An Energy-Efficient Distributed Algorithm for Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks
Yingshu Li, Longjiang Guo, and Sushil K. Prasad
Department of Computer Science, Georgia State University, USA
yli@cs.gsu.edu

Outline
- Introduction
  - Sensor Networks
  - Data Aggregation
  - Problem Definition
  - Our Contribution
- Related Work
- Clu-DDAS Algorithm
  - Construct a Clu-DAT
  - Distributed Aggregation Scheduling
  - Theoretical Analysis
  - Adaptive Clu-DDAS
- Simulations
- Conclusion

Introduction
- Wireless Sensor Networks
  - Ad hoc networks, no infrastructure.
  - Energy conservation is a primary concern.
  - Communication consumes more energy than computation.
  - Densely deployed to a large region.
  - Network topology may change often.

Introduction (Cont.)
- Data aggregation is a popular operation
  - E.g. Max, Min, Sum, Avg, and etc.
  - In-network data processing.
  - Try to reduce latency by designing a reasonable schedule.
  - Communication collision is a main reason for large latency.
  - Collision problem left to the MAC layer, not efficient

Introduction (Cont.)
- Minimum-Latency Aggregation Scheduling problem:
  - Seek a collision free transmission schedule for data aggregation of all sensors, aiming at minimizing latency and minimizing energy consumption considering dynamic topologies.
  - Centralized algorithms: not practically applicable to WSNs.
  - Distributed algorithms: conserve more energy.
Problem Definition

In (a), (b) and (c), $u \rightarrow v$ and $x \rightarrow y$ are conflicting transmission schedules. In (d), $u \rightarrow v$ and $x \rightarrow y$ are not conflicting transmission schedules, despite a collision occurs at node $w$.

Problem Definition (Cont.)

A data aggregation schedule is a sequence of transmission schedule sets $\{SH_i, SH_2, \ldots, SH_l\}$, where $SH_i$ $(1 \leq i \leq l)$ is a transmission schedule set satisfying the following conditions:

1) $\forall i \neq j, \text{Sender}(SH_i) \cap \text{Sender}(SH_j) = \emptyset$;
2) $\bigcup_{i=1}^{l} \text{Sender}(SH_i) = V - \{r\}$, All sensor nodes in $V$ are organized as a data aggregation tree, where $r$ is the root of the data aggregation tree, $l$ is called the data aggregation latency.
3) Data are aggregated from $\text{Sender}(SH_k)$ to $V - \bigcup_{k=1}^{l} \text{Sender}(SH_k)$ in time-slot $k$, for all $k = 1, \ldots, l$ and all the data are aggregated to $r$ in $l$ time-slots.

Our Contribution

An energy-efficient distributed algorithm: Clu-DDAS.

Clu-DDAS constructs a novel Cluster-based Data Aggregation Tree (Clu-DAT) which is different from the commonly used CDS-based or MIS-based aggregation trees.

The aggregation latency is $4R^* + 2\Delta - 2$ which is smaller than those presented in [Xu et al, FOWANC09] and [Yu et al, INFOCOM09] for typical scenarios.

Clu-DDAS outperforms the previously best distributed algorithm DAS [Xu et al, FOWANC09] and the best centralized algorithm E-PAS [Wan et al, MobiHoc09] on energy consumption.

Outline

- Introduction
  - Sensor Networks
  - Data Aggregation
  - Problem Definition
  - Our Contribution
- Related Work
- Our Algorithm-Clu-DDAS
  - Construct a Clu-DAT
  - Distributed Aggregation Scheduling
  - Theoretical Analysis
  - Adaptive Clu-DDAS
- Simulations
- Conclusion
Related Work

Centralized Algorithms
- Chen et al. MSN05
  - MLAS problem is NP-hard.
  - Latency bound of $(\Delta - 1)R$
- Huang et al. INFOCOM07
  - Latency bound of $23R + \Delta - 18$
- Wan et al. MobiHoc09
  - Three algorithms: SAS, PAS, EPAS.
  - Latency bounds of $15R + \Delta - 4$, $2R + O(\log R) + \Delta$, and $(1 + O(\log R / \sqrt{3} R))R + \Delta$.

Related Work (Cont.)

Existing problems of centralized algorithms
- Aiming at latency.
- Not consider energy consumption in WSNs.
- Consuming lots of energy, especially when network topology changes often.

Related Work (Cont.)

Distributed Algorithms
- Yu et al. INFOCOM09
  - Latency bound of $24D + 6\Delta + 16$, where $D$ is the network diameter.
- Xu et al. FOWANC09
  - Latency bound of $16R' + \Delta - 14$, where $R'$ is the inferior network radius which satisfies $R' \leq R \leq D \leq 2R'$

Related Work (Cont.)

Existing problems of distributed algorithms
- Aiming at latency.
- Not consider energy consumption in WSNs.
- Latencies are still large.

Outline

- Introduction
  - Sensor Networks
  - Data Aggregation
  - Problem Definition
  - Our Contribution
- Related Work
- Our Algorithm - Clu-DDAS
  - Construct a Clu-DAT
  - Distributed Aggregation Scheduling
  - Theoretical Analysis
  - Adaptive Clu-DDAS
- Simulations
- Conclusion

Clu-DDAS

- Clu-DDAS
  - Two phase algorithm
    - Construct an aggregation tree (Clu-DAT) in a distributed manner.
    - Distributed aggregation scheduling
Construct a Clu-DAT

Network Center

Construct a Clu-DAT (Cont.)
Construct a Clu-DAT

14 and 16 have the largest number of neighbors and the smallest ID among their neighbors.

19 has the largest number of neighbors and the smallest ID among their neighbors.

All neighbors do not belong to any cluster?
Construct a Clu-DAT

(v', clu-head-id).level < v'.level < u'.level
\( u \) is sender, \( v \) is receiver

Construct a Clu-DAT

\[ Clu-DDAS \]

- **Clu-DDAS**
  - two phase algorithm
    - Construct an aggregation tree (Clu-DAT) in a distributed manner.
    - Distributed Aggregation Scheduling

\[ Distributed \; Aggregation \; Scheduling \]

- Preliminary

**Definition 8:** Competitor and Competitor Set. For a node \( u \), a node \( v \) is called a Competitor of \( u \) if \( v \) cannot send data while \( u \) is sending data due to collision. The set of all the competitors of \( u \) in a Clu-DAT \( T \) is called \( u \)'s Competitor Set [3] with respect to \( T \), denoted by
\[
CS(u) = Nbr(p(u)) \cup \bigcup_{v \in Nbr(u)} Ch(v) \setminus \{ p(u), u \},
\]
where \( p(v) \), \( Ch(u) \), and \( Nbr(v) \) are \( v \)'s parent in \( T \), \( u \)'s children set in \( T \), and \( v \)'s 1-hop neighbor set, respectively.
Distributed Aggregation Scheduling

Preliminary

Scheduling algorithm allocates a time slot \( t_s(u) \), for each node \( u \).

In the process of data aggregation, each node \( u \) sends data to its parent \( p(u) \) in the Clu-DAT in its assigned time slot after \( u \) has collected data from all of its children in the Clu-DAT.

Node color decides scheduling priority.

The descending order of the priorities is: GREEN, WHITE, BLACK, BLUE and YELLOW.

For any two nodes \( u \) and \( v \), we say \( u \)'s priority is lower than \( v \)'s priority, i.e. 
\[
(u.\text{color}, u.\text{ID}) < (v.\text{color}, v.\text{ID}) \text{ if } \begin{cases} u.\text{color} < v.\text{color} \text{ or } \\ u.\text{color} = v.\text{color} \text{ and } u.\text{ID} < v.\text{ID} \end{cases}
\]

An Example

Once a node has got its time slot, its schedule state DONE is set to True. Logically, such a node is removed from the Clu-DAT, resulting in some new leaves.

This process is repeated until \( v_c \) is the only node left.
16’s priority is larger than 18’s because 16 is BLACK.
Theoretical Results

**Definition 6:** Inferior Network Radius. The Inferior Network Radius of $G$, denoted by $R' = \max\{|d_G(u, v)| \mid u \in V\}$, which is the maximum distance between $v_s$ and any other node $u$ in $G$.

**Definition 8:** Network Radius. The Network Radius of $G$, denoted by $R$, is the maximum distance between the sink $s$ and any other node in $G$.

**Definition 9:** Network Diameter. The Network Diameter of $G$, denoted by $D = \max\{|d_G(u, v)| \mid u \in V \mid v \in V\}$, is the maximum distance between any two nodes in $G$.

Theoretical Results

For example, in Fig. 2, $R' = 4$, $R = 7$, and $D = 8$.

Theoretical Results

**Proposition 11:** $R' \leq R \leq D \leq 2R'$.

**Theorem 10:** The time latency of Clu-DDAS is at most $4R' + 2\Delta - 2$.


---

### Adaptive Clu-DDAS

- **Two phase**
  - Maintenance of Clu-DAT
    - Two cases
    - Node joining
    - Node failure
  - Basic idea
    - Maintaining the property of current Clu-DAT
    - Joining nodes find their parents
    - Failure nodes are deleted from current Clu-DAT
    - Nodes that lose their parents find new ones
  - Adaptive Scheduling
    - All the nodes who change their parents are marked “renewed”
    - Any node who is an ancestor of a node who changes its parent is marked “renewed”
    - All the “renewed” nodes run distributed aggregation scheduling
    - New schedules generated

---

### Theoretical Results

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Algo.</th>
<th>Bound of latency</th>
<th>Estimated upper bound of latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>SDAT</td>
<td>$\Delta(R - 1)$</td>
<td>$\Delta(2R' - 1)$</td>
</tr>
<tr>
<td>[4]</td>
<td>Scott</td>
<td>$2R + \Delta - 4$</td>
<td>$16R' + 4\Delta - 18$</td>
</tr>
<tr>
<td>[1]</td>
<td>SAS</td>
<td>$15R + \Delta - 4$</td>
<td>$30R' + 4\Delta - 4$</td>
</tr>
<tr>
<td>[1]</td>
<td>ALAS</td>
<td>$2R + O(\log R) + \Delta$</td>
<td>$4R' + O(\log(2R')) + \Delta$</td>
</tr>
<tr>
<td>[1]</td>
<td>E-PAS</td>
<td>$(1 + O(1))R + \Delta$</td>
<td>$(1 + O(1))R' + \Delta$</td>
</tr>
<tr>
<td>[3]</td>
<td>DAS</td>
<td>$24D + r_{14} - 16$</td>
<td>$48R' + 2\Delta - 16$</td>
</tr>
<tr>
<td>[2]</td>
<td>Das</td>
<td>$16R' + \Delta - 14$</td>
<td>$16R' + \Delta - 14$</td>
</tr>
<tr>
<td>This paper</td>
<td>Clu-DDAS</td>
<td>$4R' + 2\Delta - 2$</td>
<td>$4R' + 2\Delta - 2$</td>
</tr>
</tbody>
</table>

Centralized: [1], [4], [5]; Distributed: [2], [3]
Outline

- Introduction
  - Sensor Networks
  - Data Aggregation
  - Problem Definition
  - Our Contribution
- Related Work
- Our Algorithm-Cllu-DDAS
  - Construct a Clu-DAT
  - Distributed Aggregation Scheduling
  - Theoretical Analysis
  - Adaptive Clu-DDAS
- Simulations
- Conclusion

Simulations

- Setup
  - Randomly and uniformly deploy N sensors into a square region of size 200m \( \times \) 200m.
  - The topology simulator takes in an input of \( R \), and a sensor transmission range of \( r \).
  - All the sensors have the same transmission range \( r \).
  - The sink is always the node with ID 0. Its position is random.
- Objective
  - Evaluation aspects: average aggregation latency and total messages.
  - Compare with SAS, PAS and E-PAS (the best centralized algorithm) proposed in [1], and DAS (the best distributed algorithm) proposed in [2].

Simulations (Cont.)

- Construct an aggregation tree (Clu-DAT).
- A distributed scheduling algorithm based on a Clu-DAT with a latency bound of \( 4R' + 2\lambda - 2 \), where \( \lambda \) is the maximum degree and \( R' \) is the inferior network radius which is smaller than the network radius \( R \).
- The simulation results indicate that Clu-DDAS has comparable latency as the previously best centralized algorithm E-PAS, while Clu-DDAS consumes much less energy.
- Clu-DDAS outperforms the previously best distributed algorithm DAS on both latency and energy consumption.
- An adaptive strategy for updating the schedule to accommodate dynamic network topology.