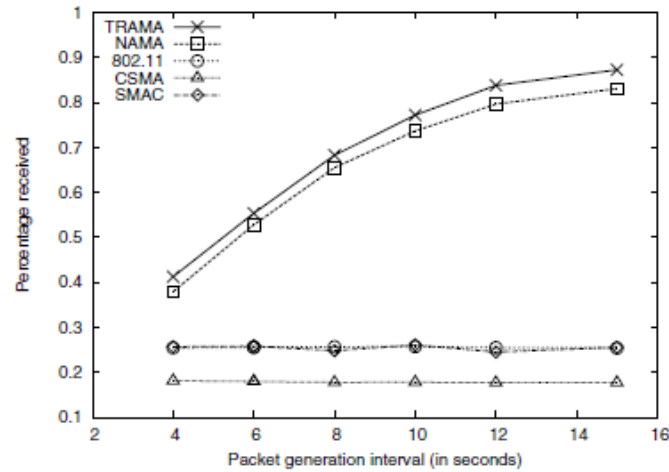


Figure 7. Data gathering application.

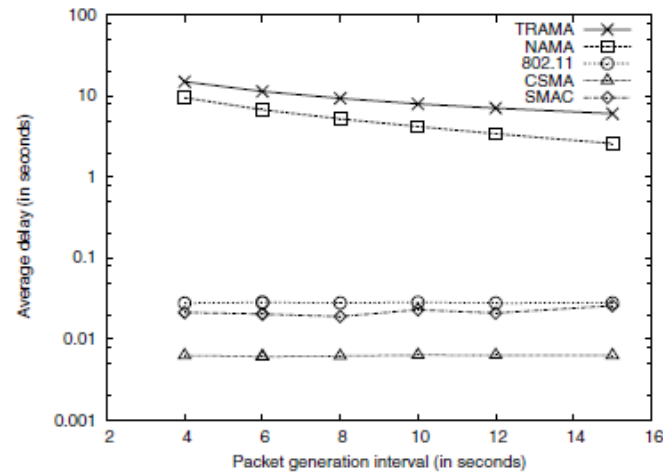
- Sink sends out a broadcast query requesting data from the sensors
- Sensors respond back with the data
- Simple reverse-path routing used to forward data from sensors to the sink



Simulation Results: Data Gathering Application



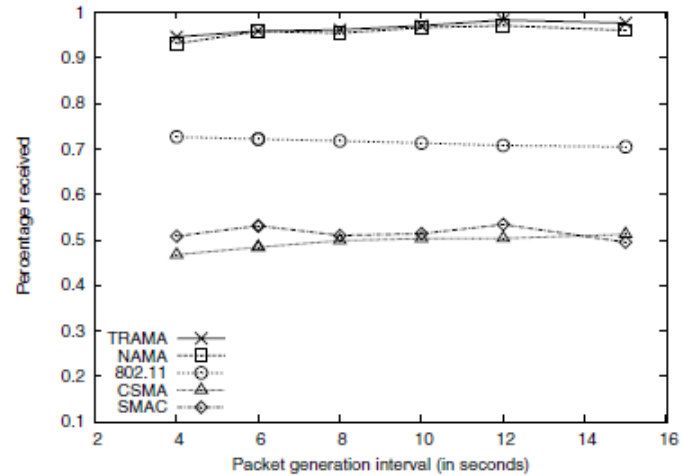
(a) Average delivery ratio



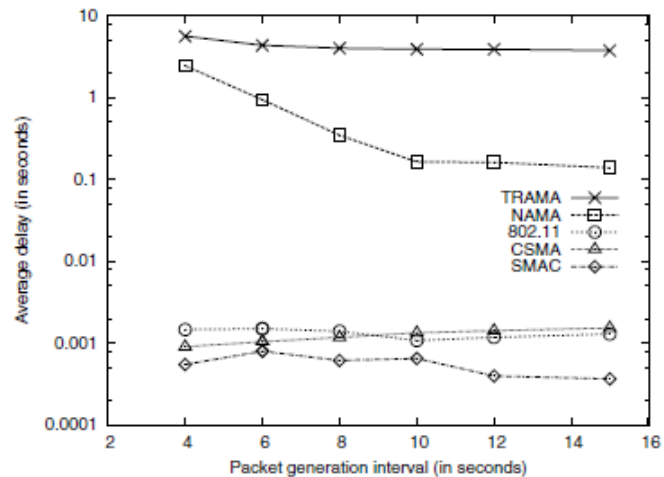
(b) Average queuing delay



Simulation Results: Data Gathering Application



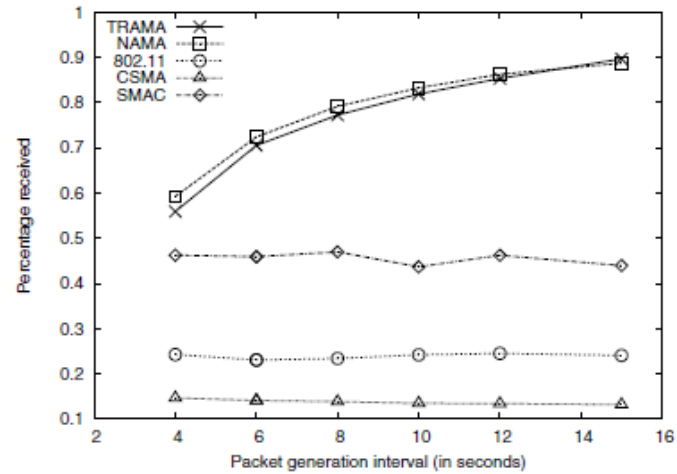
(a) Average delivery ratio



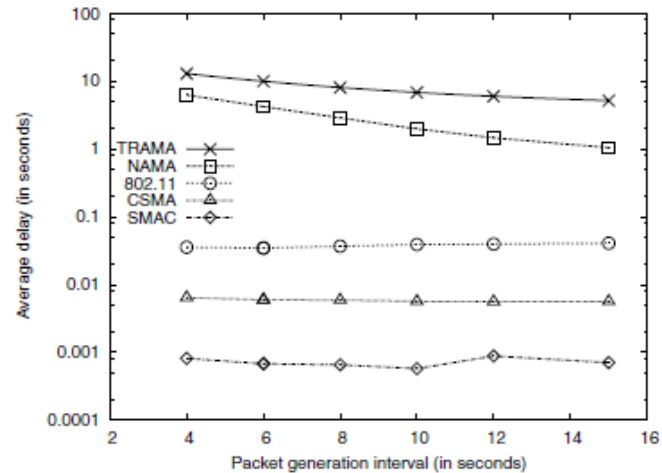
(b) Average queuing delay



Simulation Results: Data Gathering Application



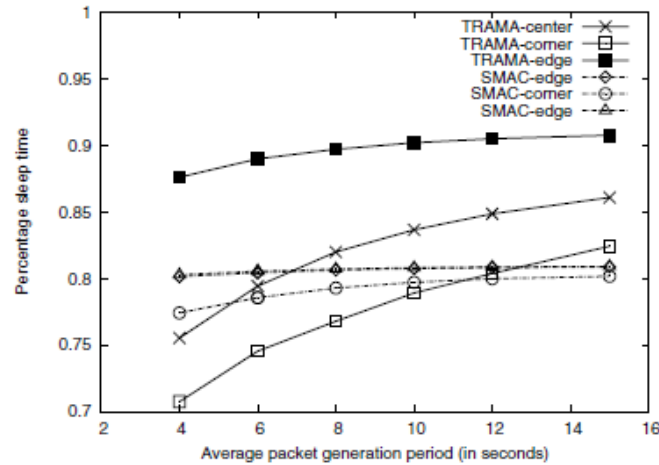
(a) Average delivery ratio



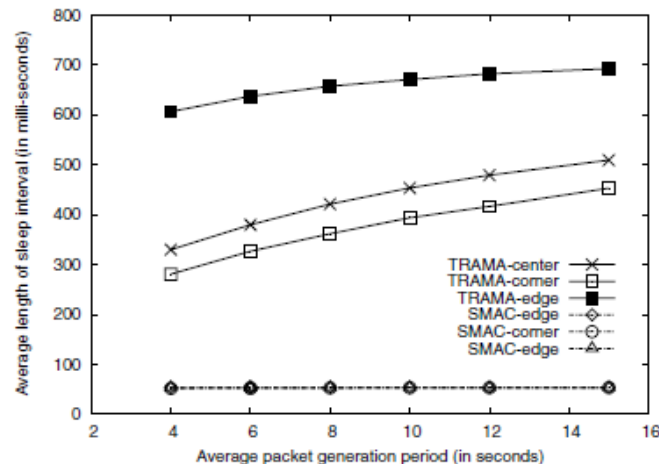
(b) Average queuing delay



Simulation Results: Data Gathering Application



(a) Percentage Energy Savings



(b) Average Sleep Interval



Outline

- Introduction
- Background
- Overview of TRAMA
- Neighbor Protocol
- Schedule Exchange Protocol
- Adaptive Election Algorithm
- Experimental Setup
- Simulation Results
- Conclusions



Conclusion

- Traffic-based scheduling to avoid wasting the empty slots
- And to switch nodes to a low-power standby mode when they are neither transmitter nor receivers
- Significant energy savings (sleep % upto 87%)
- Higher throughput (40% over S-MAC and 20% over 802.11)
- Higher delay – in general
- Well suited for application not-so-delay-sensitive



Acknowledgements

Some slides, texts or images taken from

- Lecture slides by *A. Krugar, University of Michigan*
- Presentation on TRAMA by *Yung-Lin Yu*
- Presentation on MACs by *Huaming Li*



AEA Pseudocode (for your viewing pleasure)

```
1 Compute  $tx(u)$ ,  $atx(u)$  and  $ntx(u)$ 
2 if ( $u = tx(u)$ ) then
3   if ( $u.isScheduleAnnouncedForTx = TRUE$ ) then
4     let  $u.state = TX$ 
5     let  $u.receiver = u.reported.rxd$ 
6     Transmit the packet and update the announced schedule
7   else if ( $u.giveup = TRUE$ ) then
8     call HandleNeedTransmissions
9   endif
10  else if ( $tx(u) \in N1(u)$ ) then
11    if ( $tx(u).announcedScheduleIsValid = TRUE$  AND  $tx(u).announcedGiveup = TRUE$ ) then
12      call HandleNeedTransmissions
13    else if ( $tx(u).announcedScheduleIsValid = FALSE$  OR  $tx(u).announcedReceiver = u$ ) then
14      let  $u.mode = RX$ 
15    else
16      let  $u.mode = SL$ 
17      Update schedule for  $tx(u)$ 
18    endif
19  else
20    if ( $atx(u)$  hidden from  $tx(u)$  AND  $atx(u) \in PTX(u)$ ) then
21      if ( $atx(u).announcedScheduleIsValid = TRUE$  AND  $atx(u).announcedGiveup = TRUE$ ) then
22        call HandleNeedTransmissions
23      else if ( $atx(u).announcedScheduleIsValid = FALSE$  OR  $atx(u).announcedReceiver = u$ ) then
24        let  $u.mode = RX$ 
25      else
26        let  $u.mode = SL$ 
27        Update schedule for  $atx(u)$ 
28      endif
29    else
30      call HandleNeedTransmissions
31    endif
32  procedure HandleNeedTransmissions
33  if ( $ntx(u) = u$ ) then
34    let  $u.state = TX$ 
35    let  $u.receiver = u.reported.rxd$ 
36    Transmit the packet and update the announced schedule
37  else if ( $ntx(u).announcedScheduleIsValid = FALSE$  ||  $ntx(u).announcedReceiver = u$ ) then
38    let  $u.mode = RX$ 
39  else
40    let  $u.mode = SL$ 
41    Update the schedule for  $ntx(u)$ 
42  endif
```

¿Questions?

