

Data Aggregation and Data Dissemination in Wireless Sensor Networks

Shashi Prabh
ISEP, Portugal

Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Introduction

WSN nodes perform sensing of a physical environment. The sensed data from multiple sources is collectively used to make inferences.

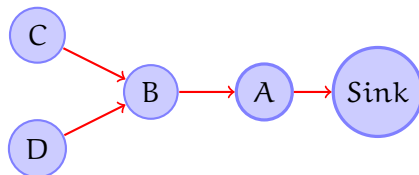
- Large amount of raw data
- Correlated data

Communication in the network is significantly reduced by:

- Elimination of redundant data
- Accumulation and processing at intermediate nodes
 - ▶ Opposed to unicasts to sinks
- Sensor node selection – not the topic of this talk

Why Data Aggregation in WSN? - I

- Communication is the most energy consuming functionality
 - ▶ Energy consumed in transmitting one bit over 100m $\approx 1000 \times$ Energy consumed per instruction execution
- Efficient bandwidth utilization

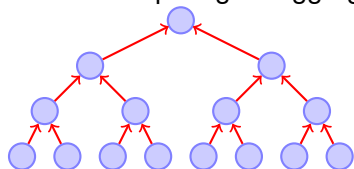


Example: What is the average temperature?

- Without aggregation: 9 messages
- With aggregation: 4 messages

Why Data Aggregation in WSN? - II

Cost of computing an aggregate such as AVERAGE on a binary tree:



Message count **per query response** without aggregation in a binary tree of depth d :

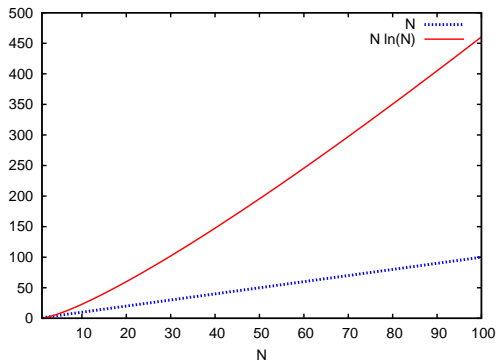
$$\begin{aligned}
 S &= 2(1 * 2^0 + 2 * 2^1 + 3 * 2^2 + \dots + d * 2^{d-1}) \\
 &= (d - 1)2^{d+1} + 2 \approx N \lg N,
 \end{aligned}$$

N is the number of nodes.

In general, S grows as $N \log_b N$, b is the branching factor.

Why Data Aggregation in WSN? - II

- Message count if aggregation is used : N
- Message count if aggregation is not used : $N \log_b N$



Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Address-Centric vs. Data-Centric Networking

Address-centric networking paradigm (The Internet):

- Data is external to the network
 - ▶ Typically, the network is agnostic to the contents of a payload
- The network provides end-to-end packet delivery service
- Routing of data packets depends upon the addresses of end hosts
- The addresses reflect network topology

Address-Centric vs. Data-Centric Networking

Sensor networks are different:

- Data is generated in the network itself
- Data – not the identity of the node that generates the data – is of significance
 - ▶ Tracking, monitoring
- Assignment of globally unique addresses in large-scale deployments is a problem
 - ▶ WSN hardware don't have unique identifiers (Exceptions exist.)
- The topology of a typical WSN deployment is too dynamic

How do we route packets without addresses?

Data-Centric Networking Fundamentals

Data-centric communication is carried out using publish/ subscribe mechanism. Data is named by attribute-value pairs.

Addressing: [\[CARZANIGA ET AL., 2001\]](#)

- **Receiver predicate** defines the messages of interest to a node
- **Sender predicate** defines the messages that a node will send

Let \mathcal{M} denote the set of all possible messages, and let $\mathcal{P} : \mathcal{M} \rightarrow \{\text{true}, \text{false}\}$ denote the set of all predicates over \mathcal{M} . Then \mathcal{P} and \mathcal{M} define a data-centric (or, content-based/ attribute-based) addressing scheme.

$$\text{cprog} = \begin{cases} \text{true} & \text{if payload}(m) \text{ is a C program} \\ \text{false} & \text{otherwise} \end{cases}$$

Data-Centric Networking Fundamentals

- The receiver predicate p_n is a data-centric address of node n
- Message m 's destinations include node n if $p_n(m) = \text{true}$

Routing:

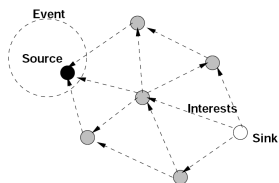
- At all intermediate nodes n' , $p_{n'}(m) = \text{true}$
 - ▶ No need for name binding service

Note:

- Messages for which $p_n(m) = \text{true}$ can be aggregated at node n
- Aggregation should be done close to the sources

Data-Centric Networking: Directed Diffusion [INTANGOWIWAT ET AL., 2000]

Sensing tasks are described as attribute-value pairs called **interests**. Sink “pulls” the data by flooding interests in the network.



Data-Centric Networking: Directed Diffusion [INTANGOWIWAT ET AL., 2000]

Example of an interest:

```

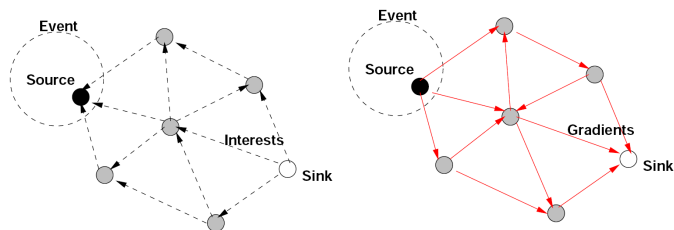
type                =  vehicle //detect any vehicle
sensing-interval    =  10 ms //report detections every 10 ms
target-area         =  [(0,0), (100,100)]
timestamp           =  12:01:25
expire-at           =  12:20:34
  
```

Example of a response:

```

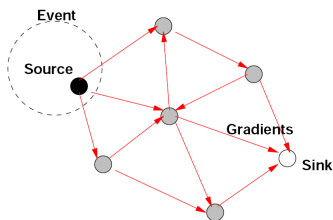
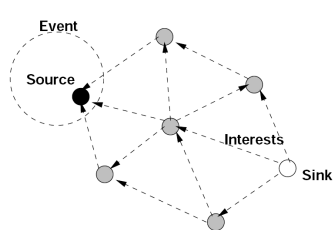
type                =  vehicle
instance            =  car //target classification
location            =  (50,90)
timestamp           =  12:11:25
confidence          =  0.8
  
```

Routing in Directed Diffusion: Gradients



- The protocol works in unicast mode
- Interests flooding: nodes may resend interests
- An entry in the interest cache for each interest received
- For each re-sent interest received, a gradient is added to the interest

Routing in Directed Diffusion: Gradients



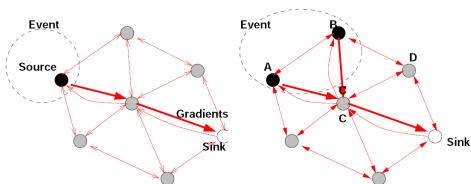
The gradient fields contain:

- Data rate requested by the neighbor
- Lifetime of the entry

Problem:

- Loops
 - ▶ Maintain a cache of recent interests propagated

Route Reinforcements in Directed Diffusion: Two-Phase Pull



- Exploratory phase of lower data rate → higher data rate from fewer neighbors.
 - ▶ Needs warm-up time
- Nodes with the earliest time stamps are preferred
- Reinforce nodes that report new events
- Negatively reinforce nodes with no recent events

Provides very basic aggregation: suppression of duplicate messages.

Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination**
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Data Dissemination in WSN

Some applications:

- Query propagation
- Routing
- Time synchronization
- Multicast
- Updating nodes, calibrating sensors, changing network parameters

Data Dissemination in WSN

Characteristics of dissemination algorithms suited for WSN:

- Simplicity
- Scalable to large number of nodes, localized
- Adaptive to topological changes
- Energy-efficient

Epidemic Algorithms

Epidemic algorithms disseminate information via local interactions.

Transmission of a message triggers re-broadcasts by receivers of the message.

Message handling schema [\[BASED ON GANESHAN ET AL., 2002\]](#):

Let S be local state of node and R a random number.

If message m is received for the first time, then

Take local action based on m : $S \leftarrow f_1(m, S)$

Make Boolean retransmit decision $D = f_2(S, R)$

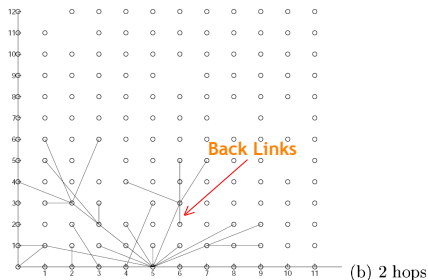
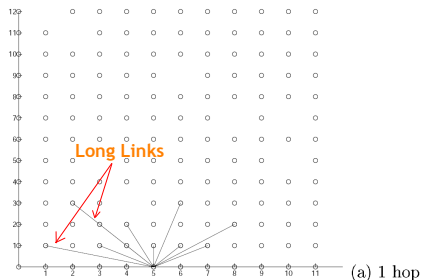
If D is yes, then

Compose message $m = f_3(m, S)$

Broadcast m .

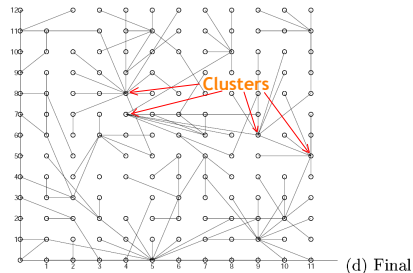
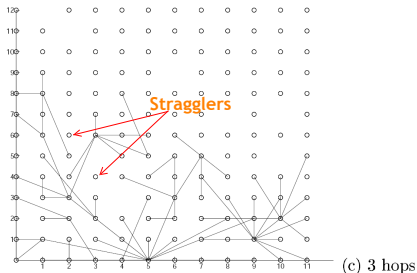
Flooding

- Every node re-broadcasts a message once
- The most frequently used dissemination algorithm in WSN
- Empirical study in WSN [GANESHAN ET AL., 2002]



Flooding

- Every node re-broadcasts a message once
- The most frequently used dissemination algorithm in WSN
- Empirical study in WSN [GANESHAN ET AL., 2002]

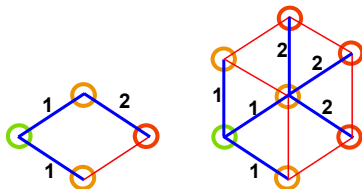


Snapshots from a single run of flooding on the experimental testbed

Flooding - II

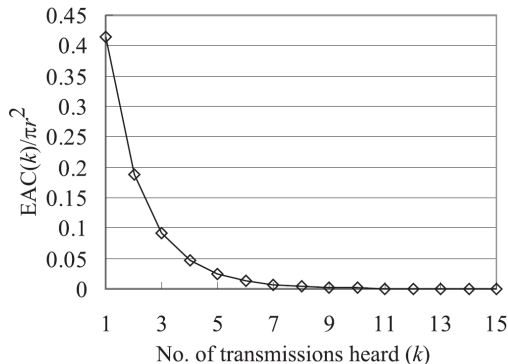
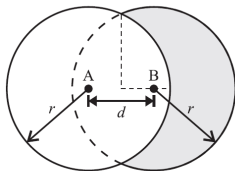
Flooding begets problems!

- Non-uniform propagation
- Broadcast Storm (This discussion of flooding follows [\[NI ET AL., 1999\]](#))
 - ▶ Redundancy, excessive collision and contention
 - ▶ emails: “Send it to 10 people for good luck ...”



Only 2 transmissions are needed in the two cases but flooding algorithm will send 4 in the first and 7 in the second.

Flooding - II



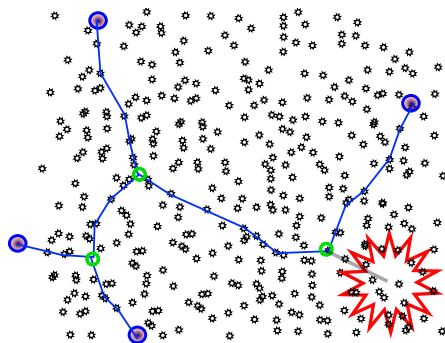
A rebroadcast covers only 41% additional area on average. (The empirical data obtained by [\[GANESHAN ET AL.\]](#) is similar.) The range of additional coverage is 0 – 61%.

Fixing the Flooding!

- **Gossiping:** forward a message with some probability $P < 1$
- **Counter-Based:** inhibit when the number of broadcasts received is larger than certain threshold
- **Distance-Based:** inhibit when additional coverage is low
- **Location-Based:** Same as distance-based except that precise location of the nodes are known (GPS)
- **Cluster-Based:** only cluster heads or gateways broadcast. Can be combined with above variants

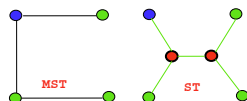
Multicast in WSN

- Streaming data about a fire to a group of firefighters
- Sending periodic actuation commands to a group of nodes
- Pushing data to a group of sinks after aggregation



Multicast in WSN

Minimum Spanning Tree vs. Steiner Minimum Tree



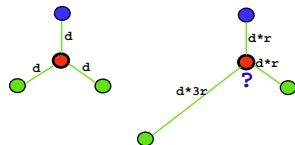
- Minimum Spanning Tree can be computed in $O(n \log n)$ time
- Steiner Minimum Tree is an NP-hard problem
 - ▶ Approximation algorithm time complexity: $O(mn^2)$, where $m = \#$ non-terminals, $n = \#$ terminals.

Minimum Multicast Tree [PRABH ET AL., 2005]

What is the structure, if any, of a minimum cost multicast tree?

- Minimum Steiner Data Caching Tree (MSDCT)
- Generalized form of Euclidean Steiner minimum tree (SMT)
 - ▶ Bandwidth-weighted SMT
 - ▶ Reduces to Euclidean SMT when data rate is the same on all branches

Euclidean Steiner Tree vs. MSDCT:



The geometry of our problem is different!

Minimum Multicast Tree - II [PRABH ET AL., 2005]

- Nodes of the multicast tree are locations of data caches
 - ▶ Multicast can be asynchronous
- Caches can be arbitrarily large number of hops apart
 - ▶ Hop lengths are assumed to be approximately equal

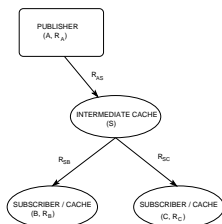
Cost Metric:

The weight w_{AB} of the edge connecting two nodes A and B is defined as:

$$w_{AB} = d_{AB} * R_{AB}$$

where d_{AB} is the Euclidean distance between the nodes, and R_{AB} is the data refresh rate on the edge.

Minimum Multicast Tree - II [PRABH ET AL., 2005]

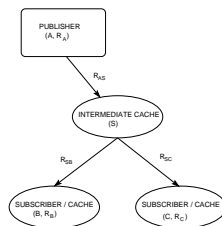


The optimal data refresh rates on the edges are determined on the basis of the following two observations:

- The maximum refresh rate on any of the branches can not exceed the rate at which data is generated by the publisher
- Data need not be sent at a higher rate than the subscriber's requested rate

Minimum Multicast Tree - II

[PRABH ET AL., 2005]



R_N = Requested data refresh rate of node N .

$R_{N_i N_j}$ = Actual refresh rate on the edge connecting N_i and N_j .

Refresh Rate Rules:

Rule-1: If $R_A \geq R_B \geq R_C$ then $R_{AS} = R_B = R_{SB}$; $R_{SC} = R_C$

Rule-2: If $R_B \geq R_A \geq R_C$ then $R_{AS} = R_A = R_{SB}$; $R_{SC} = R_C$

Rule-3: If $R_B \geq R_C \geq R_A$ then $R_{AS} = R_A = R_{SB} = R_{SC}$

Minimum Multicast Tree - III [PRABH ET AL., 2005]

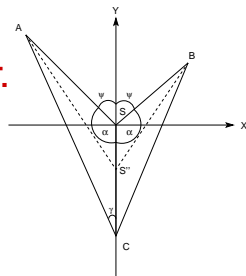
Properties of MSDCT

- The degree of internal nodes of MSDCT is at-most 3.
- At-least two of the internal angles of the nodes of MSDCT are equal.
- Internal angles are (r is the normalized data rate):

$$\psi = \arccos(r/2)$$

$$\pi/2 \leq \alpha \leq \text{MIN}(2\pi/3, \pi - \angle ACB/2)$$

The most distinguishing feature of MSDCT.



Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Tiny Aggregation (TAG) [MADDEN ET AL., 2002]

- Provides interface for queries in SQL-like syntax
- Distributes queries and forms a tree for upstream data
- Supports streaming of data aggregates
- Mostly similar to COUGAR [YAO ET AL., 2003]

Issues:

- Relies heavily on bidirectional connectivity
- No mechanism to deal with duplicates

Tiny Aggregation (TAG) [MADDEN ET AL., 2002]

Routing: via a tree rooted at the sink. Routing tree formation algorithm (invoked periodically):

```

Do upon receiving message  $M(n, l)$  from node  $n$  at level  $l$ 
  if this node's level  $> l + 1$ 
    this node's level =  $l + 1$ 
    this node's parent =  $n$ 
    broadcast  $M(\text{this node's id}, l + 1)$ 
  
```

Detection of a leaf node n' : n' does not hear any message of the form $M(n', .)$

Tiny Aggregation (TAG) - II [MADDEN ET AL., 2002]

Query Model:

- Single table called **sensors**
 - ▶ Append-only relational table
 - ▶ One attribute per sensing functionality

Form of queries:

```
SELECT {agg(expr), attrs} FROM sensors
      WHERE {selPreds}
      GROUP BY {attrs}
      HAVING {havingPreds}
      EPOCH DURATION i
```

Example:

```
SELECT {MAX(temperature), building} FROM sensors
      WHERE block = ALL
      GROUP BY building
      HAVING MAX(temperature) > 100
      EPOCH DURATION 60s
```

Tiny Aggregation (TAG) - III [MADDEN ET AL., 2002]

The agg clause:

- An initializer i
- A merging function f
- An evaluator e

The aggregated record $\langle Z \rangle = f(\langle x \rangle, \langle y \rangle)$, where $\langle x \rangle$ and $\langle y \rangle$ are partial records.

Example: [AVERAGE](#)

A partial record is the tuple $\langle \text{SUM}, \text{COUNT} \rangle$

$i(\text{SUM}) = \langle \text{SUM}, 1 \rangle$

$f(\langle \text{SUM}_1, \text{COUNT}_1 \rangle, \langle \text{SUM}_2, \text{COUNT}_2 \rangle) = \langle \text{SUM}_1 + \text{SUM}_2, \text{COUNT}_1 + \text{COUNT}_2 \rangle$

$e(\text{SUM}, \text{COUNT}) = \text{SUM} / \text{COUNT}.$

Tiny Aggregation (TAG) - III

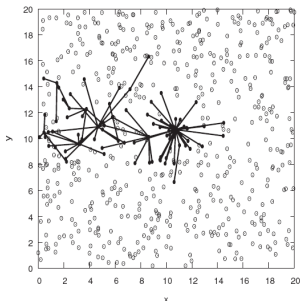
[MADDEN ET AL., 2002]

A classification of aggregates:

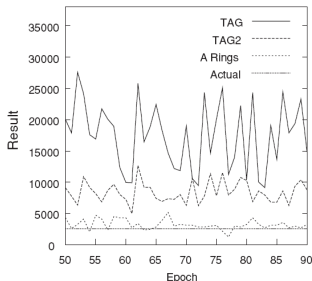
	MAX, MIN	COUNT, SUM	COUNT-DISTINCT	MEDIAN
Duplicate sensitive	No	Yes	No	Yes
Type	Exemplary	Summary	Summary	Exemplary
Monotonic	Yes	Yes	Yes	No
Partial State	Distributive	Distributive	Unique	Holistic

Synopsis Diffusion [NATH ET AL., 2008]

Motivation: If trees are used for data aggregation, such as in TAG, a link failure leads to loss of data from an entire sub-tree.



(a) Nodes accounted for in TAG



(b) Avg computed by different schemes

However, if aggregation of duplicate sensitive aggregates, such as COUNT, is done on a graph, one needs to solve the problem of making the process insensitive to duplicate messages.

Synopsis Diffusion [NATH ET AL., 2008]

- Synopsis is a digest of data
- Partial aggregates are represented by synopses
- **Order and duplicate insensitive** (ODI) synopses

The aggregation process:

Let i denote sensor data, s denote synopsis and α denote the desired aggregate.

- ① Synopsis generation function $SG : i \rightarrow s$
- ② Synopsis fusion function $SF : (s_1, s_2) \rightarrow s$
- ③ Synopsis evaluation function $SE : s \rightarrow \alpha$

Synopsis Diffusion - II [NATH ET AL., 2008]

Finding ODI synopses and their fusion functions is the main difficulty. We look at two cases, one trivial and the other not so trivial!

Example 1: ODI synopsis for MAX. Let X be the variable.

Synopsis : X (the number itself)

$$SG() = X_i$$

$SF(X_i, X_j) =$ The larger of X_i, X_j

$$SE(X_i) = X_i$$

Synopsis Diffusion - II [NATH ET AL., 2008]

Example 2: COUNT (Using Flajolet-Martin algorithm [FLAJOLET & MARTIN, 1985])

[A small detour] **Flajolet-Martin algorithm**

Probabilistic counting of the number of distinct elements in a multiset.

Assume a uniform hash function

$$\text{hash} : \text{Data} \rightarrow \text{Bitmap} : \{0, 1\}^L,$$

where L is a very small number (≈ 32).

Define $b(y, k) = k^{\text{th}}$ bit in binary representation of y . Define

$$\rho(y) = \begin{cases} \min_{k \geq 0} b(y, k) \neq 0 & \text{if } y > 0 \\ L & \text{if } y = 0 \end{cases}$$

$\rho(y)$ indicates the position of the least significant 1-bit in y .

Synopsis Diffusion - II [NATH ET AL., 2008]

Example 2: COUNT (Using Flajolet-Martin algorithm [FLAJOLET & MARTIN, 1985])

[A small detour] **Flajolet-Martin algorithm**

Let M be the multi-set, and $\text{Bitmap}[L]$ be a bit vector. Let R be the index (counting from 0) of first 0-bit of $\text{Bitmap}[\]$.

Initialization: $\text{Bitmap}[] = \{0\}$

for all $x \in M$ do

 Set $\text{Bitmap}[p(\text{hash}(x))]$

return $2^R/0.77351$

1 1 1 1 1 0 0 1 1 0

hash(x)	1...	01...	001...	0001...
p(hash(x))	1/2	1/4	1/8	1/16

In general, $p(0^k 1 \dots) = 2^{-k-1}$ for uniformly distributed hash(x).

Synopsis Diffusion - II [NATH ET AL., 2008]

Example 2: COUNT (Using Flajolet-Martin algorithm [FLAJOLET & MARTIN, 1985])

Synopsis : A bit vector V of length $k = \log(n)$

$SG() = \text{Set } V[\rho(\text{hash}(x))]$

$SF(V, V') = \text{Bitwise } V \text{ OR } V'$

$SE(V) = 2^i / 0.77351$

where i is the index (counting from 0) of least-significant 0-bit in V

Improving the accuracy of the algorithm:

$$j = \frac{i_1 + i_2 + \dots + i_m}{m}$$

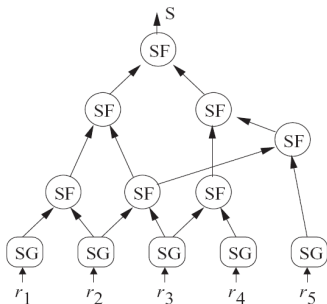
$SE(V) = 2^j / 0.77351$

Synopsis Diffusion - III

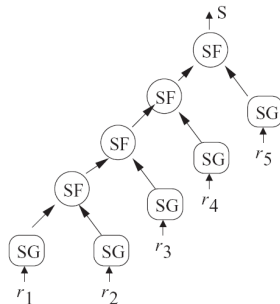
[NATH ET AL., 2008]

ODI Correctness Test

A synopsis diffusion algorithm is ODI-correct if SF and SG are order and duplicate-insensitive functions.



(a) Aggregation DAG



(b) Canonical left-deep tree

Synopsis Diffusion - III [NATH ET AL., 2008]

ODI Correctness Test

A synopsis diffusion algorithm is ODI-correct if SF and SG are order and duplicate-insensitive functions.

Define a projection operator Π_q : Multiset of sensor readings \rightarrow ordered set of values.

- 1 SG preserves duplicates:
 $\forall r1, r2 \in R : \Pi_q(\{r1\}) = \Pi_q(\{r2\}) \Rightarrow SG(r1) = SG(r2)$. The same synopsis is generated for all duplicates.
- 2 SF is commutative: $\forall s1, s2 \in S : SF(s1, s2) = SF(s2, s1)$.
- 3 SF is associative:
 $\forall s1, s2, s3 \in S : SF(s1, SF(s2, s3)) = SF(SF(s1, s2), s3)$.
- 4 SF is idempotent: $\forall s \in S : SF(s, s) = s$.

Wireless Controller Area Network (CAN) Based

- Prioritized MAC using Bit-wise arbitration [MOK ET AL., 1979]
 - ▶ Used in Bosch's CAN 2.0
 - ▶ Extended to wireless channel [ANDERSSON ET AL., 2006; YOU ET AL., 2003]

0	1	1	0
1	0	1	0
1	1	0	0

- The highest priority packet gets transmitted first
- Compute MAX by using data as the priority
- Compute MIN by using complement of data as the priority
- Excellent time-complexity for MAX/ MIN in single broadcast domains
- Limitations for other aggregates

Outline

- 1 Data Aggregation
- 2 Data-Centric Networking
 - Basics of data-centric networking
 - Directed diffusion
- 3 Data Dissemination
 - Flooding
 - Multicast
- 4 Data Aggregation Systems
 - TAG
 - Synopsis Diffusion
 - Wireless CAN
- 5 Conclusion

Open problems

- Joint aggregation and scheduling problem
 - ▶ Spatio-temporal optimization
- Multi-query optimization
- Correlated source coding, compressed sensing

Messages to take home

- Data-aggregation leads to bandwidth and energy efficiency
- Data-centric, and not address-centric, networking paradigm
- Pure flooding is wasteful
- Multicast tree with large node degrees is not optimal

Thank You.