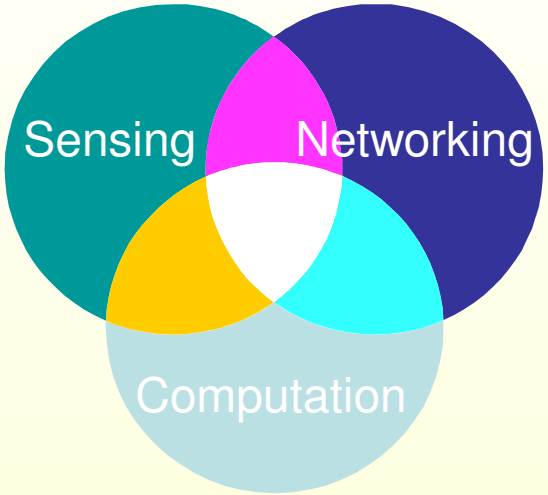
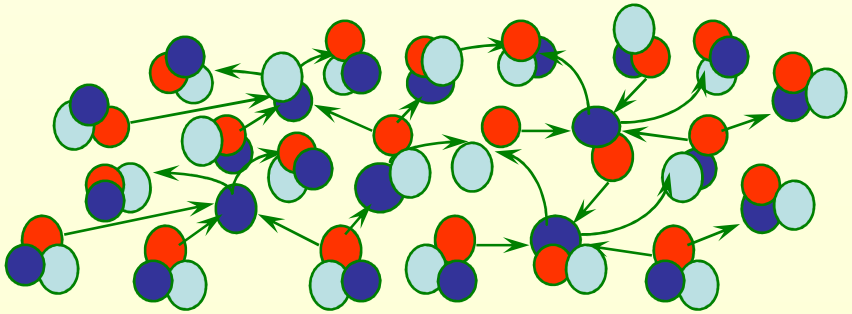
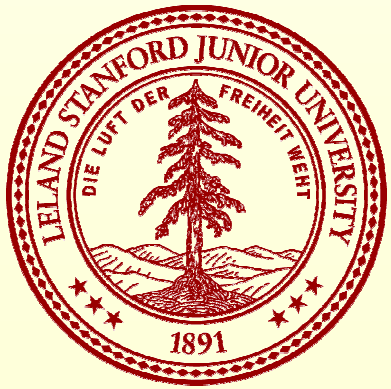
CS321: Synopsis Diffusion and Other Order-and-Duplicate Insensitive Aggregation Schemes



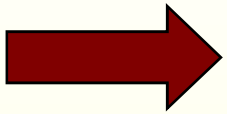
Leonidas Guibas,
Primoz Skraba
Computer Science Dept.
Stanford University





Review

- Query-based sensor networks
  - TinyDB
  - Aggregate queries



Average
Max
Min
Quartile

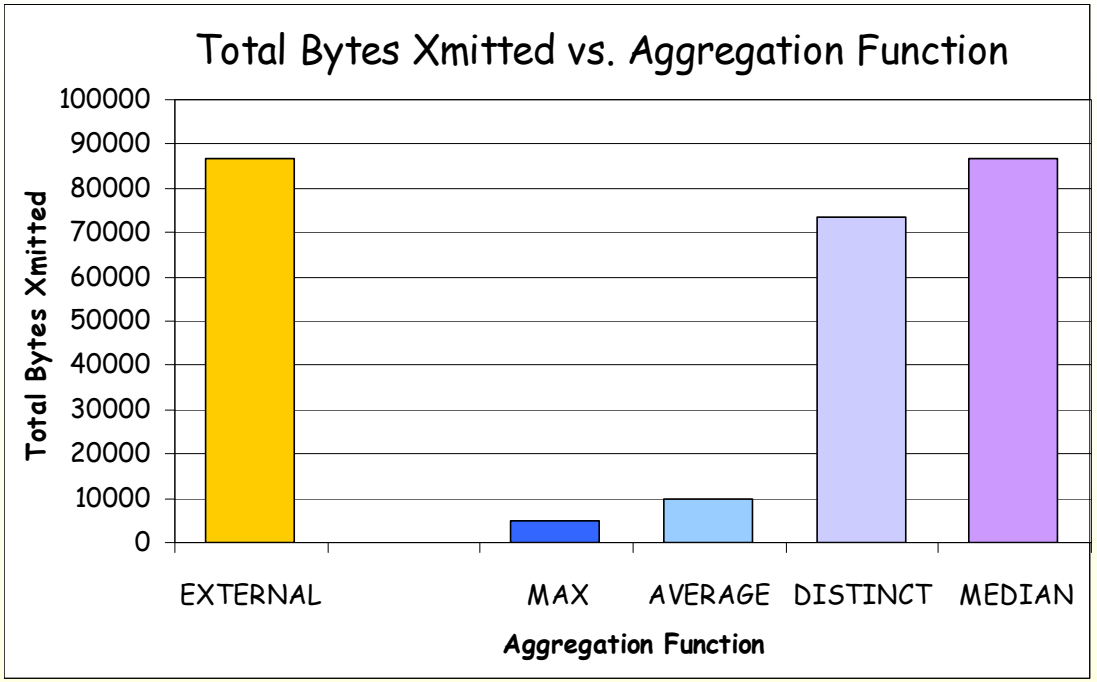
Data Collection

In-network Processing

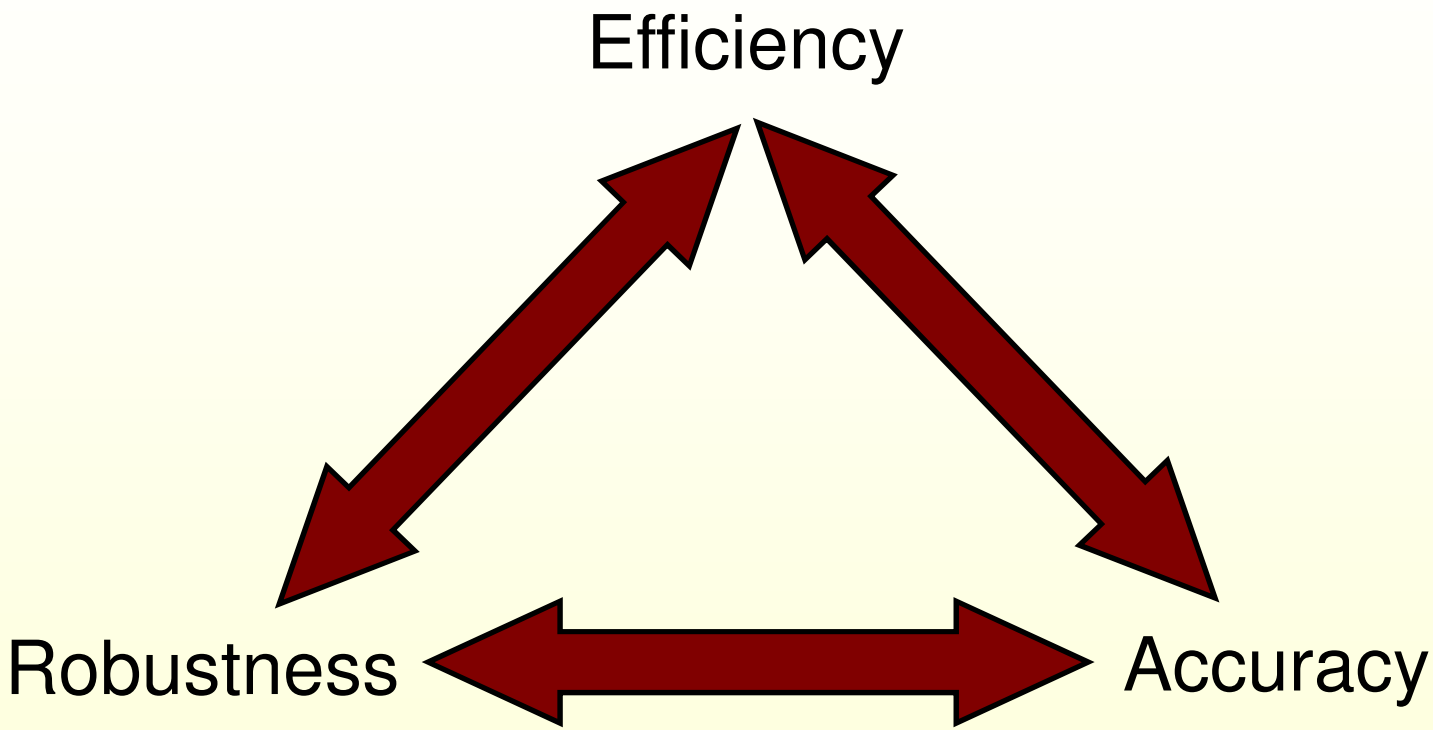
COMMUNICATION COST!

Partial State Record

- Summary of types
  - Algebraic: |PSR| = 1
    (e.g. MIN)
  - Distributive: |PSR| = c
    (e.g. AVG)
  - Holistic: |PSR| = n
    (e.g. MEDIAN)
  - Unique: |PSR| = d
    (e.g. COUNT DISTINCT)
  - Content Sensitive:
    |PSR| < n (e.g. HISTOGRAM)
- Communication Cost
  - Tree-based

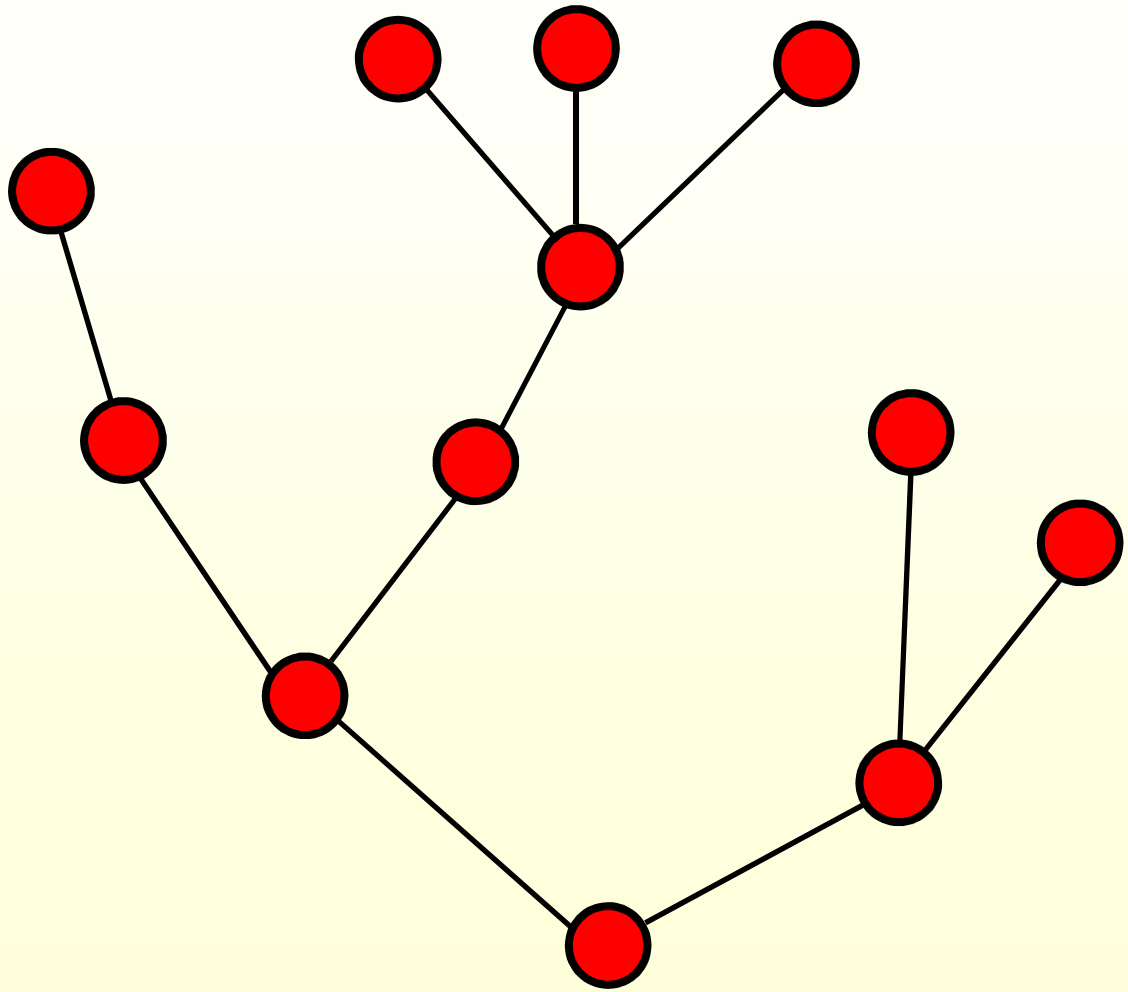


Tradeoff



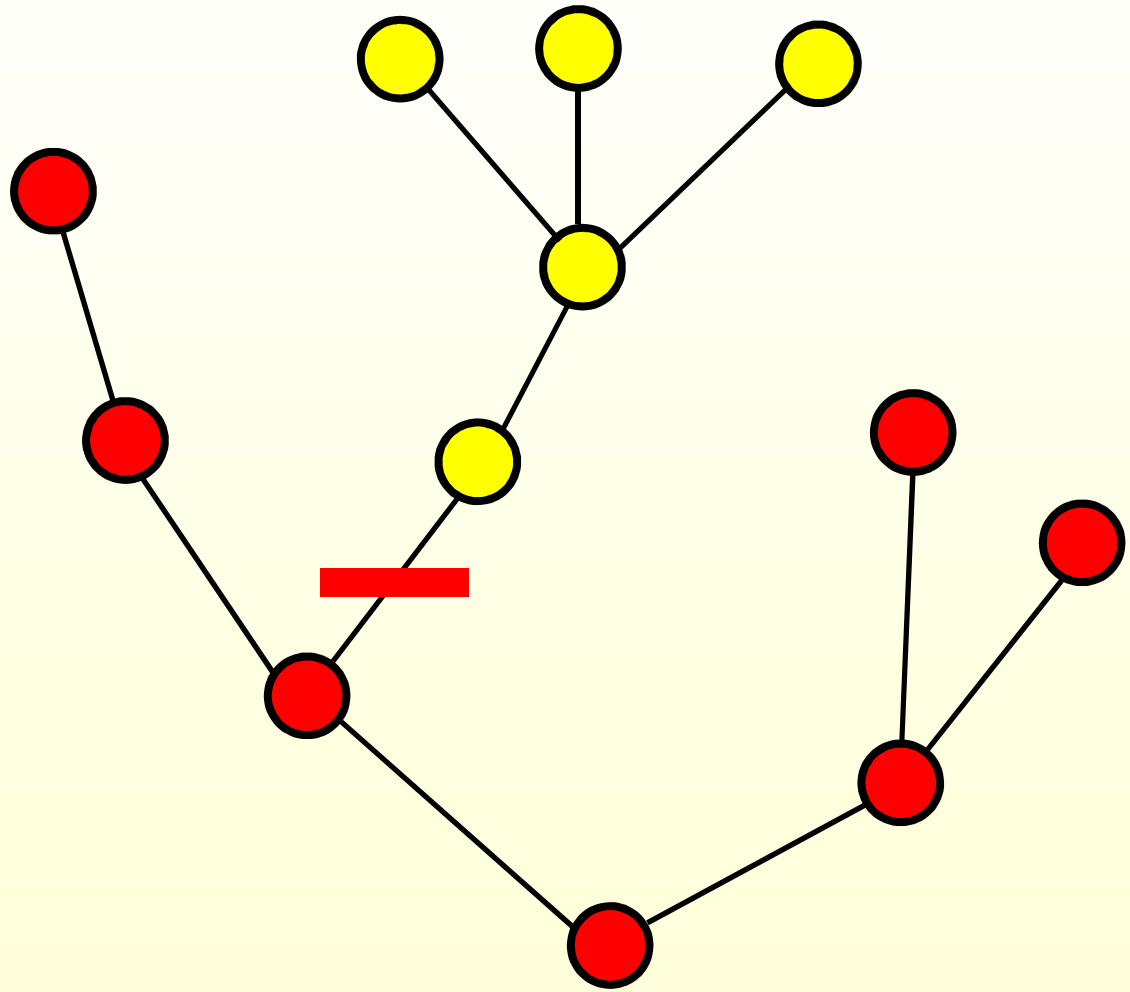
Robustness

- Link failure
- Node failure



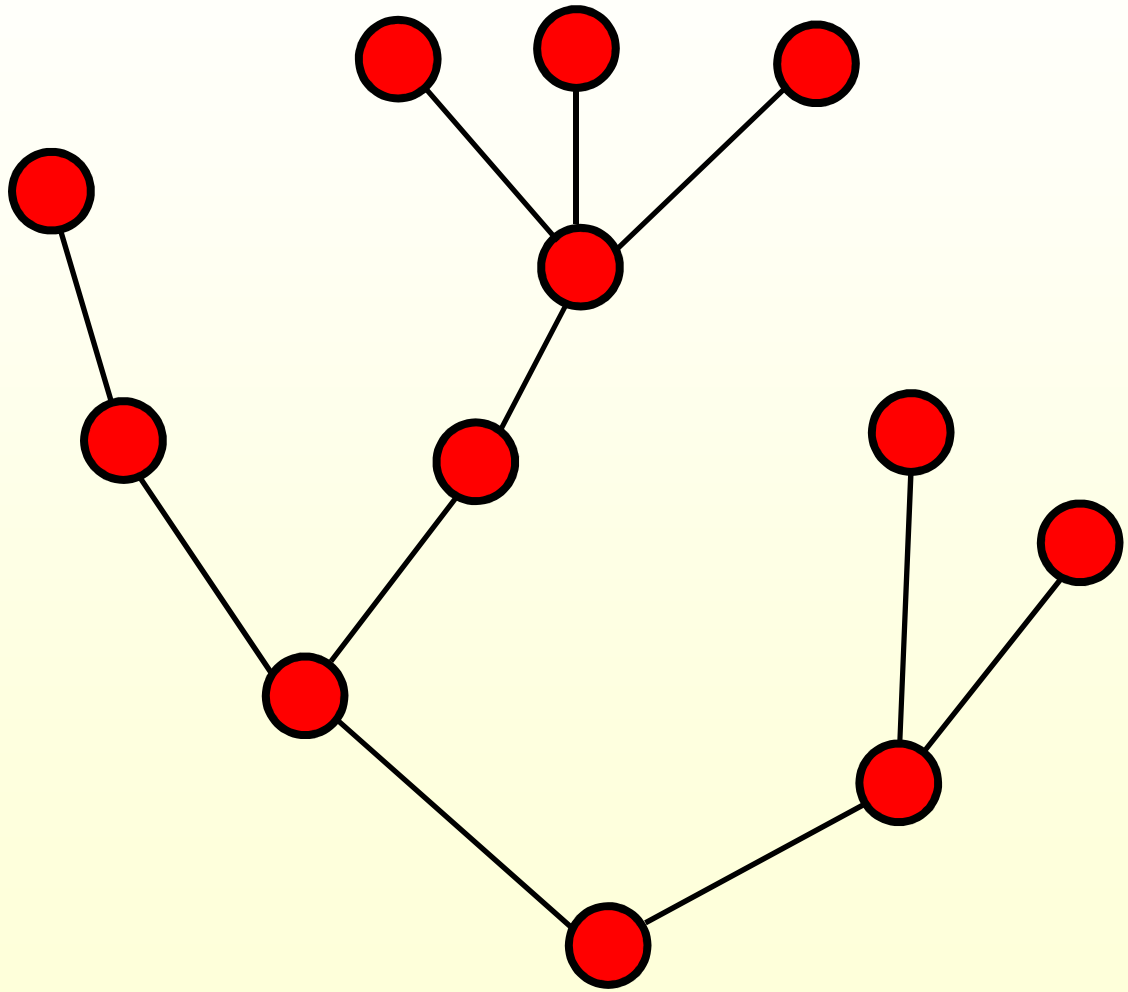
Robustness

- Link failure
- Node failure



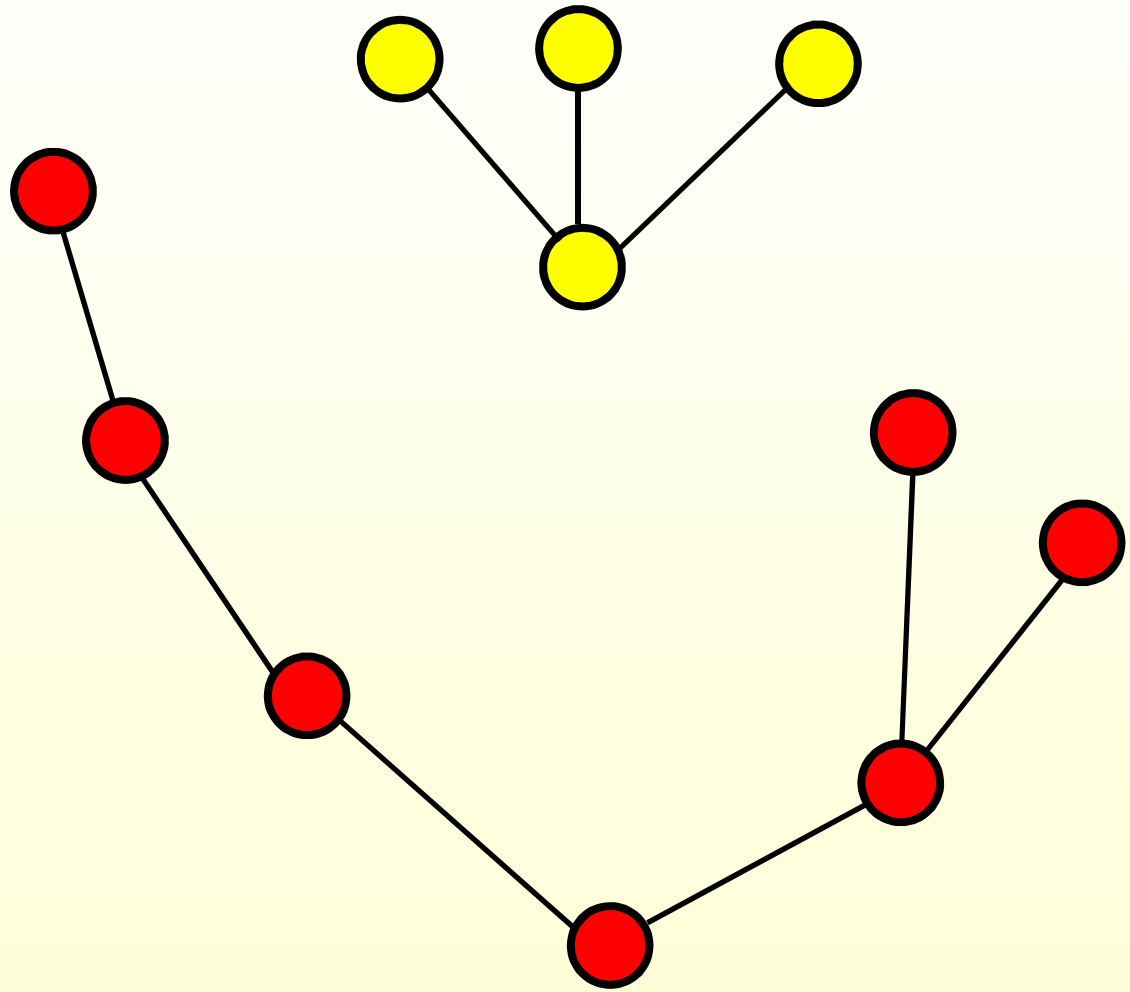
Robustness

- Link failure
- Node failure



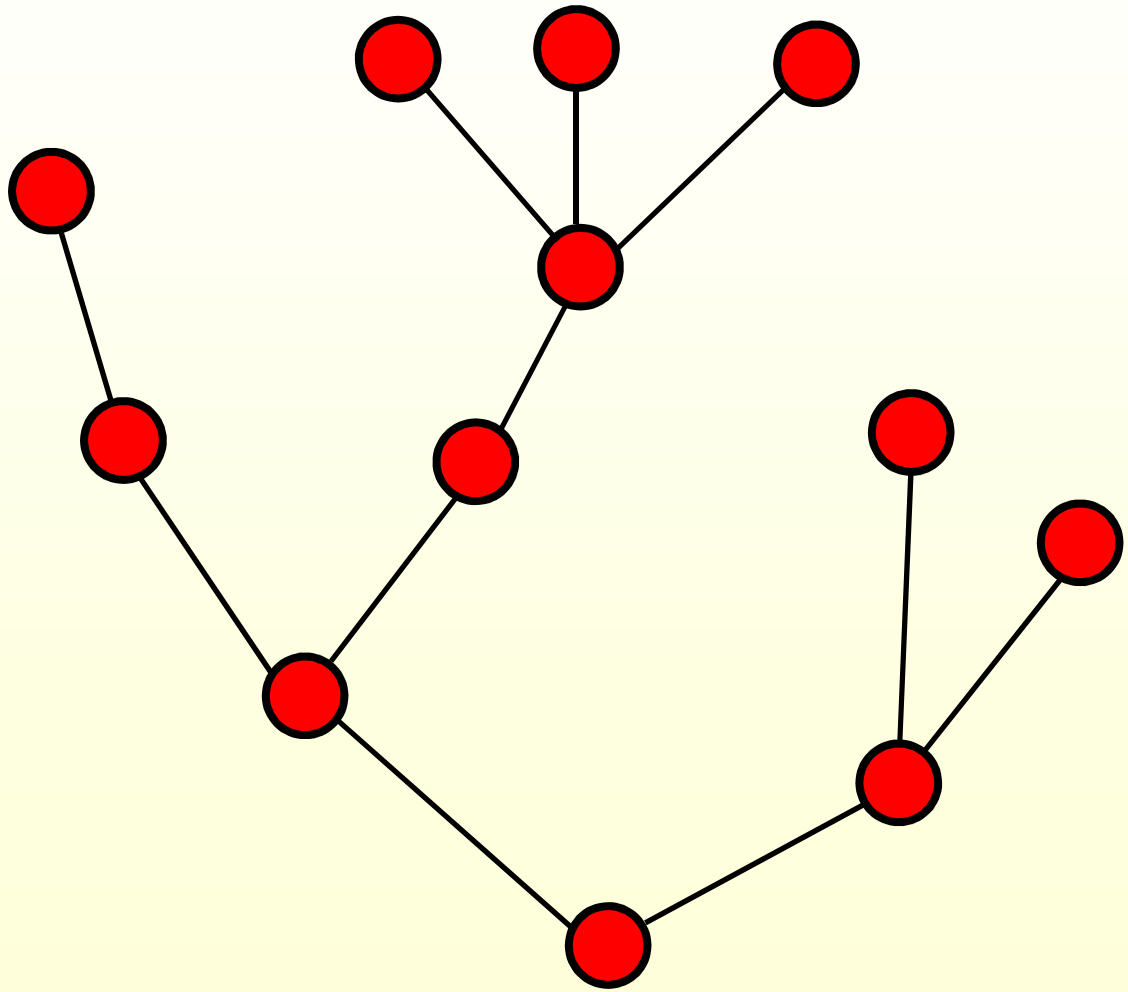
Robustness

- Link failure
- Node failure



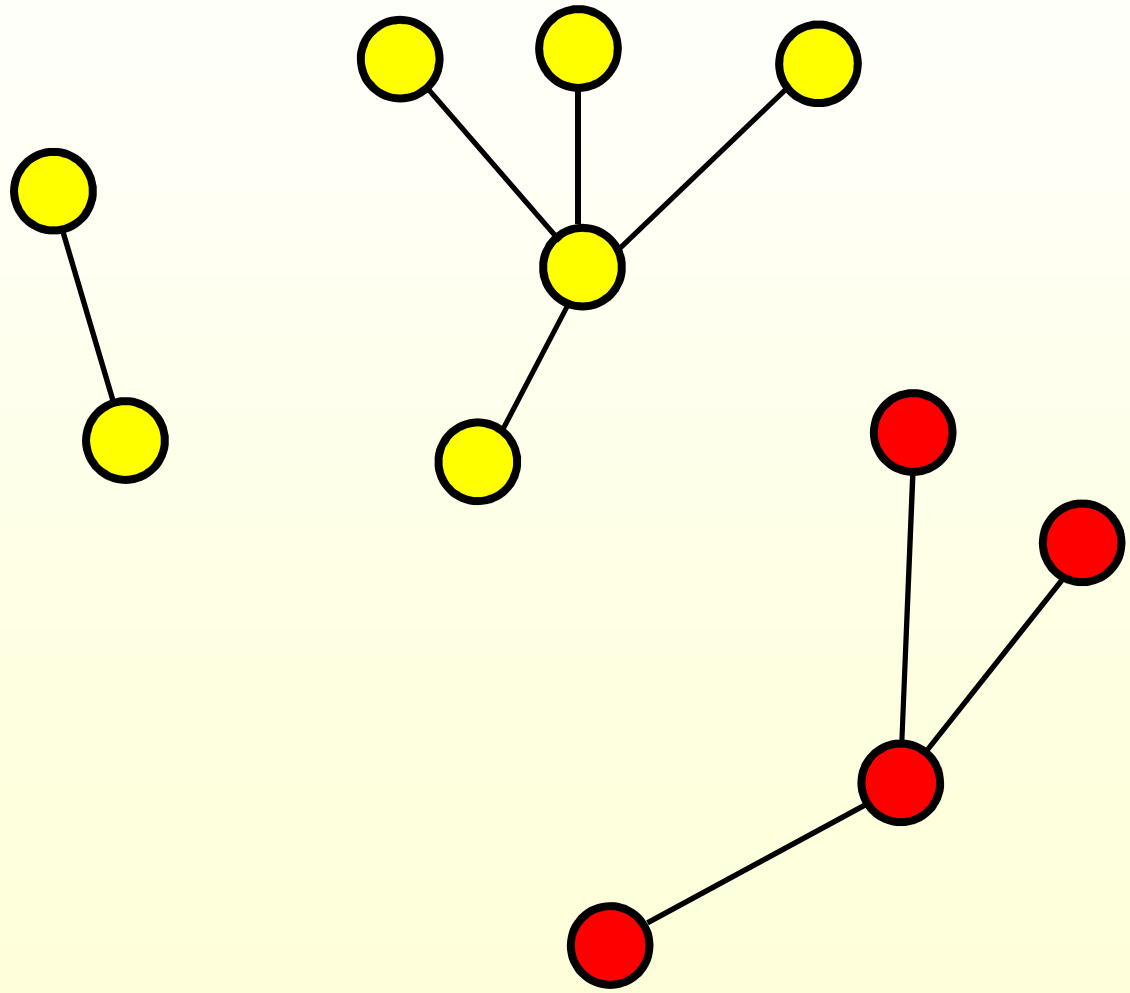
Robustness

- Link failure
- Node failure



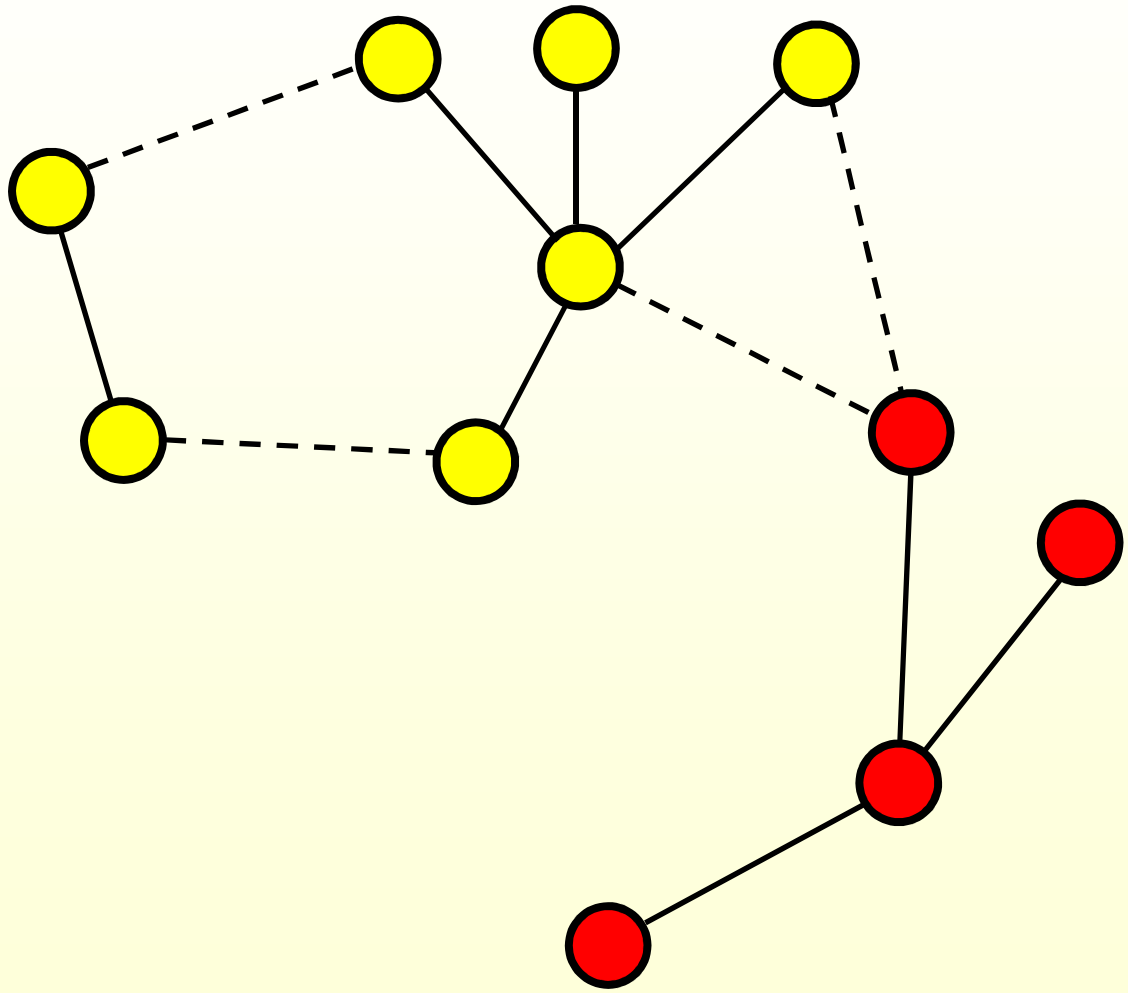
Robustness

- Link failure
- Node failure



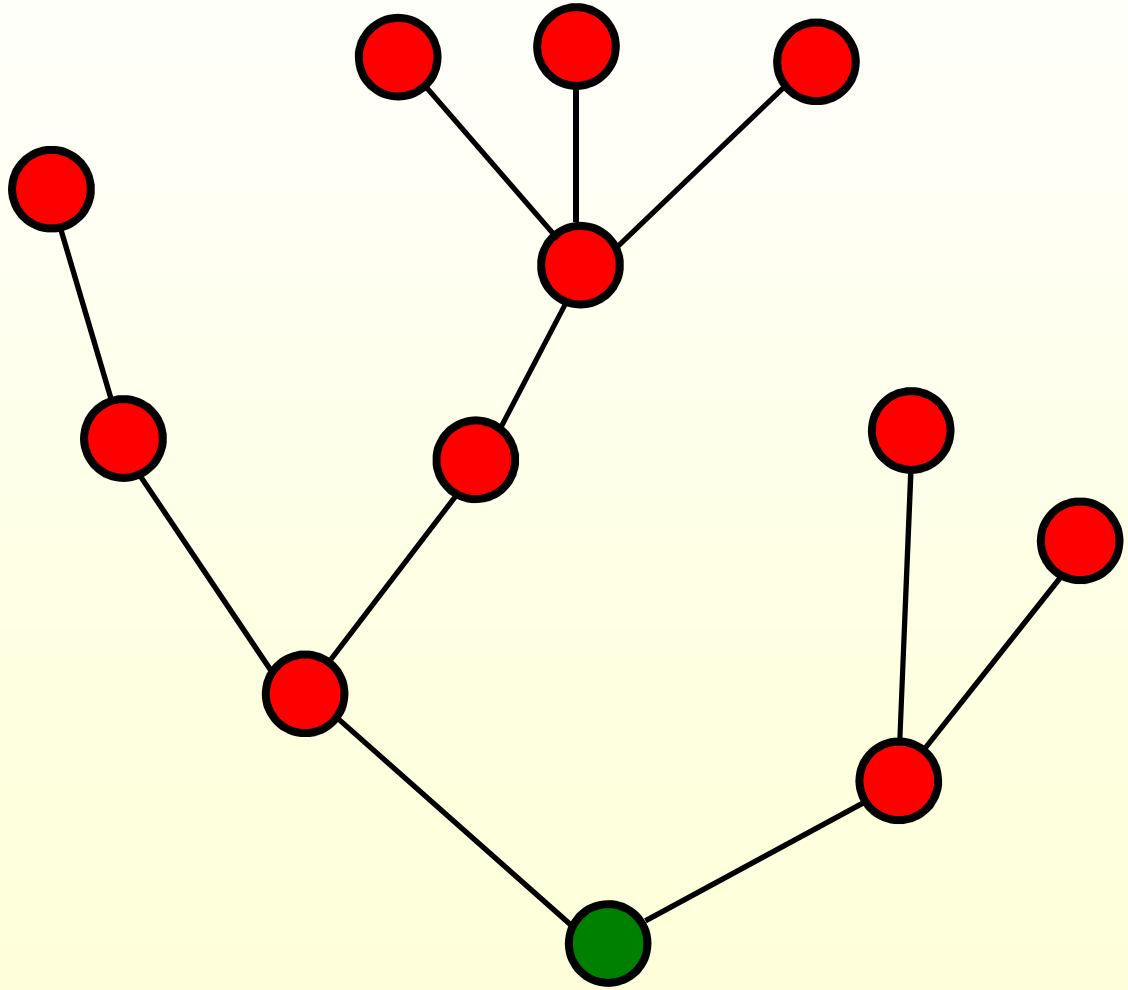
Robustness

- Link failure
- Node failure



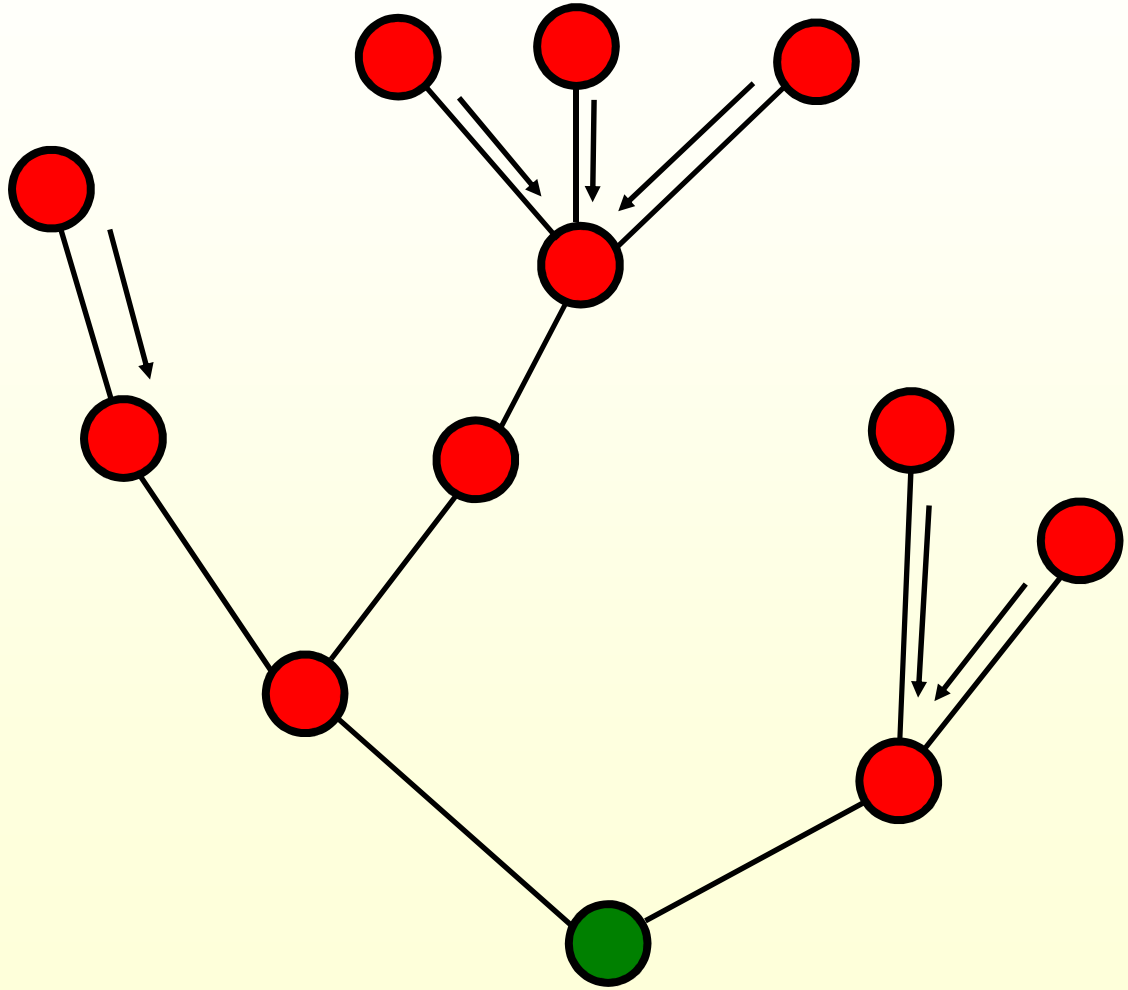
Efficiency

- Number of packets sent
- Rooted MST
- Each node sends 1 packet



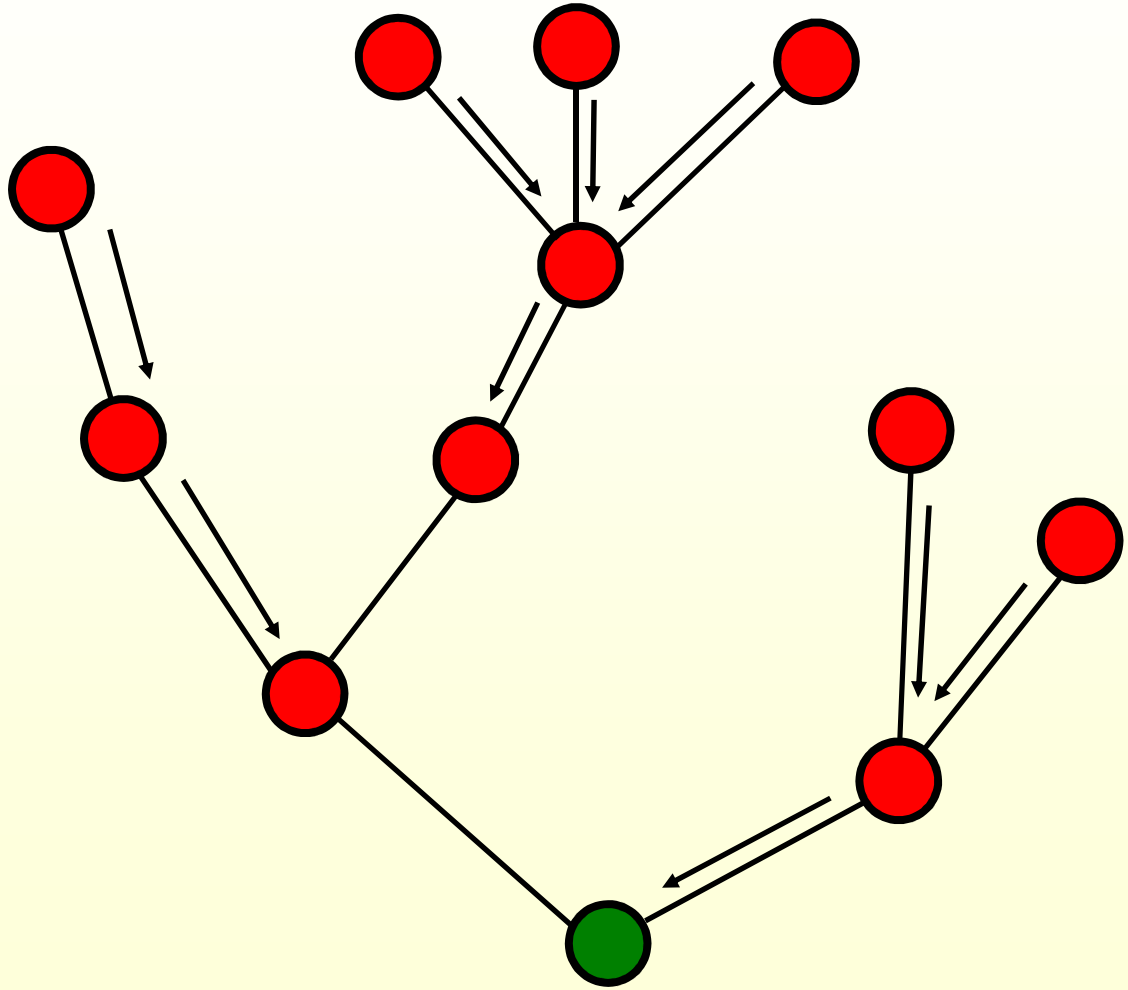
Efficiency

- Number of packets sent
- Rooted MST
- Each node sends 1 packet



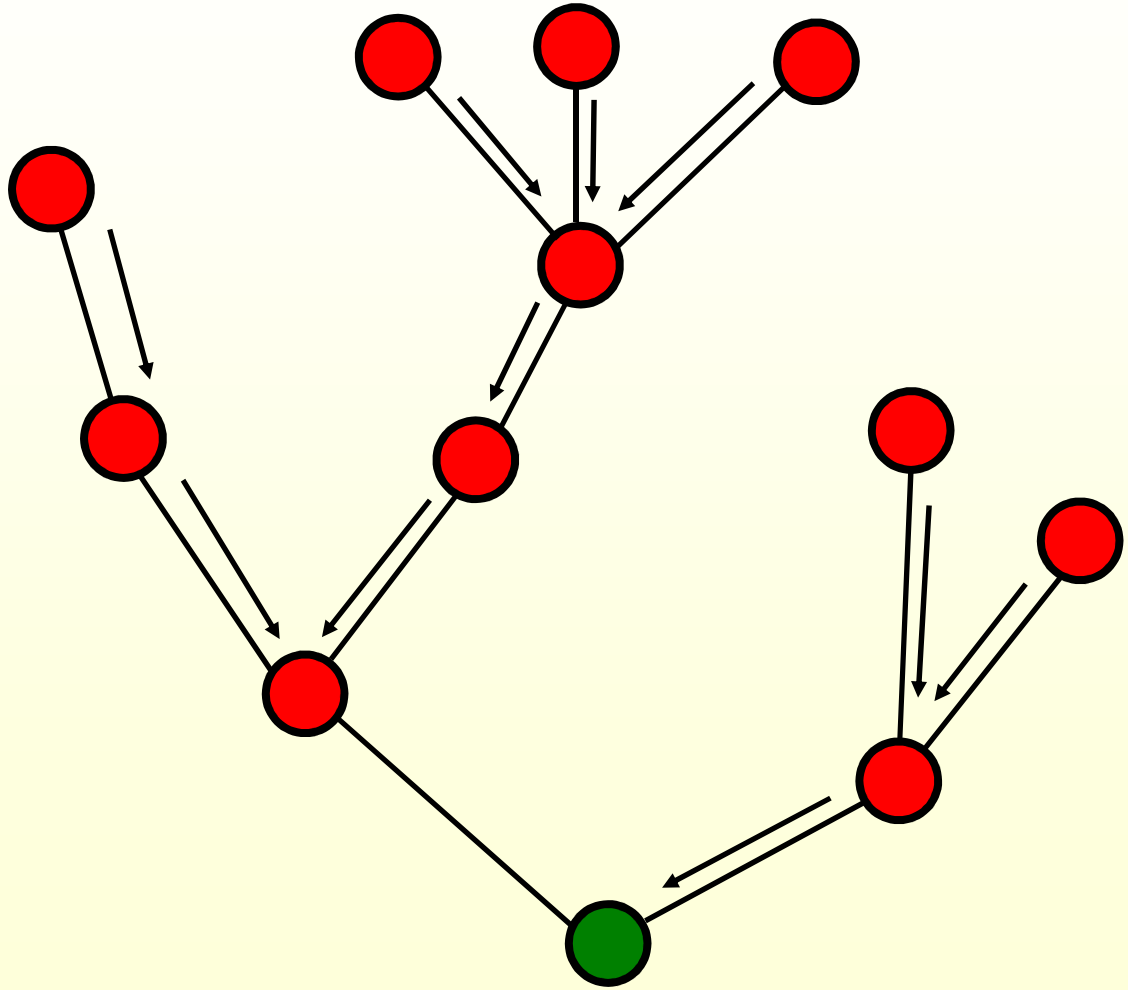
Efficiency

- Number of packets sent
- Rooted MST
- Each node sends 1 packet



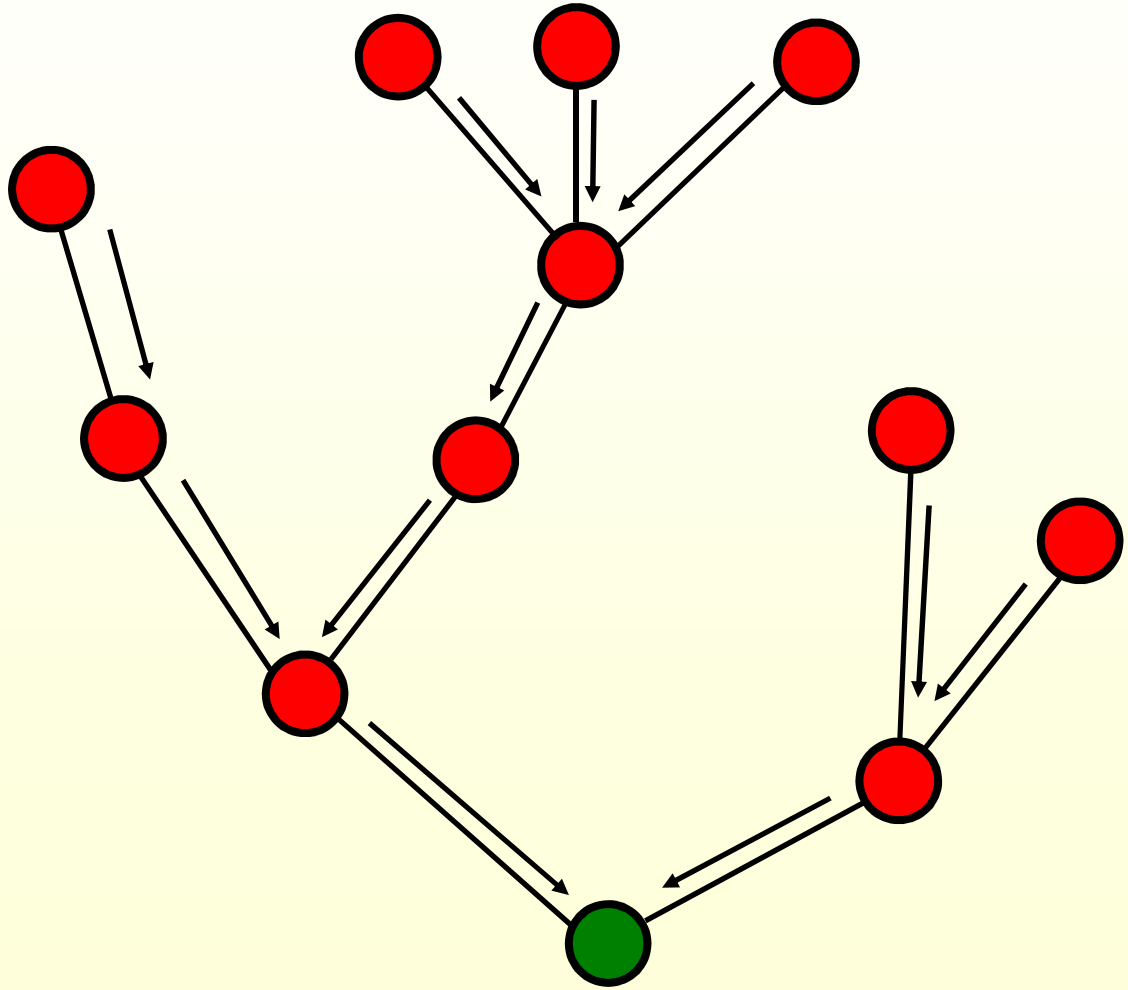
Efficiency

- Number of packets sent
- Rooted MST
- Each node sends 1 packet



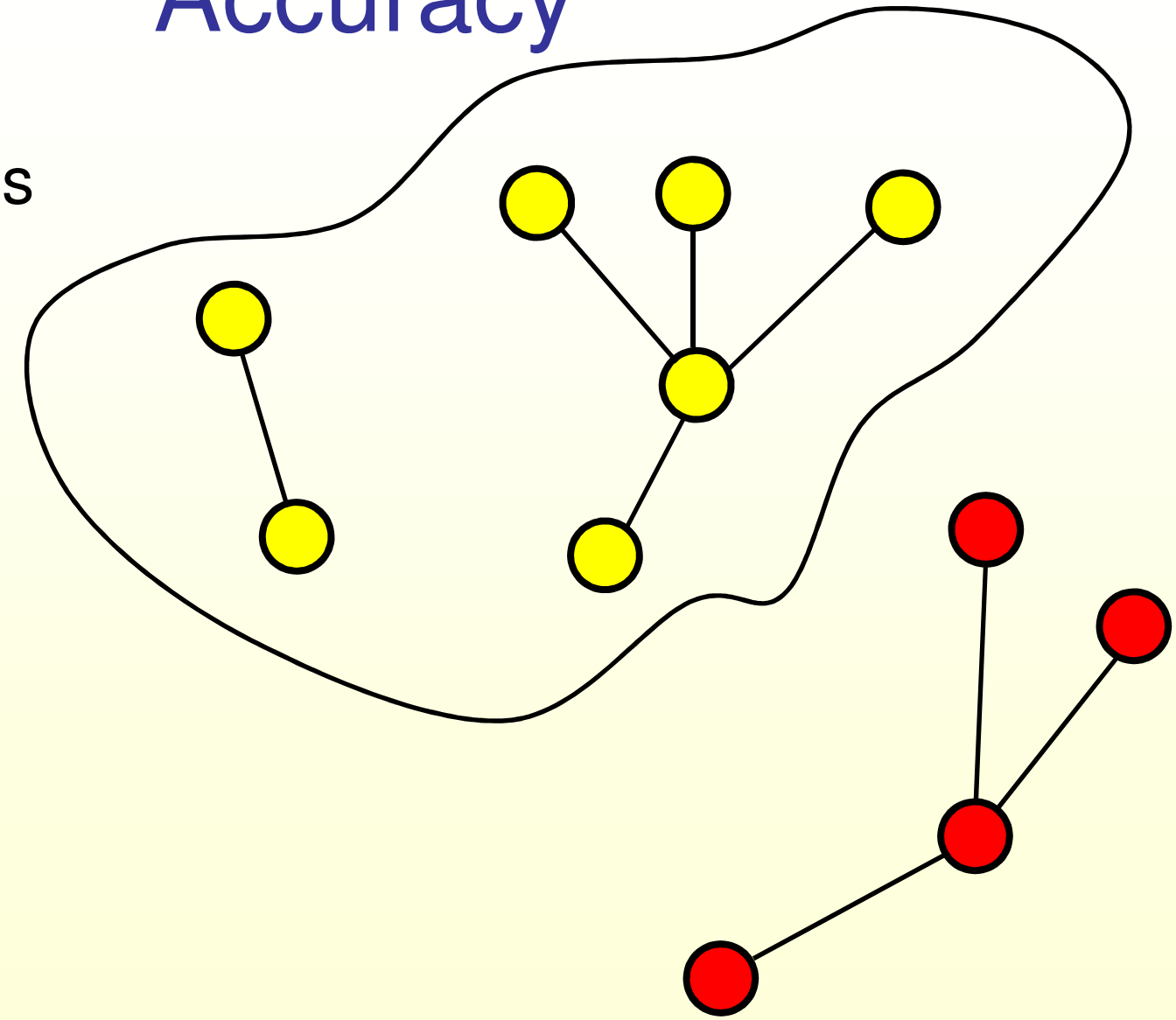
Efficiency

- Number of packets sent
- Rooted MST
- Each node sends 1 packet



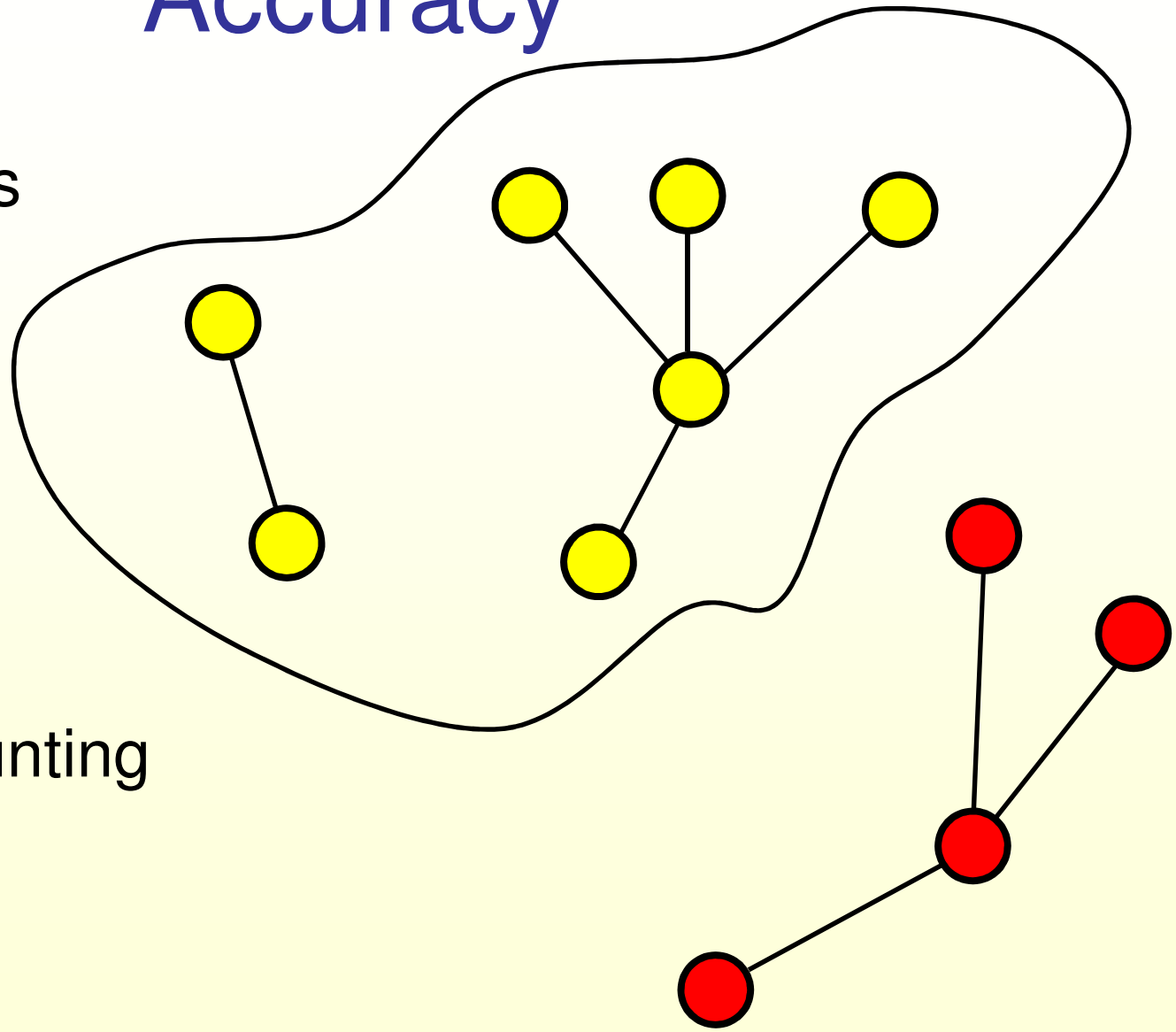
Accuracy

- Robustness



Accuracy

- Robustness



- Double counting

Delivery Methods

- Tree (TinyDB)
  - Not robust
- Complete data collection
  - Not efficient
- Multipath routing (Using multiple parents)
  - Not always efficient/robust/accurate

Detecting Duplicates

- Why not send everything over many paths?
- Duplicates affect most aggregates

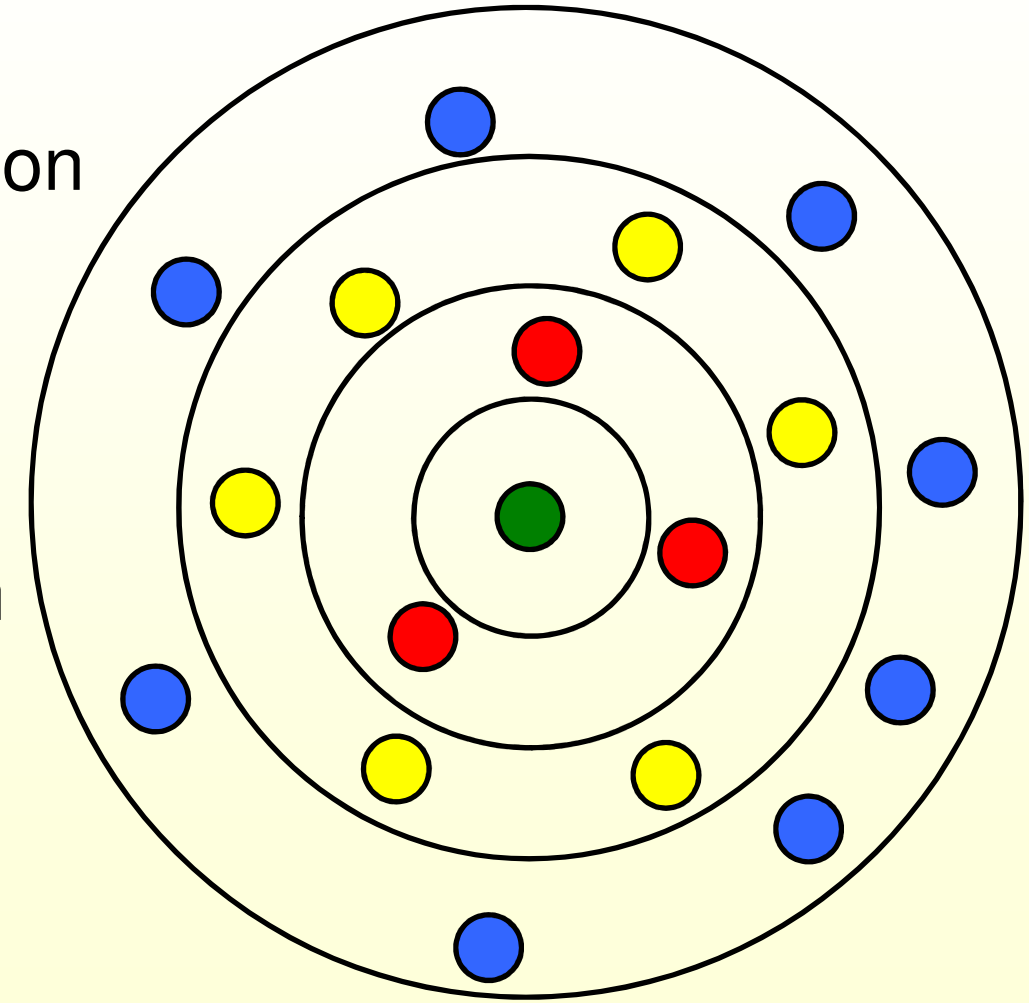
Property	Examples	Affects
Partial State	MEDIAN : unbounded, MAX : 1 record	Effectiveness of TAG
Monotonicity	COUNT : monotonic AVG : non-monotonic	Hypothesis Testing, Snooping
Exemplary vs. Summary	MAX : exemplary COUNT: summary	Applicability of Sampling, Effect of Loss
Duplicate Sensitivity	MIN : dup. insensitive, AVG : dup. sensitive	Routing Redundancy

Synopsis Diffusion

- Idea : MAX, MIN are duplicate insensitive
  - Make other aggregates duplicate insensitive
- Order-Duplicate Insensitive (ODI)
  - Test for correctness
- To come later in the lecture
  - Exact conditions for test
- Multiple parents with no double counting

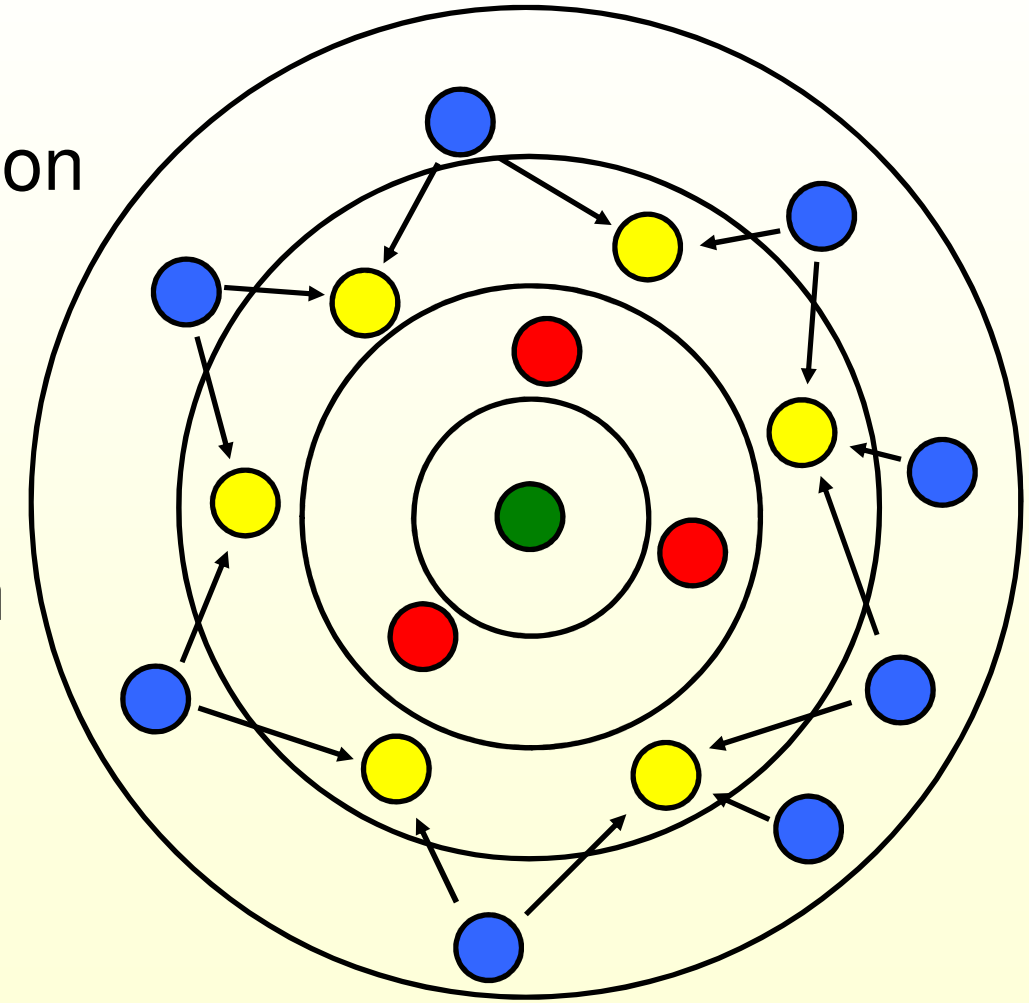
Rings Overlay

- Each node finds hop count from base-station
- Forms a DAG
- Send to all parents in the DAG



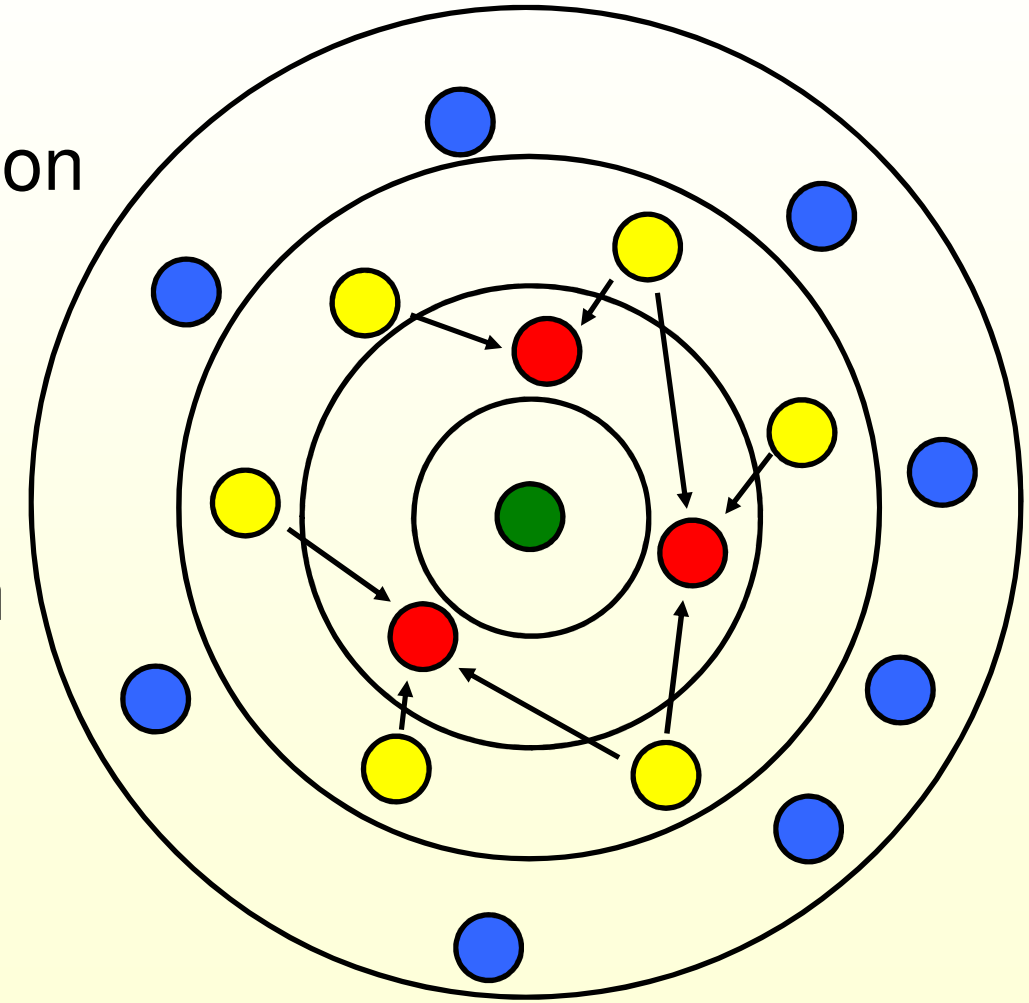
Rings Overlay

- Each node finds hop count from base-station
- Forms a DAG
- Send to all parents in the DAG



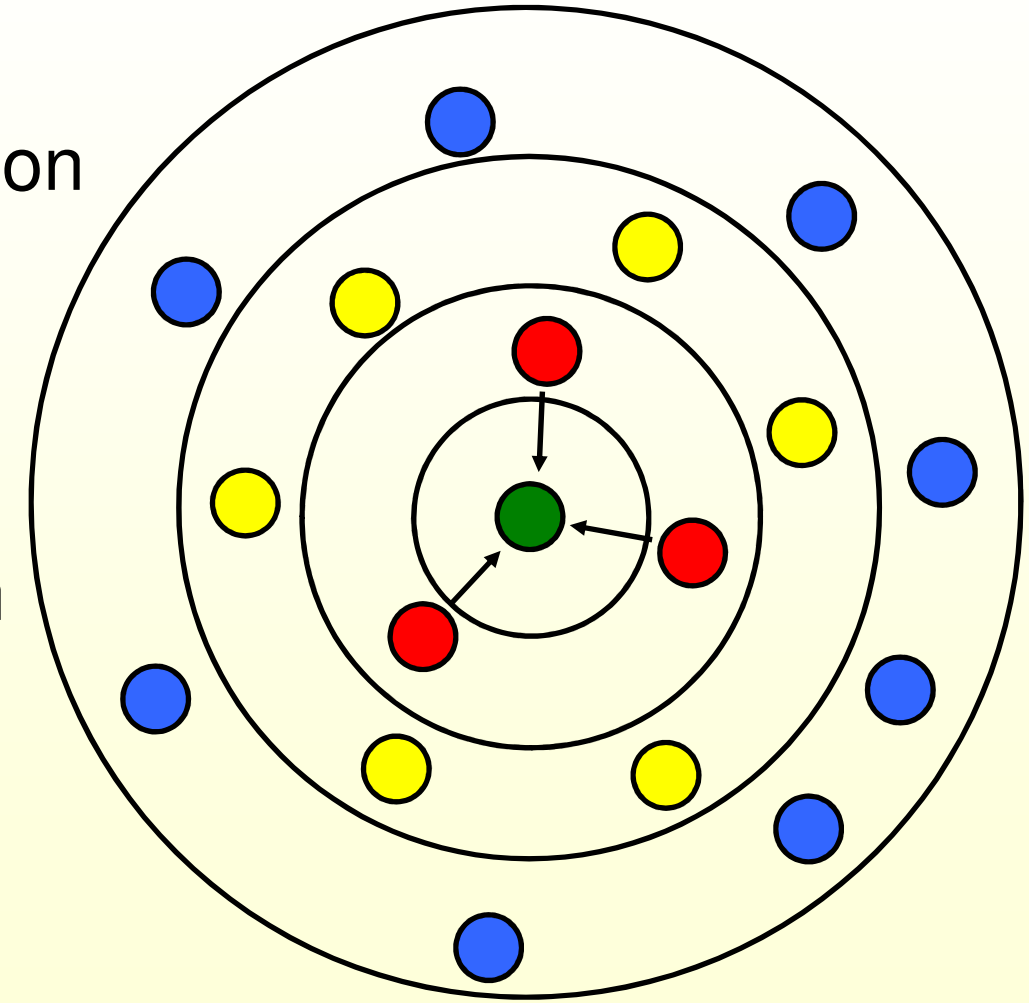
Rings Overlay

- Each node finds hop count from base-station
- Forms a DAG
- Send to all parents in the DAG



Rings Overlay

- Each node finds hop count from base-station
- Forms a DAG
- Send to all parents in the DAG



Approximate Counting

Overview

Approximate Counting

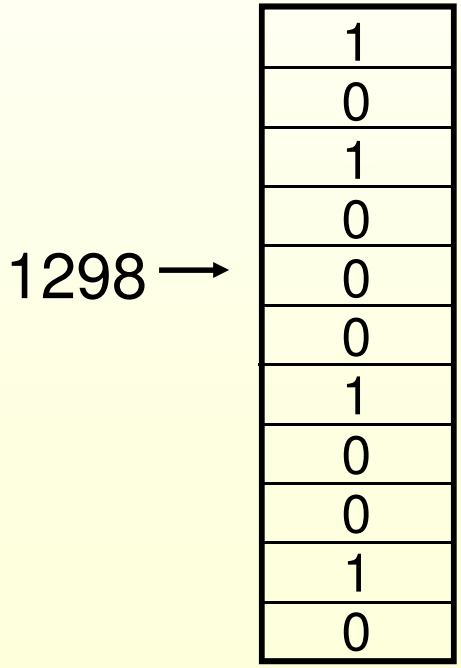
Probabilistic Counting

Distinct Item Counting

Standard Counter

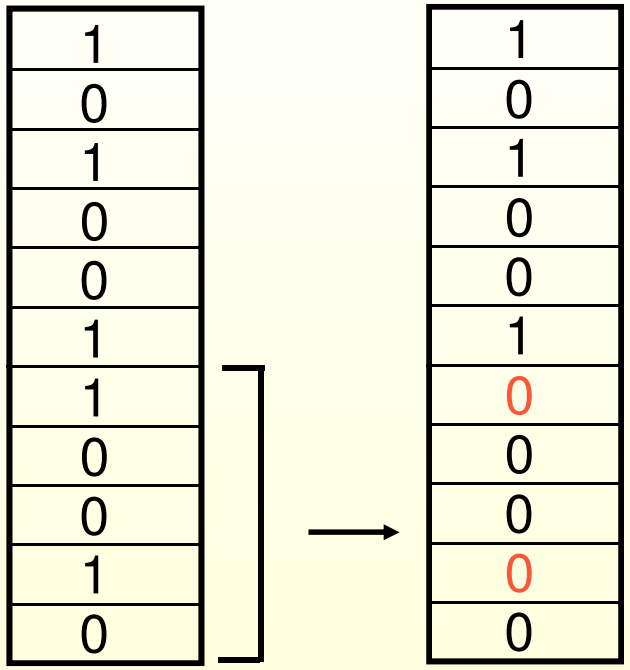
- Counting to N requires O(log n) bits

- Largest number
- A string of ones:
  - 1111….111 ~ 2^N-1



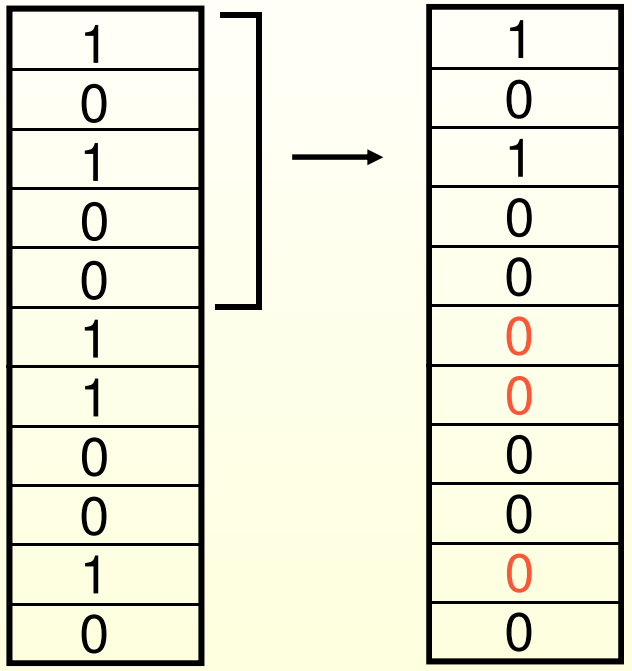
Quantization

- Drop k least significant bits
- Relative error goes down as value increases
- Still need O(log n) bits



Approximate Counter

- Binary representation of x
- Send most significant bit
  - Send its position
  - O(log log n) bits
- Error grows with value

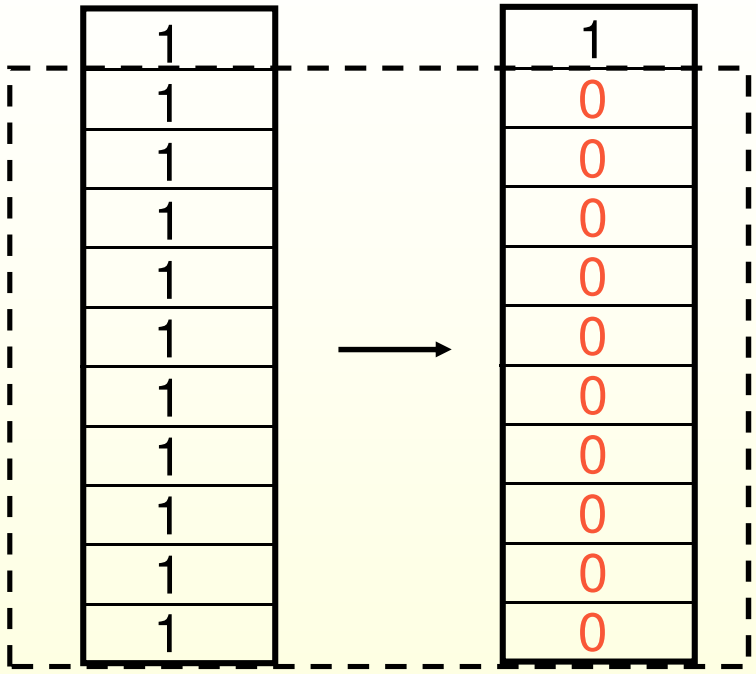


k most significant bits

Sending MSB

- Send 1 number
- Upper bound on error
  - We send 1 at position N
  - All previous bits are 1

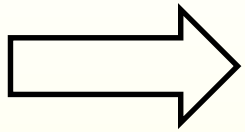
11…1111111 ⇒ 10…0000000



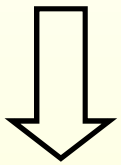
- How quickly does the error grow?
- For sending 1 bit…

Error Bound

- Difference = 2^{N-1}
- Ratio of error
  - $2^{N-1} / 2^{N} = ½$

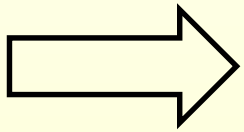


CONSTANT FACTOR APPROXIMATION



1 bit $\Rightarrow$ ½

2 bits $\Rightarrow$ ¼



O(1) bits

Arbitrarily small **relative** error

Adding to a Counter

- Compression of a number to O(log log n) bits
- How do we add to this counter?

$$10000 + 1 = ?$$

- Maintain **O(log n)** bits
- Never increase counter

$$10000 + 1 = 10000$$

- Generally, add m to x
  - Add 1 to x
  - Repeat m times

Flipping a Coin

- How many flips until we see a head?
- Fair coin
  - p = ½
  - E[X] = 2
- Generally
  - p = ?
  - E[X] =1/p
- Suppose each time we add a one, we flip a coin?
- Experiment
  - Toss until you get heads
  - If x>i increment, else pass it on

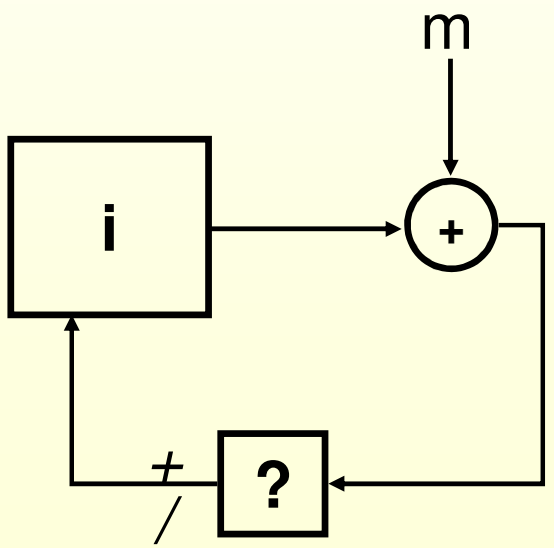
Probabilistic Counting

- To increment counter from i to i+1 requires adding 2^i
- Design an experiment that will take 2^i tries to succeed
- If $i = 5 \sim 2^5$
  - Flip a coin with $p = (½)^5$

R. Morris, “Counting large numbers of events in small registers”

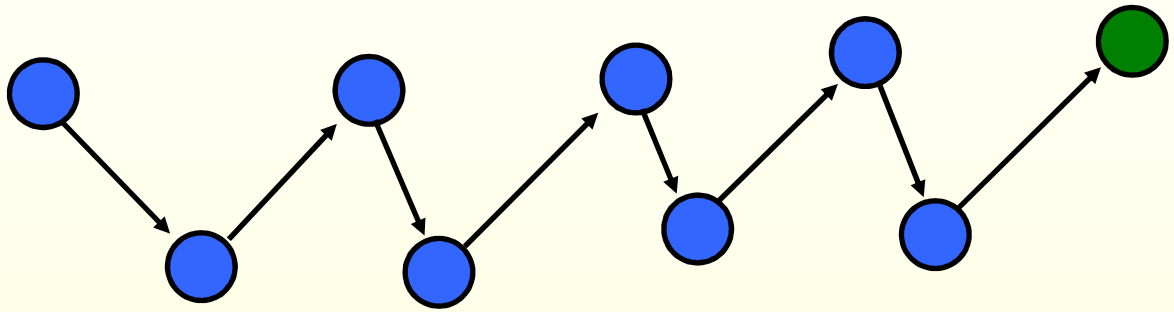
10000 + 10000 = 100000

5 +1 $\Rightarrow$ 6



Counting # of sensors

- Each sensor flips a coin and passes it on



- Cost: O(n log n) vs. O(n log log n)

- Proportional to the number of trials
- Not duplicate insensitive

Counting Distinct Values

- P. Flajolet and G. N. Martin,

  “Probabilistic counting algorithms for database applications”

- Expensive merging operations
  - Keep full lists of items
  - Standard sorting and merging

a,b: size of sets

α,β: number of distinct elements

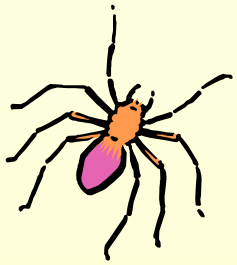
$O(a \log \alpha + b \log \beta)$

$O(a \log a + b \log b + a + b)$

How many or whom?

- Only maintain how many distinct items we've seen

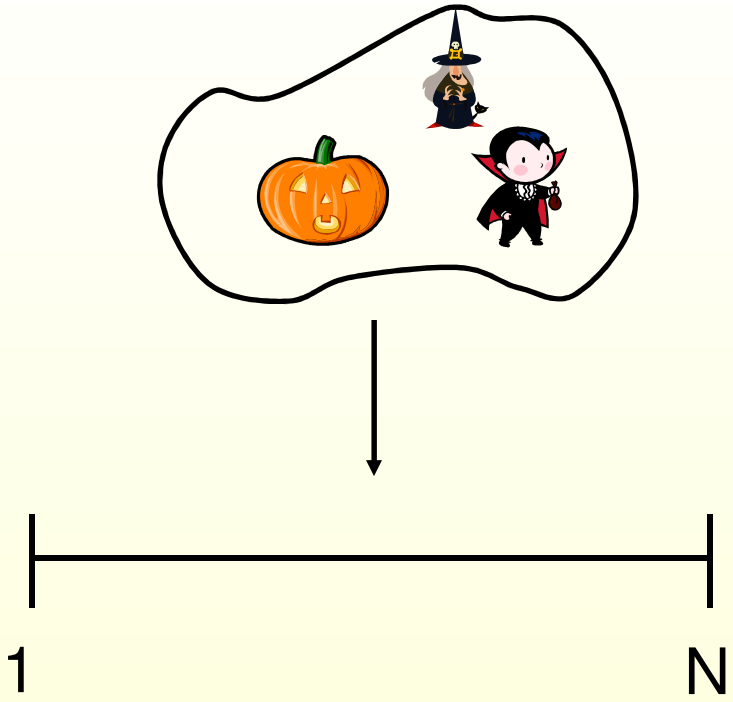
- Have to decide if we've seen this item before?



Hash Function

- Set of items
- Hash to a number
  - Uniformly distributed

    $y = 1 \ldots N$



- Distribution of most significant bit
  - Exponential

$$P\{MSB \geq i\} = P\{x > 2^i\} = 2^{-i}$$

Counting Cardinality

- OR hashes together

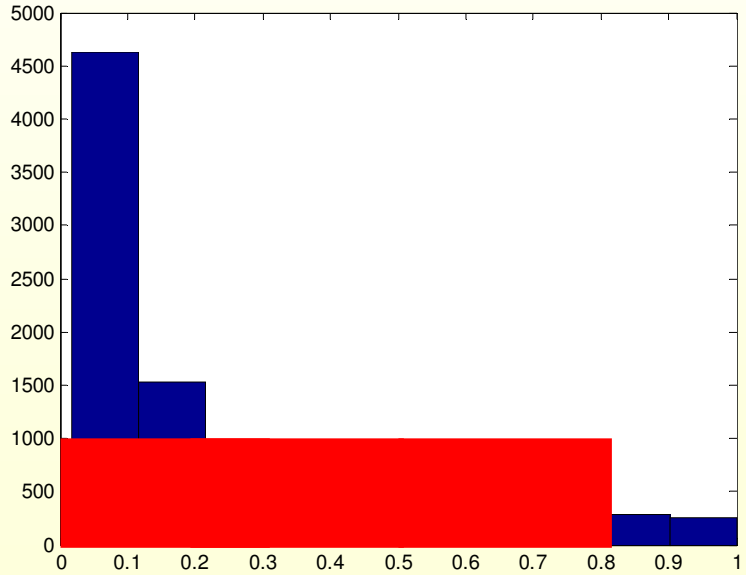
```
01010001000
01000100100
01010001000
10001001000
-----------
11011101100
```

- Left most 0
- Right most 1

→ indicates cardinality

Intuition

- OR is duplicate insensitive
- Probability distribution



$P\{i = 0 \mid i >> \log n\} = 0$

$P\{i = 1 \mid i << \log n\} = 0$

Synopsis Diffusion

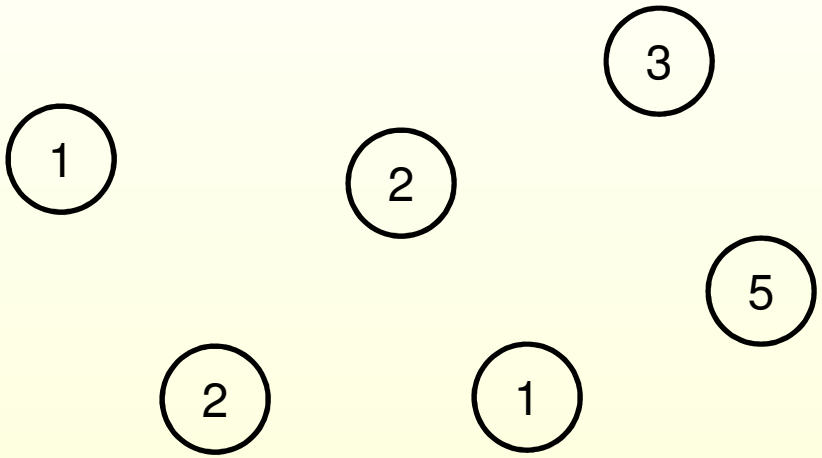
- Counting sensors
  - Each sensor is a unique item
  - Each does a coin toss
  - i = # of coin tosses until heads
- Set i^{th} bit to 1

# of tosses	Probabilitiy
1	2^{-1}
2	2^{-2}
3	2^{-3}
4	2^{-4}
⋮	⋮
n	2^{-n}

S. Nath, P.B. Gibbons, S. Seshan, Z.R. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks"

Algorithm

- N = max. number of nodes
- k > log(N)



10000
01000
01000
10000
00001
10000

11001

$2^2 = 4 \Rightarrow 5.17$

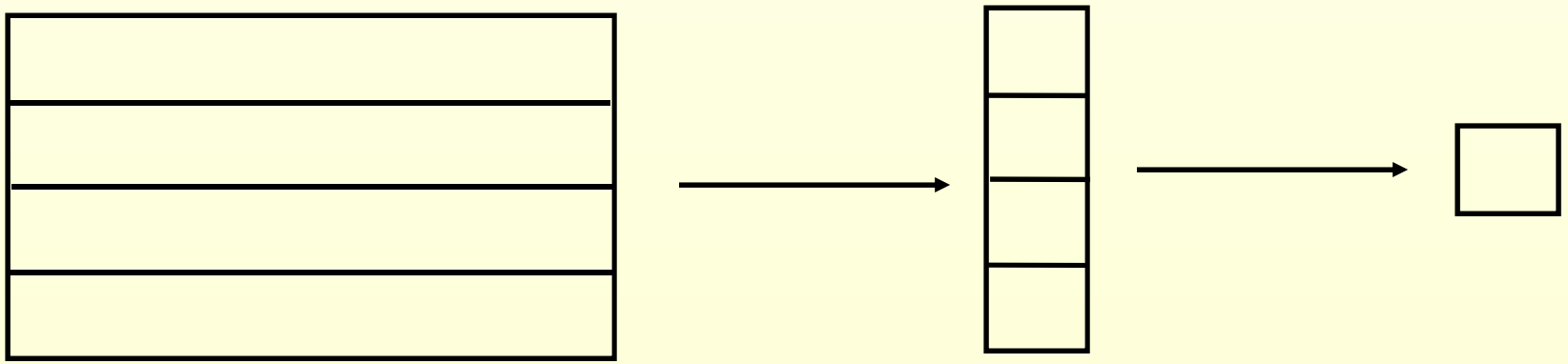
- Return $2^{i-1}/0.77351$

Experiment

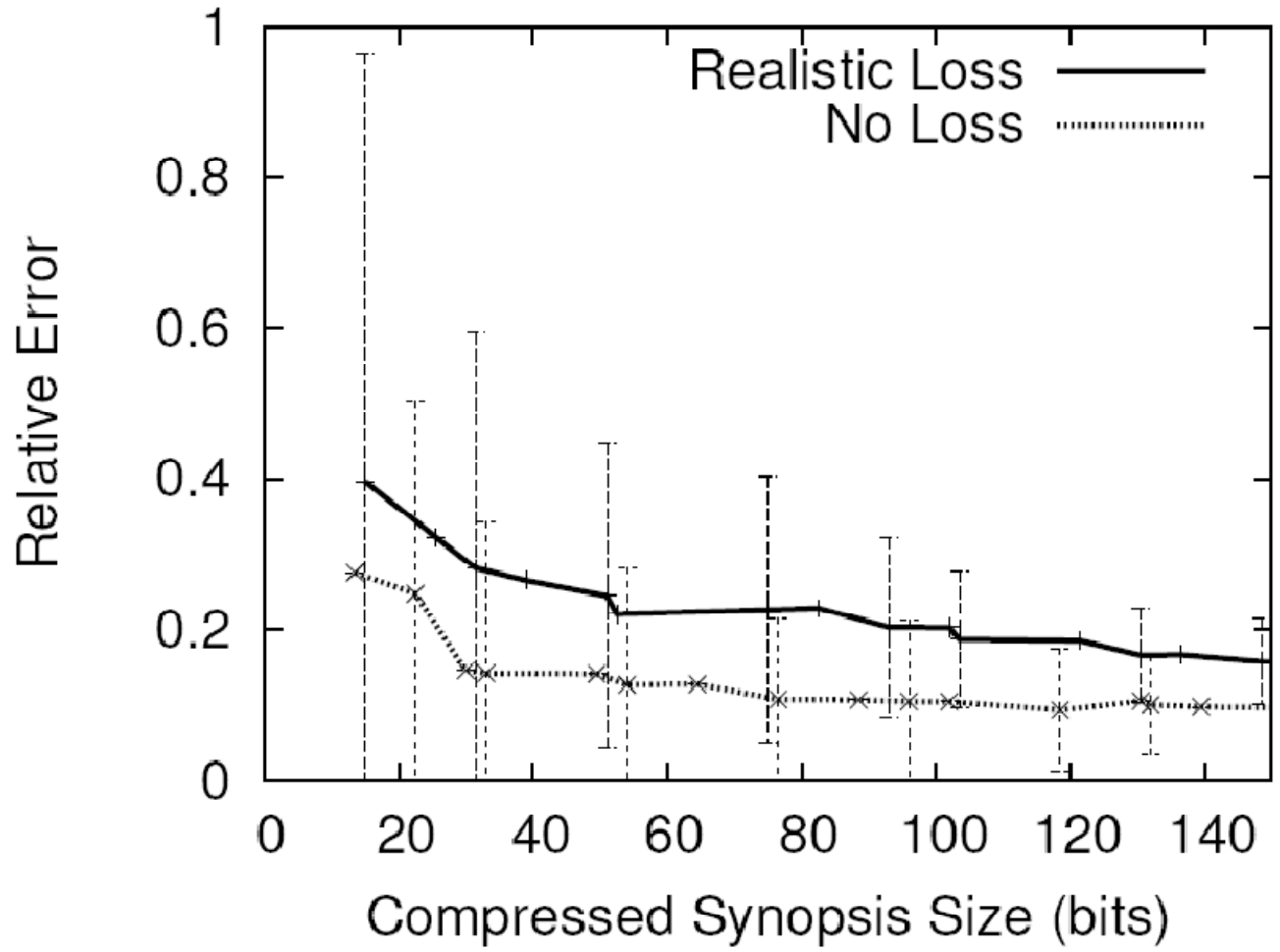
Error

- Send most significant bit
- Reducing error
- Repeat experiment in parallel
- Average results

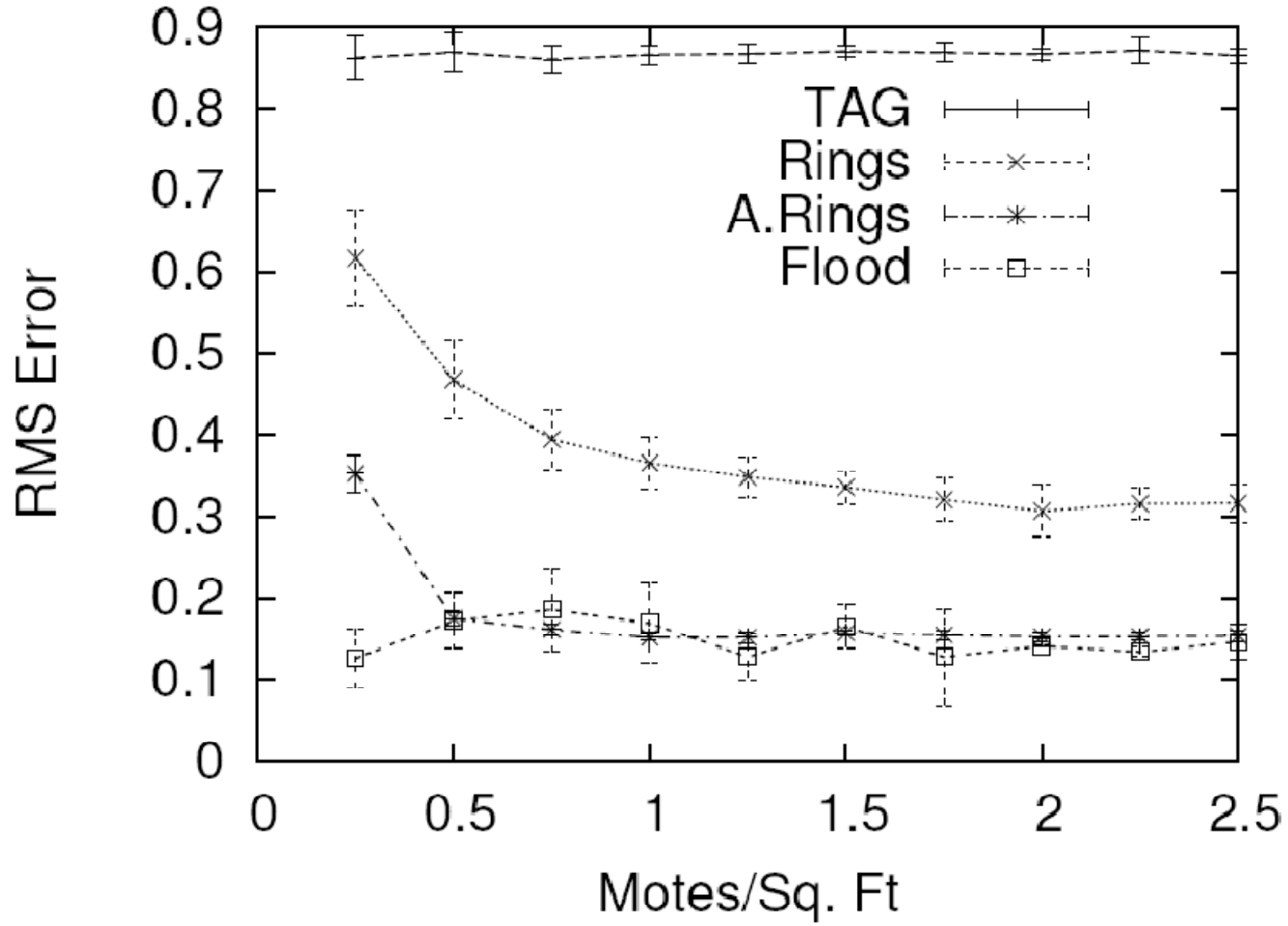
Send K synopsis



Synopsis Size



RMS Error


RMS Error
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
Motes/Sq. Ft
TAG
Rings
A.Rings
Flood

ODI Framework

- Definitions
  - Synopsis generation SG(·)

$$\text{Value} \Rightarrow \text{SG}(\cdot) \Rightarrow \text{Synopsis}$$

  - Synopsis fusion SF(·)

$$\{\text{Synopsis},\text{Synopsis}\} \Rightarrow \text{SF}(\cdot) \Rightarrow \text{Synopsis}$$

  - Synopsis evaluation SE(·)

$$\text{Synopsis} \Rightarrow \text{SE}(\cdot) \Rightarrow \text{Value}$$

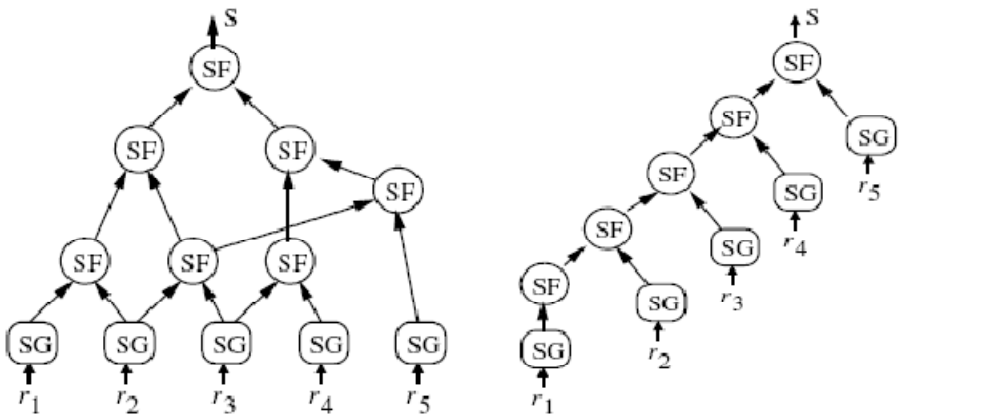
ODI Test

- Definitions
- **P1**: SG() preserves duplicates
- **P2**: SF() is commutative
- **P3**: SF() is associative
- **P4**: SF() is same-synopsis idempotent
  - SF(s,s)=s

Correctness

- P1-P3 are obvious
- P4: SF(s,s)=s
  - Nothing about intersection of sets

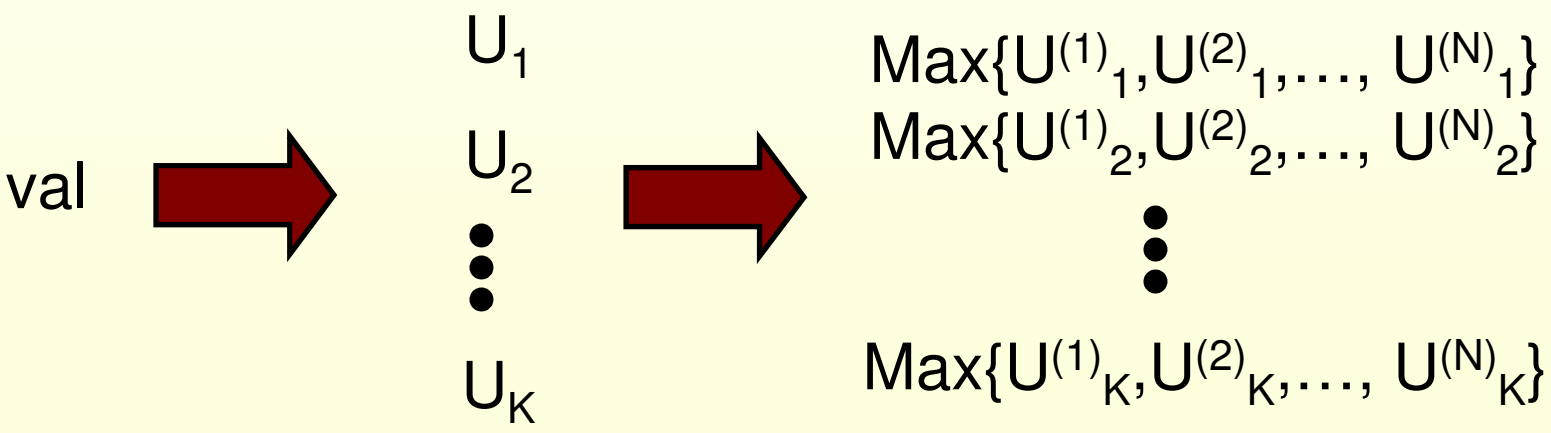
- Proof: Transform any DAG to a canonical tree



(a) Aggregation DAG (b) Canonical left-deep tree

Uniform Sampling

- Associate a sensor reading with a bit
- Keep only track of most significant bit item
- Repeat *K* times



Aggregates

- Holistic aggregates
- k-th statistical moments
- k-th percentile value

- Can be approximated using uniform sample
  - ε additive error, w.p. 1-δ
  - Sample size $O(1/\varepsilon^2 \log 1/\delta)$

Z. Bar-Yossef, R. Kumar, and D. Sivakumar,
"Sampling algorithms: lower bounds and applications"
STOC, 2001.

Most Popular Items

- Keep k most popular items
- Generate double (value, weight)

- Keep only the values associated with K top bits
- Run over independent sets

N. Alon, Y. Matias, and M. Szegedy,
“The space complexity of approximating the frequency moments”
J. of Computer and System Sciences, 58:137-147, 1999.

Continuous Values

- Exponential variables

$$f(x) = \lambda e^{(-\lambda x)}$$

$$F(x \geq X) = e^{(-\lambda x)}$$

$$\min(x_1, x_2, \ldots, x_n) = X$$

$$F(x_1 \geq X) F(x_2 \geq X) \ldots F(x_n \geq X) =$$

$$e^{-\lambda_1 X} e^{-\lambda_2 X} \ldots e^{-\lambda_3 X} = e^{-(\Sigma \lambda_i) X}$$

E. Cohen, “Size-estimation framework with applications to transitive closure and reachability”

Difference

- Efficient sketch/synopsis not known
- Problem:
  - Subtracting 2 large numbers can give you a small number.
- Approach for adding
  - No longer a constant approximation

Summary

- Difficulty of aggregate
  - Partial State Record Size
  - Synopsis Size
- Can gain efficiency and robustness by allowing a small error
- Randomized algorithms can be very efficient