A Smart Cross Layer Architecture for Real-time Wireless Sensor Networks

Khurram Khan
Department of Electrical & Computer Engineering
University of Waterloo

ECE: 750
Instructor: Dr Otman Basir
Presentation Outline

- Introduction
- Rap (The proposed Architecture)
- Experimental Design
- Simulation
- Conclusion
MEMS devices & Embedded systems are becoming increasingly networked (via sensors)
Controller-area-networks (CAN) bus in automobiles
Services in large buildings are now run across networks
    e.g. heating, lighting, security

Large-Scale Sensor networks are deployed to perform Distributed Micro-sensing and control of Physical environments
Introduction

**Battlefield Surveillance**

**Health Monitoring**

**Home Automation and Security**

**Disaster management**

**Pedestrian Safety Video Sensors**

Data Communication in Sensor Networks has timing constraints in the form of end-to-end deadlines.
Sensor network protocols should support real-time communication by minimizing the packet deadline miss ratio, i.e., the percentage of packets that meet their end-to-end deadlines.
Introduction

Bio-attack on a part of an airport

Query or Event

Density of the detected virus

coordinates (10,10,20,20)

correlated measurement of some environmental activity

Group of new targets simultaneously entering a security area
RAP Communication Architecture

Provide general service APIs that are suitable for distributed micro-sensing and control in sensor networks

Maximize the number of packets meeting their end-to-end deadlines

Scale well with large number of nodes and hops

Introduce minimum communication and processing overhead
RAP

Sensing/Control Application

Query / Event Service APIs

Interact with RAP through

Submits the Query / Event Registration in a Sensor area which is then sent back to the base station

Query-Event Service

Co-ordination Service

Location-Addressed Protocol

Geographical Forwarding

Velocity Monotonic Scheduling

Prioritized Mac

Dynamic group management and data aggregation among sensors (e.g., multiple sensors coordinate to determine the location of a target through triangulation)

Network Stack
API provides a high-level abstraction to applications, hiding the node location/status

Specify timing constraints of queries

• query\{attribute_list, area, timing_constraints, querier_loc\}

query {
  // Issue a query for a list of attributes in an area
  virus.count, // requires the average density of the viruses
  area=\(10,10,12,12\), // located in an rectangular area \((10,10,12,12)\)
  period=1.5, deadline=5, // every 1.5 s & should reach the base
  // station within an end-to-end deadline of 5 sec
  base=(100,100) // be reported to the base station \((100,100)\) (coordinates to
  // be used by LAP)
}
• register_event{event, area, query}

register_event { // Registers a virus_count query for a virus_found event
  virusFound(0,0,100,100), // If any viruses are found in a rectangular area with coordinates
  query { //
    virus_count, // returns the average density of the viruses
    area=(Xevent-1,Yevent-1,Xevent+1,Yevent+1), // of the 2×2 square area centered at
    period=1.5, deadline=5, // the event location (Xevent,Yevent)
    base=(100,100) // every 1.5 sec with a deadline of 5 sec
  };
  // to the base station at coordinates (100,100)
RAP

**Location-Addressed Protocol**

- Connectionless transport layer in the network stack
- Similar to UDP except that all messages are *addressed by location* instead of IP address

**Unicast**
- Delivers a message to a node that is closest to the destination location, e.g. sensors sending query results to base stations

**Area Multicast**
- Delivers a message to every node in a specified area, e.g. to register for an event or send a query to an area, for coordination among nodes in a local group

**Area Anycast**
- Delivers a message to at least one node in a specified area, e.g. used for sending a query to a node in an area. The node can initiate group formation and coordination in that area
RAP

Geographical Forwarding (GF)

makes a greedy decision to forward a packet to a neighbor if:

1) it has the shortest geographic distance to the packet’s destination among all immediate neighbors

2) it (neighbor) is closer to the destination than the forwarding node
When such nodes do not exist, the Greedy Perimeter Stateless Routing protocol (GPSR) can be used to route packets around the perimeter of the void region.

Each node receiving a data packet marked as in perimeter mode uses the right-hand rule to forward packets to nodes, which are located counterclockwise to the line joining forwarding node and the destination.
Highly Scalable → Uses immediate neighborhood information

Dense network (more nodes / diameter / rate of change of topology) → Efficient greedy forwarding works well

→ #hop approx proportional to distance the packet travels

Location-addressed comm. → No location directory service (reduced management & communication overhead)

Geographical Forwarding (GF)

Packet Scheduling

Deadline-aware

packet’s priority should relate to its deadline

The shorter the deadline, the higher the packet priority

Distance-aware
determines the order in which incoming packets at a node are forwarded to an outgoing link

packet’s priority should relate to its distance from the destination

The longer the distance, the higher the packet priority
\( \text{dis} = 90 \text{ m}; D = 2 \text{ s} \) 

\text{HIGH Priority}

\( \text{dis} = 60 \text{ m}; D = 2 \text{ s} \) 

\text{LOW Priority}
RAP

Velocity Monotonic Scheduling (VMS)

Assigns the “right” priorities to packets (urgency at current hop)
Solves the fairness problem (for packets far away from the Base station)

\[
\text{Priority} = \text{Requested Velocity} \quad \text{Requested Velocity} = \frac{\text{Distance}}{\text{Deadline}}
\]

Deadline
Satisfies Distance &
RAP

dis = 90 m; D = 2 s
HIGH Priority

V = 45 m/s

dis = 60 m; D = 2 s
LOW Priority

V = 30 m/s

D

A

B

C

E

Dis

Low

Priority

High
computes a fixed requested velocity at the sender of each packet (*requested Velocity is fixed on each hop*)

Assume;
a packet is sent from a sender at $(x_0, y_0)$ to a destination at $(x_d, y_d)$, and has an end-to-end deadline $D$ sec

**Static Velocity Monotonic (SVM)**

SVM sets its requested velocity to:

$$V = \frac{\text{dis}(x_0, y_0, x_d, y_d)}{D}$$

Where;

$\text{dis}(x_0, y_0, x_d, y_d)$ is the geographic distance between $(x_0, y_0)$ and $(x_d, y_d)$
High Priority

\[ \text{dis} = 90 \text{ m}; \text{D} = 2 \text{ s} \]
\[ V = 45 \text{ m/s} \]

Low Priority

\[ \text{dis} = 60 \text{ m}; \text{D} = 2 \text{ s} \]
\[ V = 30 \text{ m/s} \]

\[ \text{dis} = 60 \text{ m}; \text{D} = 2 \text{ s} \]
\[ V = 60 / 2 = 30 \text{ m/s} \]

\[ \text{dis} = 90 \text{ m}; \text{D} = 2 \text{ s} \]
\[ V = 90 / 2 = 45 \text{ m/s} \]
Dynamic Velocity Monotonic (DVM)

\[ V = \text{dis} \left( (x_i, y_i, x_d, y_d) \right) \]
\[ (D - Ti) \]

Equ1.

Where;
\( \text{dis}(x_0, y_0, x_d, y_d) \) is the geographic distance between \((x_0, y_0)\) and \((x_d, y_d)\)

Assume;
a packet arrives at a node at location \((x_i, y_i)\); its destination is at \((x_d, y_d)\); it has an end-to-end deadline \(D\) sec, and its elapsed time, i.e., the time it has been in the network, is \(Ti\) sec;

**Dynamically re-calculates the requested velocity of a packet upon its arrival at each intermediate node**
Packet at A
Deadline D = 8 Seconds
Based on the dis, lets assume;
A-B = 4 Sec and B-C = 4 Sec

Scenario 1:
DVM at A
Dis = 10m, Deadline = 8 sec,
Current Ti = 0

\[ V = \frac{\text{Dis} \ (10m)}{8 - 0} \]
\[ V = 1.25 \text{ m/s} \]

DVM at B
Dis = 5m, Deadline = 8 sec,
Current Ti = 4

\[ V = \frac{\text{Dis} \ (5m)}{8 - 4} \]
\[ V = 1.25 \text{ m/s} \]

The velocity of the packet is maintained to meet the deadline.
Packet at A
Deadline D = 8 Seconds
Based on the dis, lets assume;
A-B = 4 Sec and B-C = 4 Sec

Scenario 2:

DVM at A
Dis = 10m, Deadline = 8 sec,
Current Ti = 0

V= Dis (10m) / 8 – 0
V= 1.25 m/s

Slack present, packet arriving at
B not on time (5 Sec)

DVM at B
Dis = 5m, Deadline = 8 sec,
Current Ti = 5

V= Dis (5m) / 8 – 5
V= 1.66 m/s

V= Dis (5m) / 8 – 3
V= 1 m/s

The velocity of the packet is increased to cover up the delay incurred.
The velocity of the packet is decreased to match the priorities.
**RAP**

**Prioritized Queue**

**Single Queue**

*Insert all packets in a single queue, ordered by priority (velocity)*

*If queue is full, higher priority incoming packets overwrite lower priority*

**Adv:** accurately reflects the order of requested velocities, and allows all packets to share the same buffer

**Disadv:** requires implementing a data structure whose insertion time, in the worst case, grows logarithmically in the number of packets
maintain multiple FIFO queues each corresponding to a fixed priority level

Each priority corresponds to a range of requested velocities

packet is first mapped to a priority, and then inserted into the FIFO queue that corresponds to its priority

no ordering needs to be performed for every incoming packet

per-packet overhead is logarithmic only in the number of priority levels, not the number packets

actively drop packets that have missed their deadlines (packets being useless)
MAC-Layer Prioritization

Local prioritization at each node not enough
MAC protocols should provide distributed prioritization on packets

Initial wait time after idle

802.11 sets a Distributed Inter frame space (DIFS) counter once the communication channel has become idle
Before sending an RTS (Request To Send) packet, a node will wait a random period of time between 0 and DIFS

Backoff window

Doubles its backoff window, CW, to extend a node’s waiting period when a transmission collision occurs

Acquire Channel

Time

Idle

Avoidance

Exponential Backoff

Contention

Contention window (CW)

Transmission
To prioritize this, set the DIFS parameter based on the packet priority:

\[ DIFS = BASE_{DIFS} \times \text{PRIORITY} \]

Packets priority

higher — 1

2

lower — 3

modified 802.11 to increase CW in accordance with the packet priority

\[ CW = \frac{CW*(2+(\text{PRIORITY}-1))}{\text{MAX\_PRIORITY} \text{ value}} \]

MAX\_PRIORITY is the maximum value of priority (corresponding to the lowest priority)

The backoff counter of a node with a pending lower priority packet increases faster than a node with a pending packet with a higher priority.
**RAP**

**MAC-Layer Prioritization**

*Give high priority packets high probability to get the channel in both the contention avoidance and contention phases*

**Scenario 1:**
*Initial wait time after idle*

Packet A with priority levels

- **Highest** 1
- **Lowest** 3

\[ DIFS = \text{BASE}_DIFS \times \text{PRIORITY} \]

- **Highest**: \( 50 \times 1 = 50 \)
- **Lowest**: \( 50 \times 3 = 150 \)

*Packets with higher priority (lower value) choose a smaller waiting period*

*Will be able to access the channel much quicker*

**Scenario 2:**
*Backoff increase window*

Packet A with priority levels

\[ CW = \frac{CW \times (2 + (\text{PRIORITY} - 1))}{\text{MAX_PRIORITY}} \]

- **Highest**: \( CW = \frac{50 \times (2 + (1 - 1))}{3} = \frac{50}{3} = 33.33 \)
- **Lowest**: \( CW = \frac{50 \times (2 + (3 - 1))}{3} = \frac{66.66}{3} = 22.22 \)

*Packets with higher priority will be able to access the channel much quicker*
Experimental Work

Simulation in GloMoSim

*Bio Metric Sensing*

Network Configuration / Workload

- **Square region**: 136×136 m²
- **Radio communication radius**: 30.5 m
- **Packet size**: 32 B (count) - 160 B (detail)
- **Bandwidth**: 200 kbps
- **MICA motes**

Users can register for *events* & query Bio sensors

Periodic query *count* on 31 nodes (longer deadlines than count)

*PRI 3 detail* on 15 nodes

**Combination of protocols / Scheduling**

<table>
<thead>
<tr>
<th>Combinations</th>
<th>Routing</th>
<th>Scheduling</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSR/FCFS</td>
<td>DSR</td>
<td>FCFS</td>
<td>802.11</td>
</tr>
<tr>
<td>GF/FCFS</td>
<td>GF</td>
<td>FCFS</td>
<td>802.11</td>
</tr>
<tr>
<td>GF/DS</td>
<td>GF</td>
<td>DS</td>
<td>802.11 Extension</td>
</tr>
<tr>
<td>GF/SVM</td>
<td>GF</td>
<td>SVM</td>
<td>802.11 Extension</td>
</tr>
<tr>
<td>GF/DVM</td>
<td>GF</td>
<td>DVM</td>
<td>802.11 Extension</td>
</tr>
</tbody>
</table>

*DSR*= Dynamic Source Routing,

*FCFS*= First come First serve

*DS*= Deadline based Scheduling

*PRI*= Priority
Experimental Work

Count:

registerEvent {
  virusFound(0,0,136,136),
}

query {
  virus.count,
  area=(Xevent-1,Yevent-1,Xevent+1,Yevent+1),
  period=Pc, deadline=Dc
  base = (134.07, 128.06)
};

Detail:

query {
  detail,
  area=(x-1,y-1,x+1,y+1),
  period=Pd, deadline=Dd
  base=(134.07, 128.06)
};
Experimental Work

Six repeated runs were made
Main performance metric is the deadline miss ratio

hot regions on the same diagonal to the base station
(worst-case congestion situation)
Simulation Results

Overall Deadline miss Ratios with deadlines (5,10)
Simulation Results

Overall Deadline miss Ratios with deadlines (5,10)

From far corner
Simulation Results

Distance Fairness

SVM provides "fairer" service to remote sensors
Critical for scalability of sensor networks
## Conclusion

### Routing Comparison: DSR/FCFS and GF/FCFS

<table>
<thead>
<tr>
<th>DSR/FCFS</th>
<th>GF/FCFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significantly higher miss ratio than GF</td>
<td>GF does not have the overflow problem (delivers packet through a straight line from source to the base station)</td>
</tr>
<tr>
<td>packet drop due to queue overflow (aggressive route-caching)</td>
<td>Packets from different sensors routed through different nodes (source sensors have different directions toward the base station)</td>
</tr>
<tr>
<td>In DSR, each node caches overheard routes; When a node receives a route discovery packet from another node, it checks its route cache and informs the sender of the requested route if it is available in its route cache</td>
<td>In hot regions only first needs to flood the network</td>
</tr>
<tr>
<td>In overload conditions, nodes on the shared route ran out of queuing space and lost packets</td>
<td>Problems, when correlated traffic patterns e.g. bio-attack</td>
</tr>
</tbody>
</table>

**GF more suitable than DSR**
Conclusion

Comparision
Packet Scheduling Policies

Prioritization-based packet Scheduling: (DS, SVM, and DVM)

FCFS

All velocity-based and deadline-based packet Scheduling significantly outperform FCFS scheduling

SVM / DVM consideration of distance and deadlines, beneficial

Prioritization policies much better approach than FCFS (specially for real-time communication & congested networks e.g. bio-attack)
Conclusion

**Velocity Monotonic Scheduling**
Reduce end-to-end deadline miss ratio
Fair service to remote sensors

**Event/query service API’s**
High-level abstraction for distributed micro sensing

**Location-based protocol stack**
Scalable Small protocol overhead