Optimizing Sensor Data Acquisition for Energy-Efficient Smartphone-based Continuous Event Processing

Lim Lip Yeow
University of Hawai`i at Mānoa

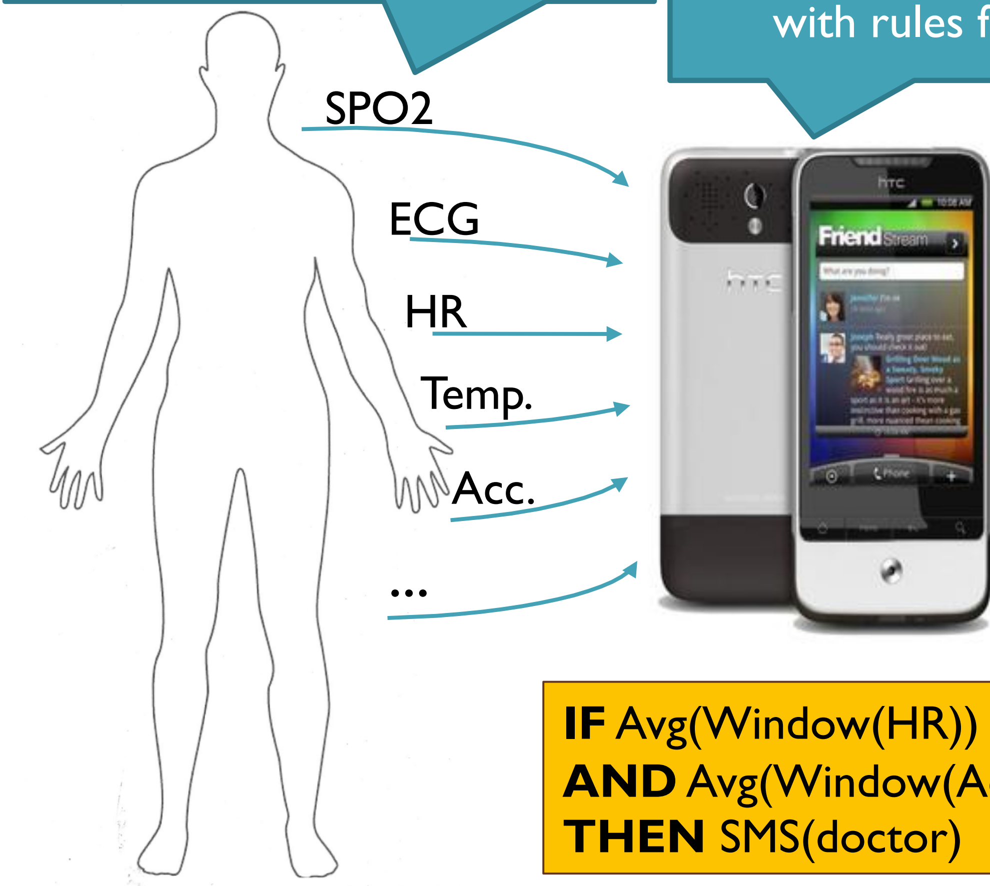
Archan Misra
Singapore Management University

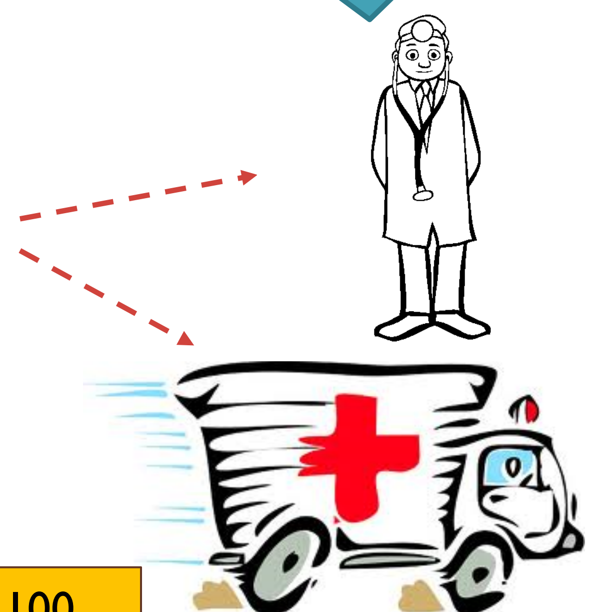
Telehealth Scenario

Wearable sensors transmit vitals to cell phone via wireless (eg. bluetooth)

Phone runs a complex event processing (CEP) engine with rules for alerts

Alerts can notify emergency services or caregiver

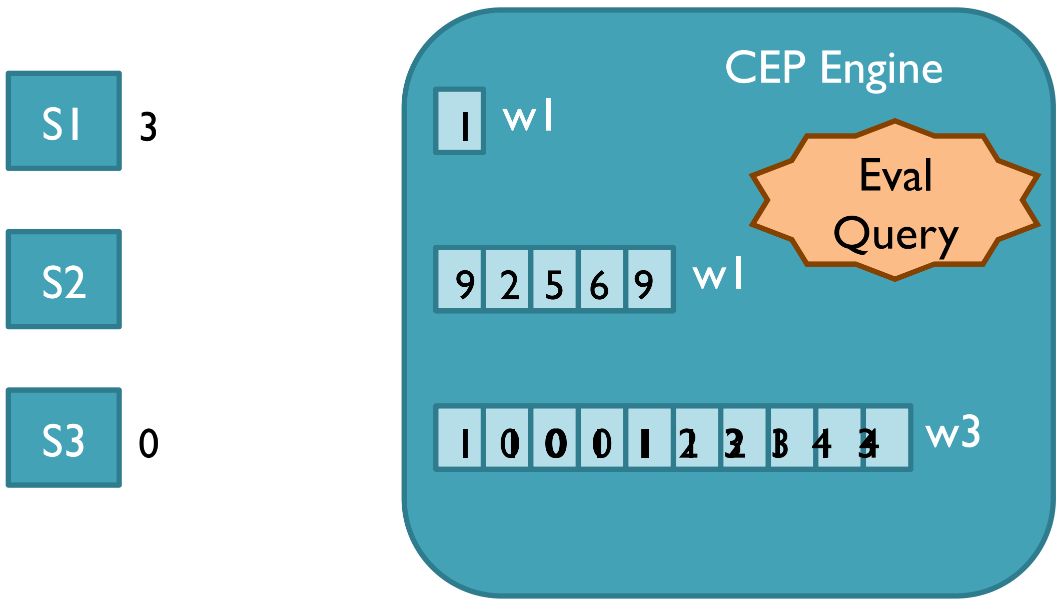




IF Avg(Window(HR)) > 100
AND Avg(Window(Acc)) < 2
THEN SMS(doctor)

Continuous/Streaming Evaluation

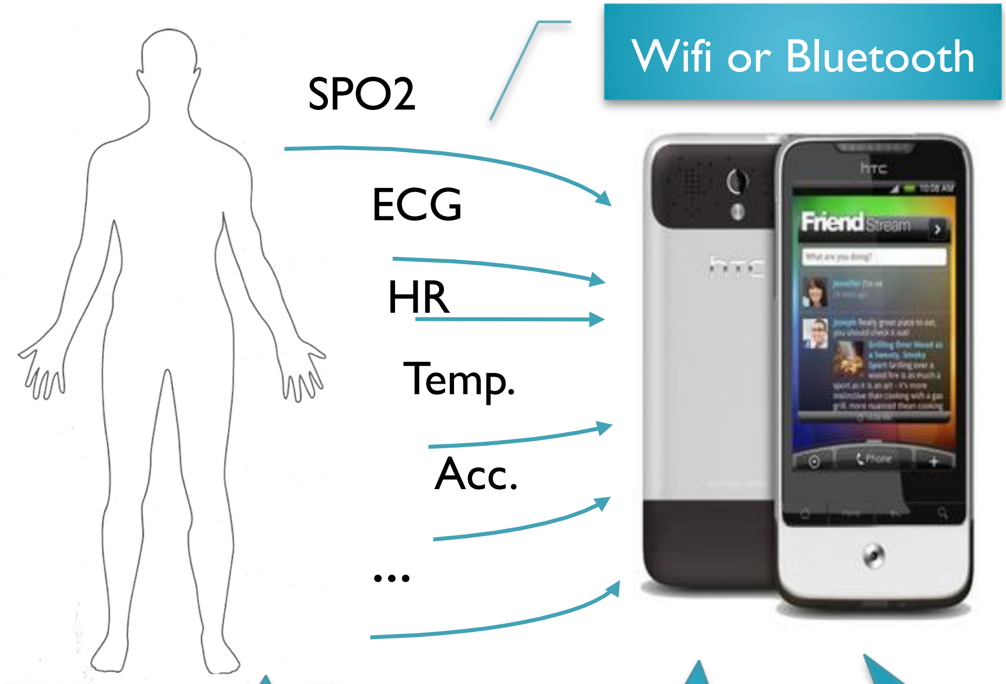
if Avg(S2, 5)>20 AND S1<10 AND Max(S3,10)<4 then email(doctor).



Algorithm

When t_i of Si arrives
  Enqueue t_i into W_i
  If Q is true,
  Then output alert

Energy Consumption

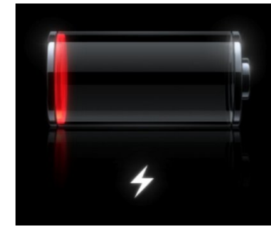


Sensor:
transmits data

Phone:
receives data

Phone:
evaluate queries

Research Question



Is there a better way to perform such complex event processing that

- Minimizes energy consumption at the phone, and/or
- Maximizes operational lifetime of the system.

Key Ideas

- Pull model
  - Evaluate a query every ω seconds
  - Acquire only data that is needed
- Evaluation order of predicates matter!
  - Shortcircuiting can avoid data acquisition
- Batching

Assuming fairly smart sensors capable of buffering and supporting "pull"

Sensor Data Acquisition

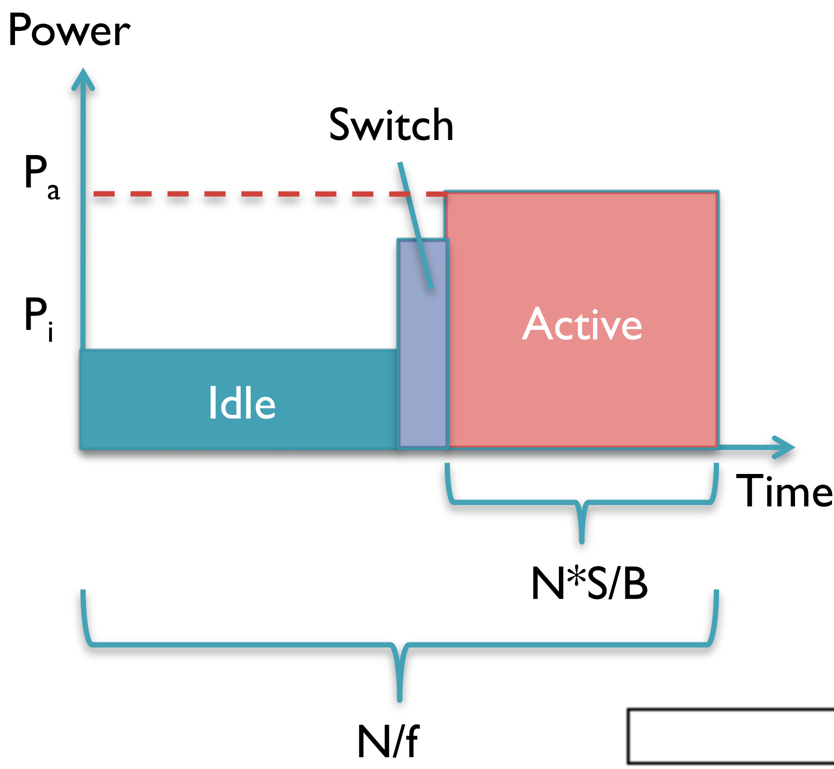
3D acc.
ECG,
EMG, GSR

Bluetooth
Or 802.11
Or 802.15

- Constant sampling rate
- 802.11 (wifi) uses 2 power modes: active, idle
- Bluetooth has 3 modes: active, idle, sleep (not relevant).
- Time needed to switch modes
- Energy expended to switch

Sensor Type	Bits/ sensor channel	Channels/ device	Typical sampling frequency (Hz)
GPS	1408	1	1 Hz
SpO2	3000	1	3 Hz
ECG (cardiac)	12	6	256 Hz
Accelero-meter	64	3	100 Hz
Temperature	20	1	256 Hz

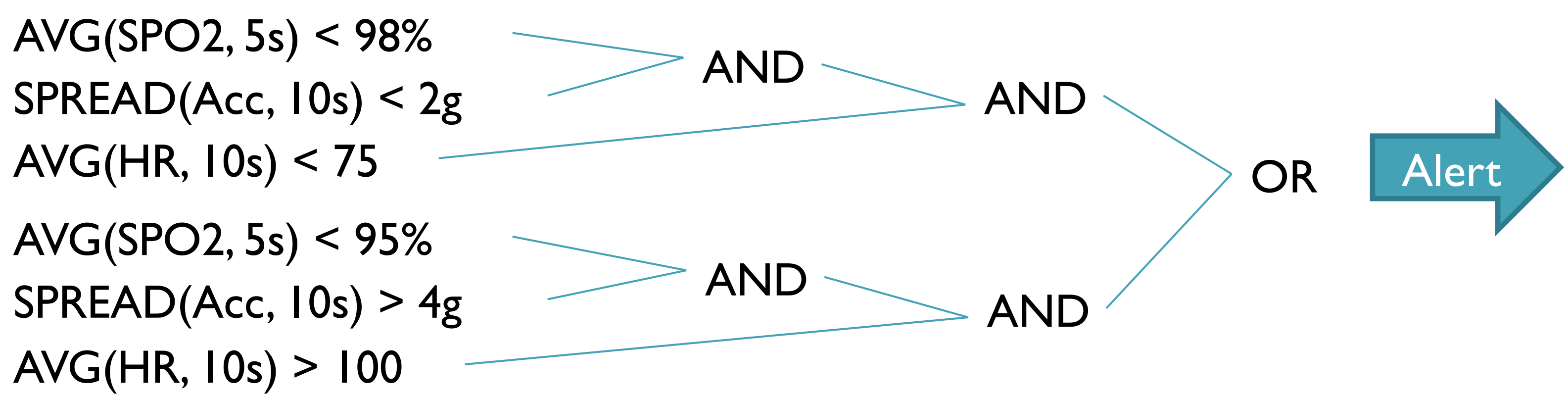
Pulling N Tuples from Sensor



- Idle mode consumes P_i mW
- Active mode consumes P_a mW
- Sensor rate is f Hz
- A tuple is S bits
- Bandwidth is B Mbps

	IEEE 802.11	Bluetooth 2.0+EDR
P_a	947 mW	60mW
P_i	231 mW	5 mW
B	54 Mbps	1 Mbps
E_{switch}	14 μJoule	–
Th_{idle}	100 ms	–
T_{switch}	–	6 msec

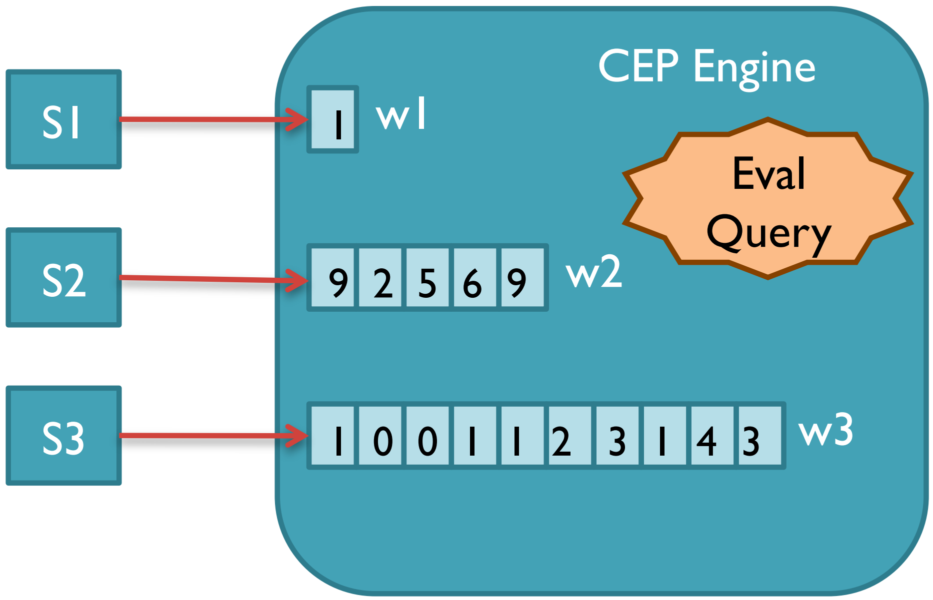
Query Model



- A query is a boolean combination of predicates
- Predicates
  - Aggregation functions over a time-based window of sensor data
- Traditional push model
  - A given query is evaluated whenever a new sensor reading arrives

Pull-based Evaluation

if Avg(S2, 5)>20 AND S1<10 AND Max(S3,10)<4 then email(doctor).

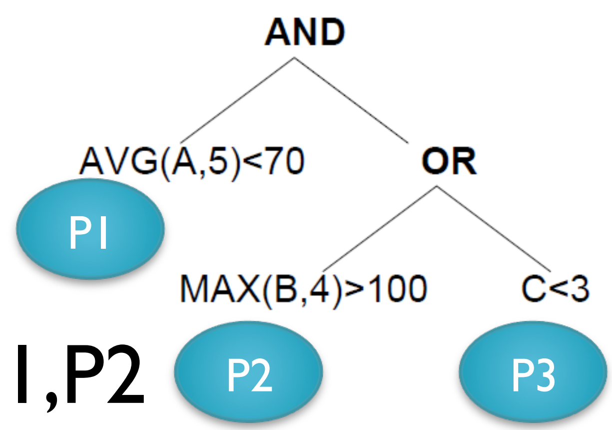


Pull

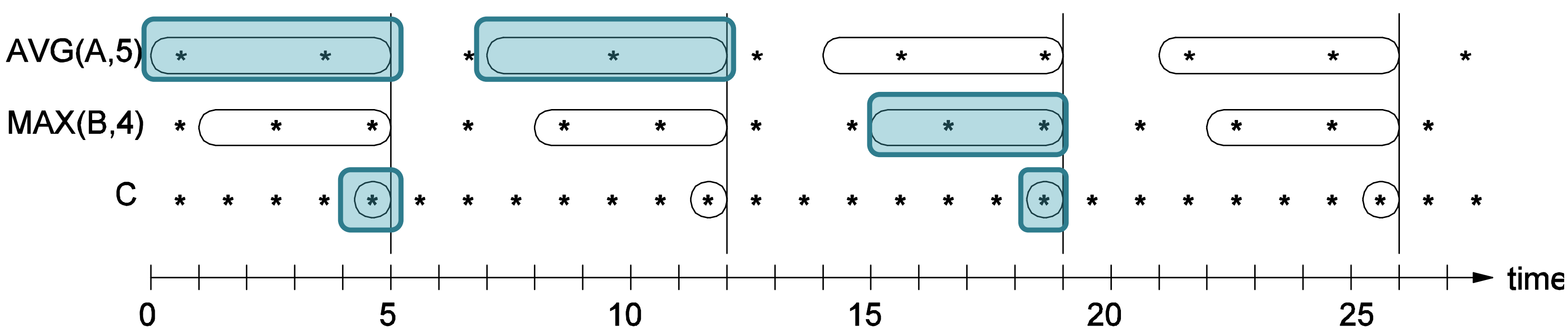
Loop every ω seconds
  For each sensor Si
    Acquire data for Si
    Enqueue data into W_i
  EndFor
  If Q is true,
  Then output alert
End loop

- Complex interaction between ω, stream rates, and predicate windows
- If predicate S1<10 is false, why bother to acquire data for S2 nand S3?

Example: ω=7



- Time 5: eval order is P3,P1,P2
- Time 12: eval order is P1,P2,P3
- Time 19: eval order is P2,P3,P1



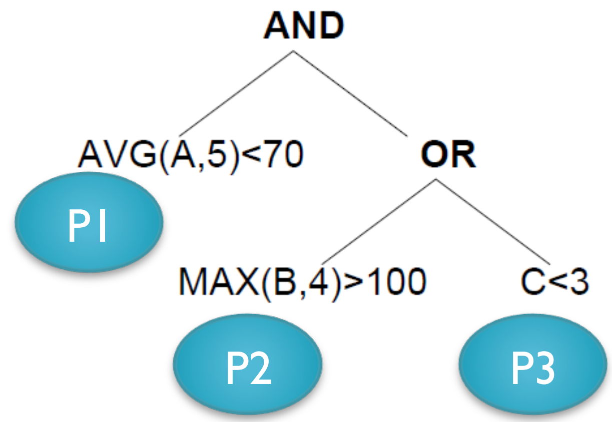
Evaluation Order

if Avg(S2, 5)>20 AND S1<10 AND Max(S3,10)<4 then email(doctor).

Predicate	Avg(S2, 5)>20	S1<10	Max(S3,10)<4
Acquisition	5 * .02 = 0.1 nJ	0.2 nJ	10 * .01 = 0.1 nJ
Pr(false)	0.95	0.5	0.8
Acq./Pr(f)	0.1/0.95	0.2/0.5	0.1/0.8

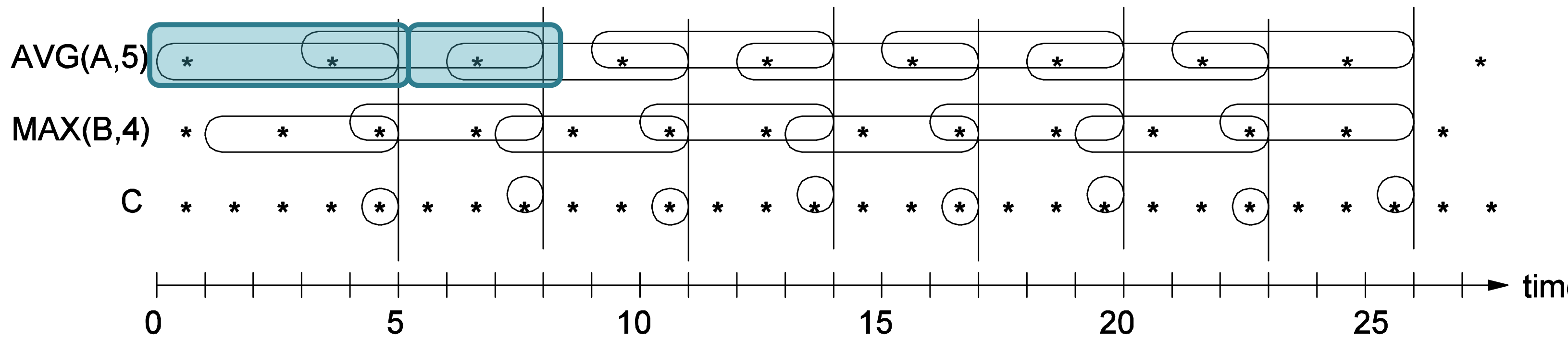
- Evaluate predicates with lowest energy consumption first
- Evaluate predicates with highest false probability first
- Hence, evaluate predicate with lowest normalized acquisition cost first.

Example: ω=3



- Time 5: P1,P2,P3
- Time 8: acquisition cost for A becomes cheaper, because some tuples are already in buffer

Acquisition cost depends on state of the buffer at time t



Algorithm Sketch

At each ω

1. Calculate normalized acquisition cost (NAC) based on buffer state and P(pred=true)
2. Find evaluation order using NAC
3. Acquire sensor data and eval pred using eval order with shortcircuiting.

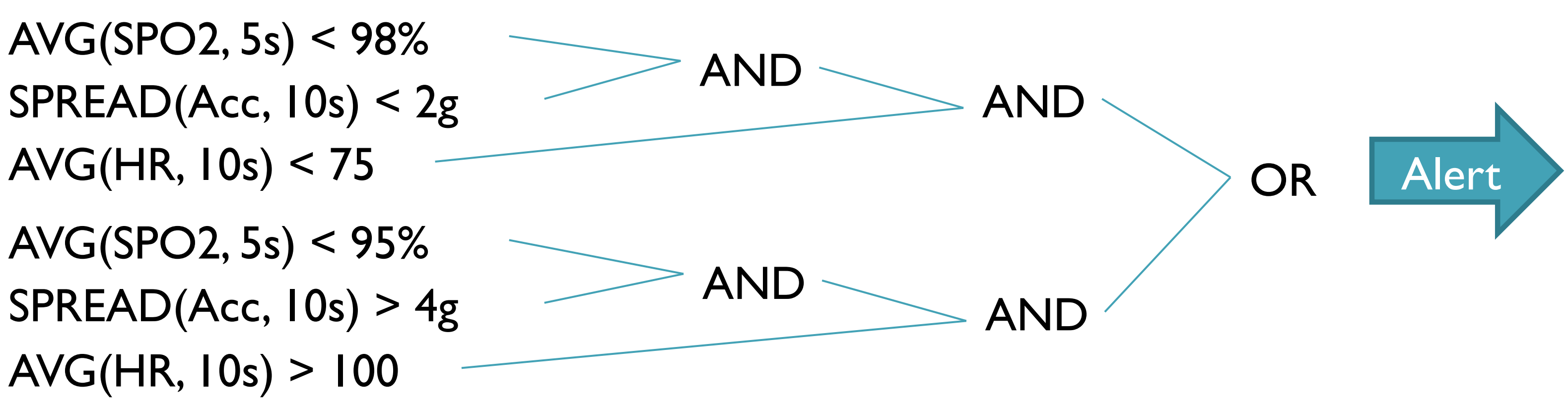
What happens if >2 predicates operate on the same sensor data stream?

Simulation Setup

- Naive
  - data from all sensors acquired in batches
- ASRS-static
  - Evaluation order determined once at initialization and never changes
- ASRS-dynamic
  - Evaluation order determined at each ω time period.
- Simulation duration is 1 hour
- P(pred=true) distributions obtained from half the data streams themselves

Simulation Data & Query

- Data streams generated using independent Gaussian distribution
  - SPO2 ~ N(96,4), 3 Hz, 3000 bits
  - HR ~ N(80,40), 0.5 Hz, 32 bits
  - Accel ~ N(0,10), 256 Hz, 196 bits

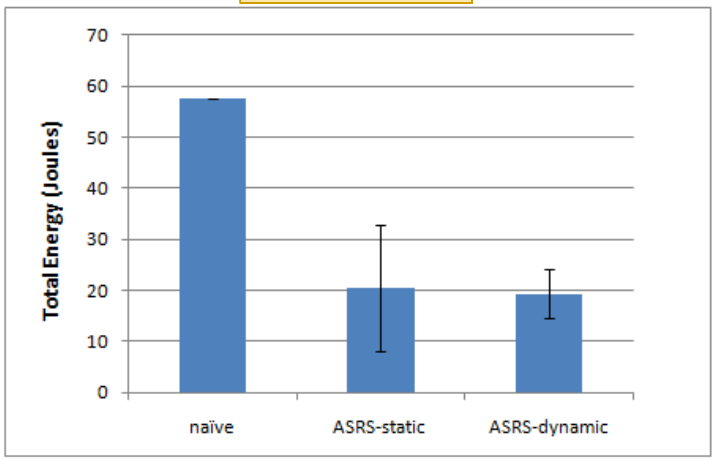


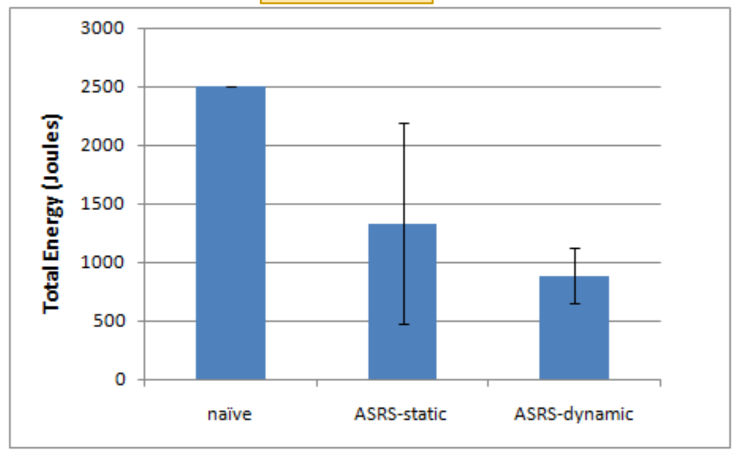
Simulation Results

Bluetooth

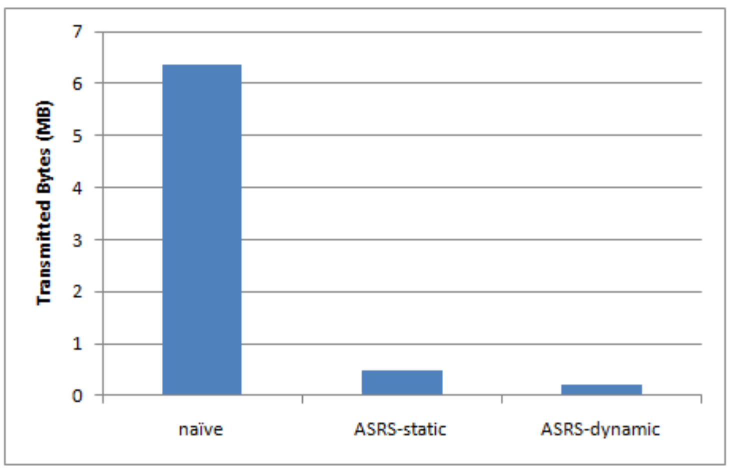
802.11

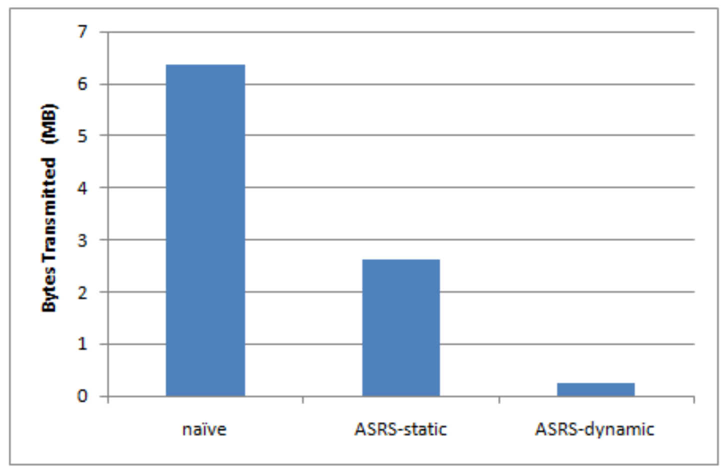
Energy





Bytes





Conclusion

- Pull-based processing paradigm can have a significant impact on data acquisition energy consumption
- Ordered evaluation of predicates can help shortcircuit the evaluation and avoid costly data acquisition
- We proposed evaluation algorithms based on these two observations to minimize data acquisition cost at CEP engine
- Results on synthetic traces show that savings up to 70% are possible.

Future Work

- Improve simulator
  - Disjunctive normal form query representation
  - More realistic data generators
- Estimation algorithms for P(pred=true)
- Batching: wait say 3 ω before query evaluation
- End-to-end evaluation on Android phone
  - Maximize operational lifetime of phone+sensors