Energy-Efficient Communication in Multi-Channel Single-Hop Sensor Networks

Amol Bakshi and Viktor K. Prasanna
Department of Electrical Engineering
University of Southern California
Los Angeles, CA USA 90089
{amol, prasanna}@usc.edu

10th Intl. Conf. On Parallel and Distributed Systems
July 08, 2004

Outline

- Motivation
- System model
- (N,p,k_1,k_2) routing with multiple data channels
- Handling node failures
- Related work
- Remarks

Wireless Sensor Networks

- Ad hoc networks of smart sensors with applications such as environment monitoring, target tracking, etc.
- A sensor node has limited computation, communication and storage capacity
- WSNs are application-specific
  - The set of tasks and the general pattern of information flow is usually known at design time
- Communication in WSNs
  - Unlike the Internet, point-to-point communication is not the primitive in WSNs
  - Application-specific protocols are designed for collaborative communication patterns to achieve energy efficiency
  - Ultimately, sensor network applications will be composed from a set of templates - `building blocks' – for commonly encountered patterns of collaboration

---

Big Picture

- Objective: A methodology and framework for composing sensor network applications from a library of high-level building blocks
- "Program the network, not the node"

- Provide a set of abstractions that allow the end user to describe the application in terms of sensed events and the related data processing
- The framework maps this description onto a target network graph and uses structured communication protocols that match the patterns of information flow

- This paper develops an efficient protocol for a generic communication primitive – (N,p,k_1,k_2) routing – for single-hop clusters

Structured Communication

- A routing problem where the pattern is known in advance
- Examples: one-to-all, all-to-one, many-to-many, all-to-all, permutation, etc.

- (N, p, k_1, k_2) routing
  - A general primitive for structured communication
  - N packets to be transferred among p nodes
  - Each node transmits at most k_1 packets
  - Each node receives at most k_2 packets
  - Each node knows destination IDs of its packets
  - No node knows the source IDs of packets for itself

TDMA vs. Contention-based

- TDMA (Time Division Multiple Access)
  - Requires global time synchronization
  - Typically involves a centralized coordinator for assignment of time slots to nodes based on demand
  - Efficient because transmissions never collide

- Contention-based channel access
  - No requirement of global time synchronization
  - No centralized coordinator; all nodes act independently
  - Inefficiency due to possibility of collisions

- Can *a priori* knowledge of the communication pattern be exploited to achieve collision-free TDMA-like behavior using decentralized, negotiation-based channel access?

System Model (1)

- Homogeneous network of p nodes: S(1), …, S(p)
- Each node has a unique identifier in the range [1,p]
- Nodes are initially time synchronized and remain time synchronized for the duration of routing
- Communication channels
  - Multiple data channel for data packets
  - One low power control channel for coordination
- Control radio can be tuned to only one frequency
- Data radio can be tuned to one of f_d frequencies
- Transmission over data channel involves r_1 times more latency and r_2 times more energy than over control channel
- State-of-the-art: r_1 in the range of 5-10, r_2 is few tens

---

System Model (2)

- A radio can be in Transmit, Receive or ShutDown state
- Energy consumption per packet is same for Transmit and Receive states; zero for ShutDown
- Instantaneous intra-node signaling between radios
- If data (control) radio in Receive state receives two or more simultaneous transmission, packet is lost
- Computation energy and latency is ignored

Our Protocol: Summary

- Functionality
  - Packet transmissions are scheduled over the low power control channel
  - The goal is to maximize bandwidth utilization
  - All co-ordination is decentralized and adaptive to faults
- Robustness
  - Each packet transfer is preceded by negotiation phase
  - Packet is transferred only if both the sender and the receiver is alive
  - Packets are transferred directly from the source to the destination, without intermediates
- Performance
  - Designed for few tens or at most few hundreds of nodes
  - k-times speedup for k data channels, compared to single channel case
  - for $k_1=k_2$, protocol is energy balanced

Preliminaries

- One Request To Send (RTS) message, followed by corresponding Clear To Send (CTS) constitutes one transaction over the control channel
- Transfer of one data packet from source to destination constitutes one transaction over the data channel

- Time is divided into fixed-size frames over both the data and control channels
- One control frame (CF) is the time required for one control transaction
- One data frame (DF) is the time required for one data transaction

State Information

- Each node maintains two data structures: CAV and PCV
- Channel Allocation Vector (CAV)
  - f_d tuples, where tuple i corresponds to data channel i
  - Tuple records sender, destination, and end time of ongoing transmission (if any) over that channel
- Packet Count Vector (PCV)
  - p tuples, where tuple i corresponds to node S(i)
  - Tuple records the number of packets transmitted by S(i) and received by S(i) till that time
- f_d and p are small compared to N; storage requirements are not excessive
- All coordination is performed on control channel and each node monitors the control channel
- Each node has enough information to maintain CAV and PCV

Phase I: Pre-processing

- If k_1 and k_2 are different for each node, the objective of pre-processing is to communicate this information to every node in the network
- For unequal k_1
  - Pre-processing phase consists of p CFs
  - In CF[i], S(i) broadcasts its value of k_1
  - All nodes are awake for all p CFs
- For unequal k_2
  - Pre-processing phase of p rounds of p CFs each
  - In CF[i] of round j, node j transmits the number of packets it has for node I
  - Each node is awake for 2p-2 slots
- Nodes can turn themselves off once they have transmitted k_1 packets and received k_2 packets

Phase II: Data Transfer

- Initially, all data channels are idle and all data radios are in ShutDown mode
- Queue of outgoing packets at each node is sorted in increasing order of destination ID

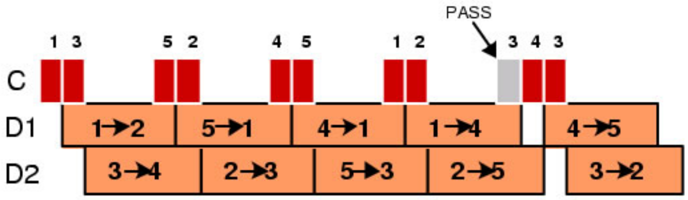
- S[1] owns CF[1]
- S(i) inspects its CAV and waits till at least one channel becomes available
- S(i) traverses outgoing queue to determine if there is at least one packet whose destination S(j) is not busy
- If such a packet is found, RTS/CTS exchange takes place between S(i) and S(j) over control channel and packet transfer starts over any available data channel
- If no such packet, S(i) broadcasts a PASS packet

Phase II: Data Transfer

- After S(i) has broadcast either an RTS/CTS or a PASS packet, who owns the next CF?
- Owner is the next higher node S(j) that
  - has not withdrawn from the protocol, and
  - is not busy transmitting or receiving during the next CF

- Each node can use CAV and PCV to derive this independently and therefore, transfer of ownership does not require explicit messaging
- Why is a PASS packet required?
  - If there is no packet in a node's queue that satisfies previously mentioned criteria, other nodes cannot know this due to the nature of the routing problem
  - PASS packet is required for explicit transfer of ownership
  - However, in a fault-free case, lack of RTS will imply PASS

Illustration



p = 5, f_d = 2, N = 10. Packet distribution: **1**: [2,4]. **2**: [5,3]. **3**: [4,2]. **4**: [5,1]. **5**: [1,3]
C denotes the control channel; D1 and D2 denote the data channels

---

Performance (1)

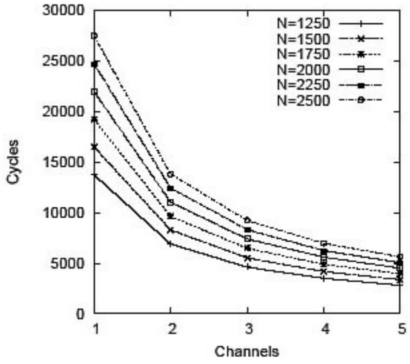


Figure 2: Speedup for Multiple Channels

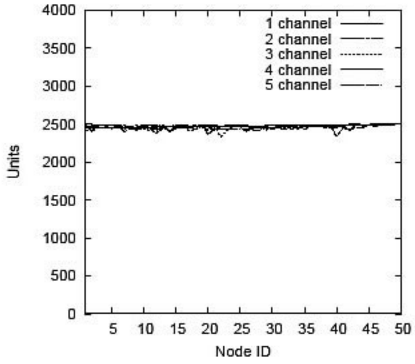


Figure 3: Distribution of Coordination Energy

Performance (2)

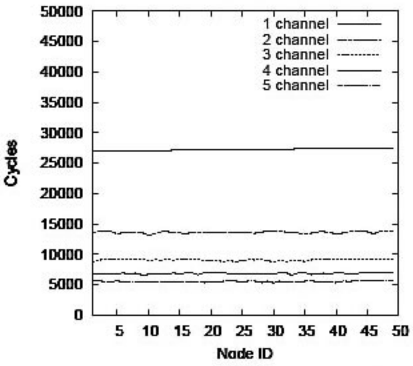


Figure 4: Node Participation (Overall)

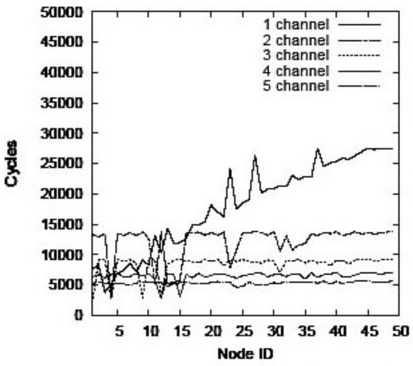


Figure 5: Node Participation (Received)

Handling Permanent Faults

- Fault model: Once failed, a node will remain failed
- How is failure detected?
  - The owner of a given CF is known to all participating nodes
  - At each node, the local PCV indicates if the current owner has outstanding transmissions
  - If the owner broadcasts neither an RTS nor a PASS packet, it is considered failed
- Collaborative response to failure detection
  - Nothing is scheduled over the data channels
  - Failed node is marked ‘withdrawn’ by all nodes
  - Failed node is excluded from further transactions (RTS and CTS)
- Energy and latency savings are achieved because packets destined for that node are never transmitted

Performance: Permanent fault model

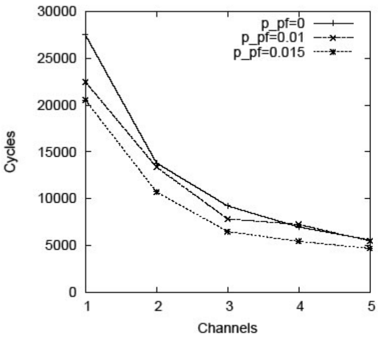


Figure 7: Effect on Routing Latency

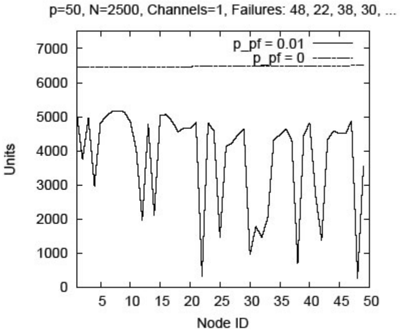


Figure 8: Distribution of Energy Consumption

Related Work

- *Energy-efficient permutation routing protocols in radio networks*, K. Nakano, S. Olariu, A. Zomaya, IEEE TPDS 2001
- *Fault-tolerant and energy-efficient permutation routing protocol for wireless networks*, A. Datta, IPDPS 2003

- Reservation phase followed by data transfer phase
- All transfers are scheduled during reservation phase
- System model
  - Time is divided into equal size slots
  - Simultaneous transmissions in same slot cause collision
  - All nodes are time synchronized
- In-channel signaling: No distinction between latency and energy cost of control and data packets
- Both assume that nodes will NOT fail during the routing

Conclusion

- A simple, decentralized, robust protocol for a generic routing primitive
- High bandwidth utilization, energy balanced and energy efficient
- Applicable for a range of communication patterns specifically in clustered sensor network topologies
- TDMA-like behavior using negotiation on contention-based medium access protocol

- Issues
  - We used a high level model; real world issues (e.g. error rates, retransmissions) will likely degrade performance to some extent
  - Performance depends on technology (values of r_1 and r_2)