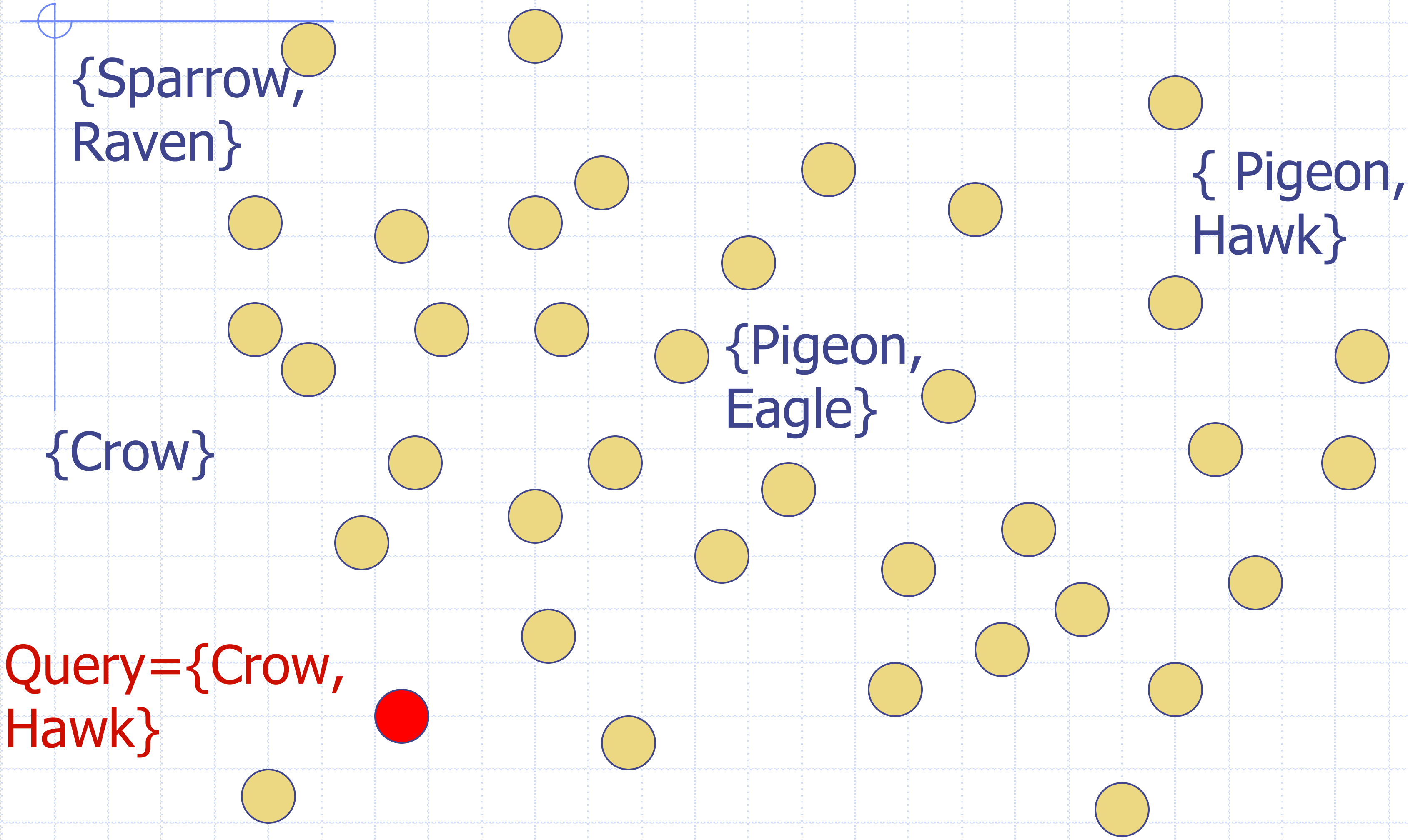
Efficient Querying in Sensor Networks: Flood,Walk,Cache and then ACQUIRE

Presentation for CS428

-Mukund Sundararajan

Sensor Networks As Distributed Databases



Types of Queries in Sensor Networks

- Continuous queries v/s One shot.
  - Report temperature for next 7 days.
  - Is the current temperature >70?
- Aggregate v/s Non-aggregate.
  - Average temperature of region.
  - What is the temperature measured by node X?
- Complex v/s Simple.
  - What are the values of vars A,B,C?
  - What is the value of A.
- Replicated v/s Unique.
  - Is there atleast one node with temp>70?

Approach #1: Flood

- Query Processing in two phases
  - Sink floods several copies of the query
  - Relevant nodes reply with the answer
- Energy efficient if continuous query
  - First phase amortizes over many rounds
- Duplicate responses result in energy losses

Approach #2: Walk

- The query performs a guided or random walk in the n/w.
- Each node partially processes query.
- Query walks home when solved.
- Alternatively events may walk.
- Latency is an issue.

ACQUIRE

- Published as:
  - N. Sadagopan, B. Krishnamachari, A. Helmy, "The *ACQUIRE* Mechanism for Efficient Querying in Sensor Networks", *First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), in conjunction with IEEE ICC 2003*, pp. 149-155, May 2003, Anchorage, AK, USA.
  - ACtive QUery forwardIng in sensoR nEtworks.
  - Simple 7 page paper.
- Contributions.
  - Query processing protocol : ACQUIRE.
  - Modeling and analysis of energy consumption.
  - Comparison with other protocols.

Central Idea

- Study trade-off between Walk and Flood approaches
- Use intelligent caching
- Good for one-shot, complex, replicated non-aggregated queries
  - Obtain sample calls for Blue-jays, Nightingale, Cardinal and Warbler

Data Tracking and Query Model

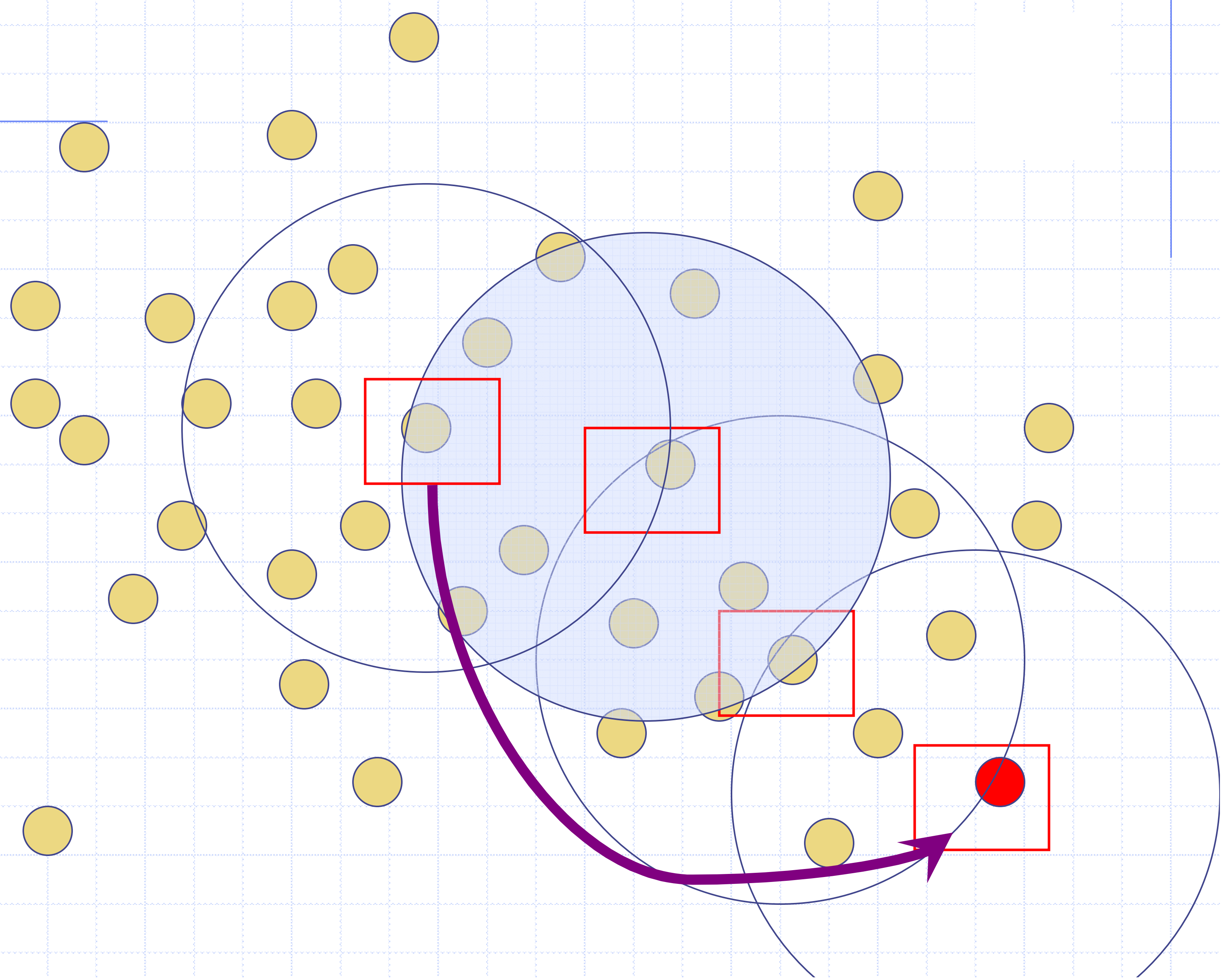
- Let $V=\{V_1, V_2 .. V_n\}$ be n variables.
- Let $Q=\{Q_1, Q_2 .. Q_m\}$ be the query.
- Each node keeps track of one variable with uniform random probability.
- Query is issued at node x*.
- Assumption that there is data replication: N/M nodes have partial answer to a query.

ACQUIRE Preliminaries

- Each active node maintains information about nodes within d hops.
- The number of such nodes is given by the n/w topology dependant function f(d).

ACQUIRE Mechanism

- A random walk is performed starting at x*.
- Current node may refresh data from neighbourhood.
- Query is resolved based on partial information
- The query is then forwarded to a random node at the edge of its neighbourhood



How Much Energy Is Consumed ?

Energy is measured in terms of number of transmissions

Energy Analysis: Notation

- Let $V=\{V_1, V_2 .. V_n\}$ be n variables.
- Let $Q=\{Q_1, Q_2 .. Q_m\}$ be the query.
  - $m<n$, $Q_i \in V$.
- S_m be the average number of steps.
- d is look-around of an active node.
- f(d) be the number of nodes within d hops.
- c is update frequency.

Energy Analysis Equation

- $$E_{avg} = (cE_{update} + d)S_M + \alpha \qquad (1)$$
- alpha is average distance to sink
- Note special cases
  - If d = Diameter then Flooding
  - If d= 0 then Random walk
- S_m reduces as d increases

Energy Analysis: Estimating Expected Number of Steps: S_m

- Suppose each node tracks a single V_i with equal probability
- Estimate number of hops with d=0. (Random Walk)
- Given a query Q, consider a trial which asks if a particular node can satisfy any variable in the query
- Success probability: M/N
- In a random walk, expected time to first success is N/M

Estimating S_m Contd…

$$E(\sigma_M) = N \sum_{i=1}^{M} \frac{1}{M - i + 1} = NH(M) \qquad (2)$$

$$E(\sigma_M) \approx N(\ln M + \gamma) \qquad (3)$$

$$S_M = \frac{E(\sigma_M)}{f(d)} \approx \frac{N(\ln M + \gamma)}{f(d)} \qquad (4)$$

Increasing look-around reduces walk length

Estimating Energy for a Triggered Update

$$E_{update} = \left(f(d-1) + \sum_{i=1}^{d} iN(i)\right) \qquad (5)$$

- N(i) is the number of nodes at distance i.

Total Energy Consumed (in a Grid)

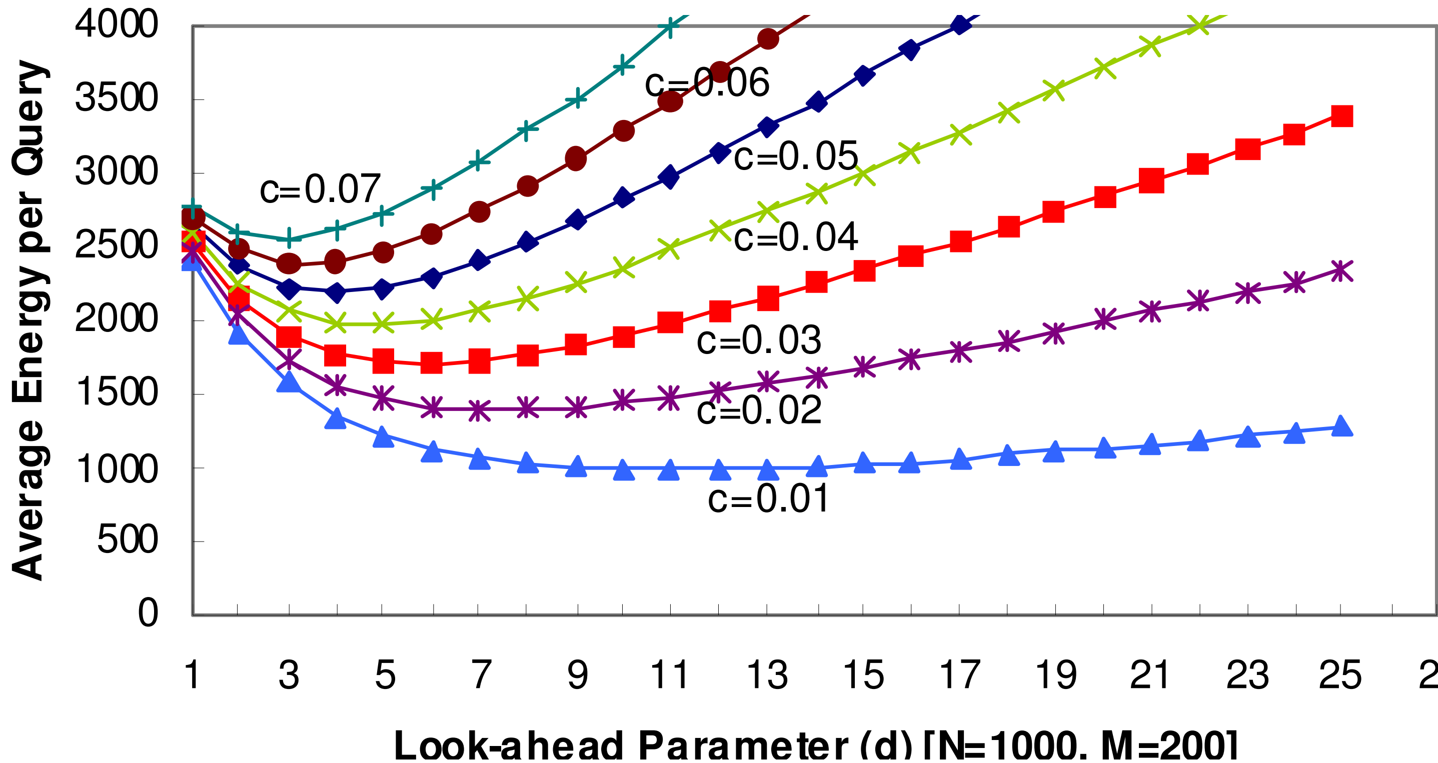
$$E_{avg} \approx \{ \frac{cN(\ln M + \gamma)}{3} \frac{4d^3 + 12d^2 - 4d + 3}{2d^2 + 2d + 1} + N(\ln M + \gamma) \frac{2d}{2d^2 + 2d + 1} \} \quad (9)$$

- N(i) = 4i for a grid (ignoring boundary)
- f(d) = 2d(d+1) +1
- Gamma is Euler constant
- To find the optimal d*, differentiate wrt d. and set to 0.

Energy Consumed.

d* is larger for smaller values of c

Performance of ACQUIRE



Flood Based Approach.

- Flood Query
- All nodes that track the variables respond
- Use the caching idea

$$E_{avg} = (f(R) + \sum_{i=1}^{R} iN_{avg}(i))c$$

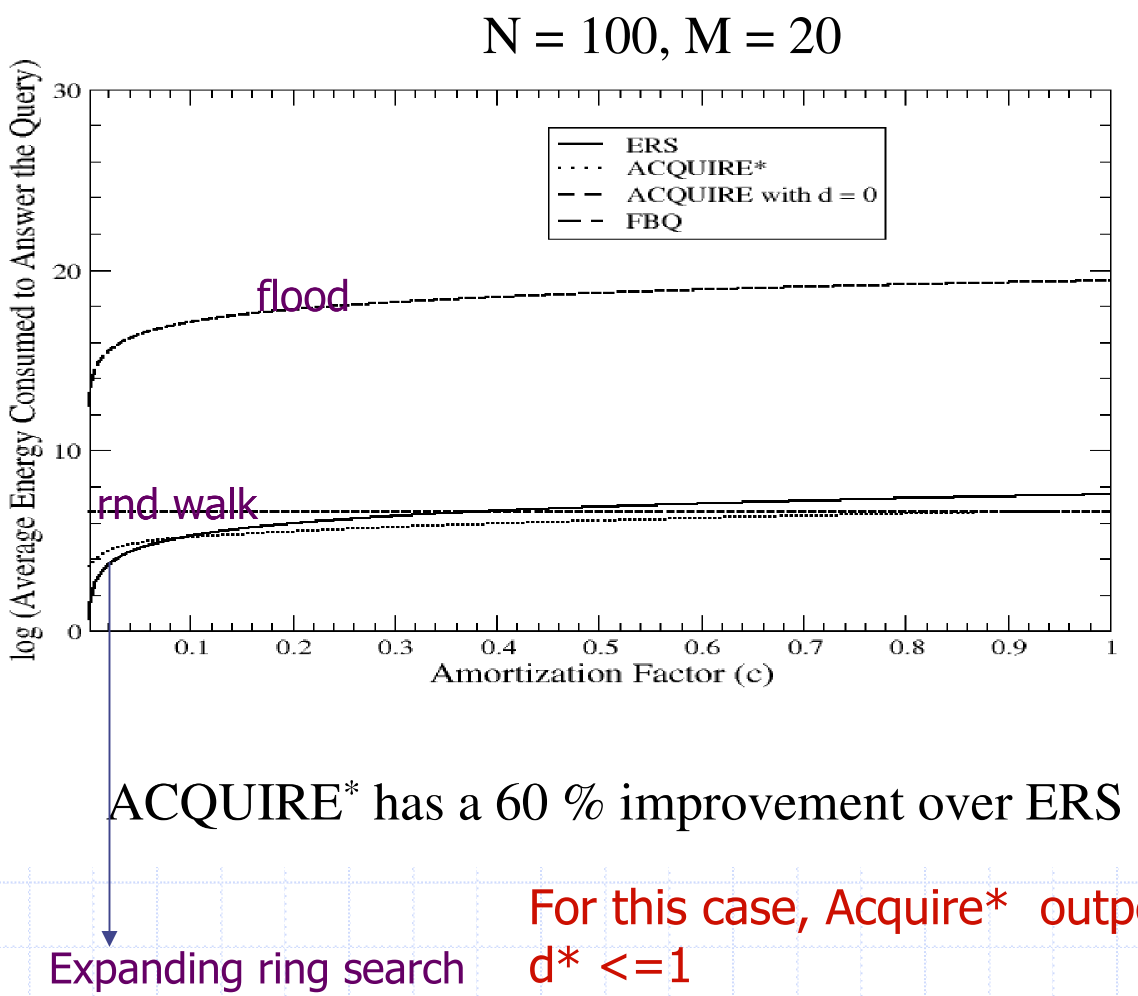
$$= (f(R) + \frac{M}{N}\sum_{i=1}^{R} iN(i))c$$

- E_{avg} proportional to X^3/2
- X is the number of nodes in the n/w.

Expanding Ring Search

- d=0 at Start.
- The query node x* tries to answer query Q using updates from nodes within distance d.
- If query is not satisfied, d is increased by 1.
- Use the caching idea.
- Similar equations, see paper.

Comparison



ACQUIRE* has a 60 % improvement over ERS

For this case, Acquire* outperforms ERS if d* <=1

Notes

- Efficiency can be improved by guiding trajectory
  - Reducing overlap
  - Guiding query towards regions of information
- Find value of the look-around d based on the amortization factor c; this factor is application dependant
- Nodes could take turns being active

Take Home Message

- Query processing in sensor n/w depends on:
  - Nature of query
  - Data Replication impacts efficacy of walks
  - Rate of change of data values impacts efficacy of caching
  - Topology of the network

ONE SIZE DOES NOT FIT ALL!