Addressing Hazards in Wireless Sensor and Actor Networks

WSANs Definition

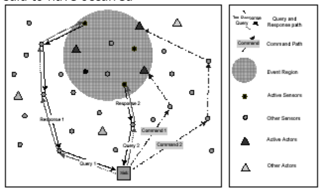
- Wireless Sensor and Actor Networks (WSAN): new class of networks capable of performing both sensing and operating tasks on the environment
  - Consists of sink, sensors and actors
  - Allows automated sensing and execution for a given application
- Typical WSNs perform only one kind of operation: sensing the environment
- WSANs perform both sensing and operating on the environment
- Actors in WSANs typically have higher energy resources, larger transmission and acting range and are fewer in number than sensors

The Problem: Hazards

- Performing both read and write tasks leads to new challenges in WSANs
- One such challenge: Hazards
  - Out-of-order execution of directives from the order that the sink has issued, due to lack of coordination between sensors, actors and the sink
  - May lead to potentially undesirable changes in the environment
  - Example: Consider a fire extinguisher example with sprinklers as actors
    - Command to activate the sprinklers followed by query to check if the fire still exists is the order in which the sink has issued
    - If the order in which the directives arrive at the sensors and actors is reversed, the sink incorrectly issues a duplicate command to activate sprinklers

Illustration of a Hazard

- Consider 2 sequential directives, X and Y, where X reaches a actor in event region via path Command1 and Y reaches a sensor via path Query2
- If the response to Y is generated before X has been executed, a YAX hazard is said to have occurred



Types of Hazards

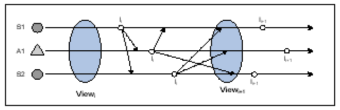
- CAC Hazard
  - If 2 sequential commands, Command1 and Command2 are issued to different actors in the event region, a CAC hazard occurs if Command2 is executed prior to Command1
  - Example: Fire extinguisher system with both sprinklers and fans as actors, where initially Command1 is sent to sprinklers and Command2 is sent to fans. If order of execution is reversed, the fan may cause the fire to spread to other areas before the sprinklers are activated
- QAC Hazard
  - Query-After-Command hazard occurs when a query issued after a command is executed prior to the command
  - Example: In the same fire extinguisher example, let Command1 be sent to fans followed by Query2 to humidity sensors. If Query2 is executed prior to Command1, the sink may incorrectly determine that the moisture level is still high and activate the fans again
- CAQ hazard
  - When a command issued after a query is executed prior to it, a Command-After-Query is said to have occurred
  - Example: Fire extinguisher system with sprinkler actors and heat sensors to detect fire, where Query1 is sent to detect the presence of fire and Command2 is sent to douse the fire. If the response for query1 reaches after Command2 was sent, the sink will have no idea whether there is still fire or not

Observations on Hazards

- Observations for hazard free operation
  - *Any pair of dependent directives issued to entities that do not have any overlapping execution regions can be executed concurrently across the two entities, although the relative ordering must be preserved within each entity*
  - *Any pair of dependent directives issued to entities with overlapping execution regions needs to be ordered in the union of the two regions*
- Dependency region: Based on the above two observations, for each entity there is a region surrounding it, where hazard free operation is required
  - Dependency region for sensor = circle of radius (sensing range + acting range)
  - Dependency region for actor = circle of radius (2 x acting range)

Design of the NC Approach

- NC introduces the notion of a *neighborhood clock* on every sensor and actor for ordering the directives within every *dependency* region
- Sink creates a unique reference clock initially and sensors and actors initialize their clocks to this value
- Each entity, Dx, maintains its own *view* of the progress in the network, based on its neighborhood clock identier, NC(x), where the view number is set to be NC(x) + 1
- Each sensor and actor will move to the next view only after all other sensors and actors have moved into its current view

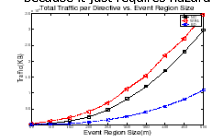


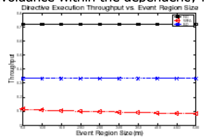
Competing Approaches

- Bounded Delay (BD) approach:
  - Sink waits for a specified amount of time after issuing a query or command
  - After issuing a query, sink waits for a time T_{WS} corresponding to a fixed specified time to receive all responses from sensors
  - After issuing a command, sink waits for a time T_{WA} corresponding to an estimate fixed time before which the command is executed
- Wait-For-All (WFA) approach:
  - Sink issues next directive only after it receives all responses or notifications of execution of that directive from all entities in the event region
  - If the sink sent a command, it will wait for all notifications about the completion of the command from all actors before it issues another query or command

Performance Evaluation

- Total traffic and directives execution throughput of Bounded Delay (BD), Wait For All (WFA) and the NC Approach
  - Proposed approach has lesser traffic overhead WFA but higher traffic than BD, which does not require any overhead as it is just based on waiting time. However, BD does not guarantee 100% correctness and hazard probability increases with event region size
  - Proposed approach has the maximum directives execution throughput because it just requires hazard avoidance within the dependency region





Conclusions and Future Work

- Conclusions
  - Identified the different types of hazards in WSANs with example
  - Identified the design principle that needs to be exploited in designing an efficient approach to address hazards
  - Proposed a distributed approach that completely addresses the hazards
- Future Work
  - To determine if QAQ is a hazard or not in WSANs
  - To extend the approach to address any related challenges including network and event area dynamics
- http://www.ece.gatech.edu/research/GNAN

Ramanuja Vedantham, Zhenyun Zhuang, Raghupathy Sivakumar, GNAN Research Group, Georgia Tech