DATA PRESERVATION IN INTERMITTENTLY CONNECTTED SENSOR NETWORK WITH DATA PRIORITY

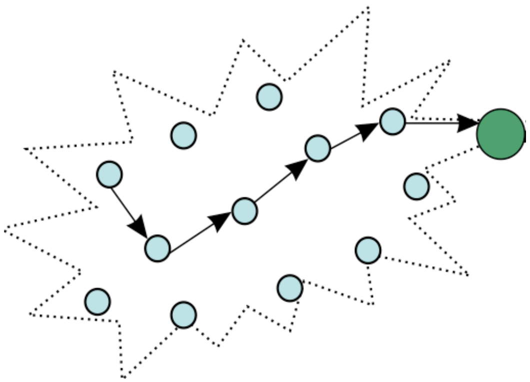
Xinyu Xue[1], Xiang Hou[1], Bin Tang[2], and Rajiv Bagai[1]

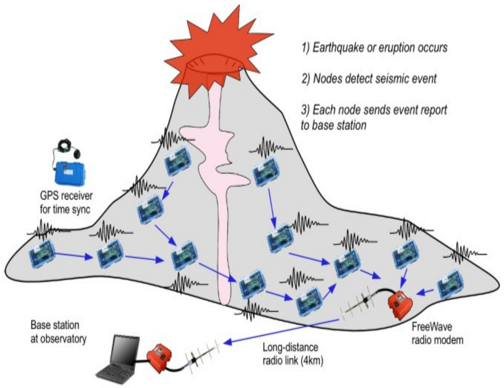
[1]Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS

[2]Department of Computer Science, Azusa Pacific University, Azusa, CA

Intermittently Connected Sensor Networks

- Underwater/ocean seismic sensor networks, volcano eruption/glacial melting monitoring
- Not feasible to install base station in field
- Data generated and stored in the network, periodically uploaded via data mules or satellite links
- Data Priority: data generated may have different importance (seismic, infrasonic, temperature)

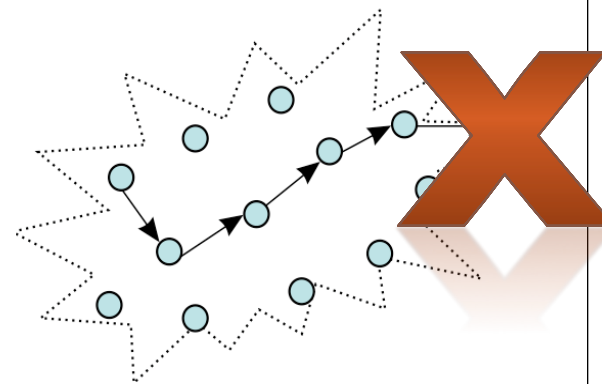


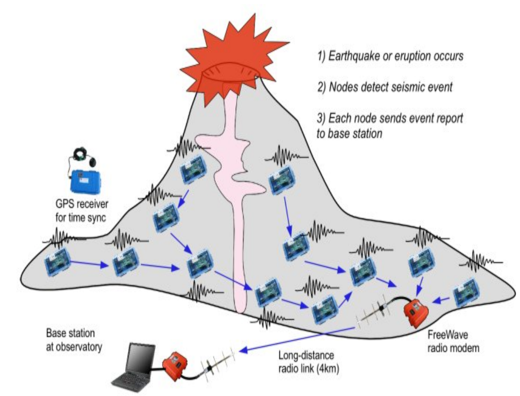


Source: http://fiji.eecs.harvard.edu/Volcano

Intermittently Connected Sensor Networks

- Underwater/ocean seismic sensor networks, volcano eruption/glacial melting monitoring
- Not feasible to install base station in field
- Data generated and stored in the network, periodically uploaded via data mules or satellite links
- Data Priority: data generated may have different importance (seismic, infrasonic, temperature)

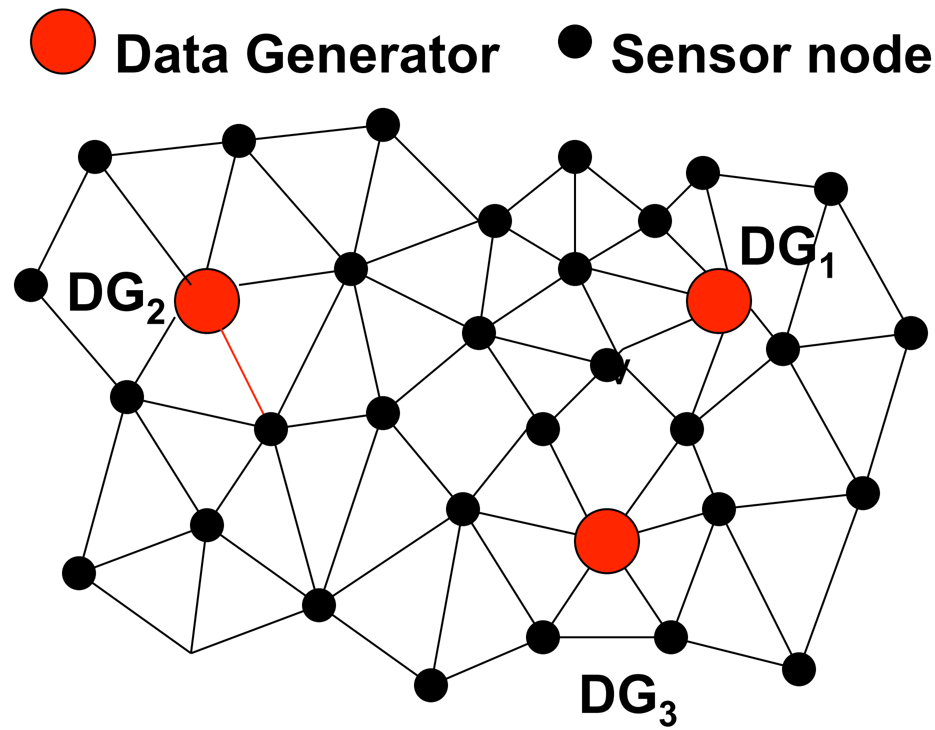




Source: http://fiji.eecs.harvard.edu/Volcano

Data Preservation In Intermittently Connected Sensor Networks

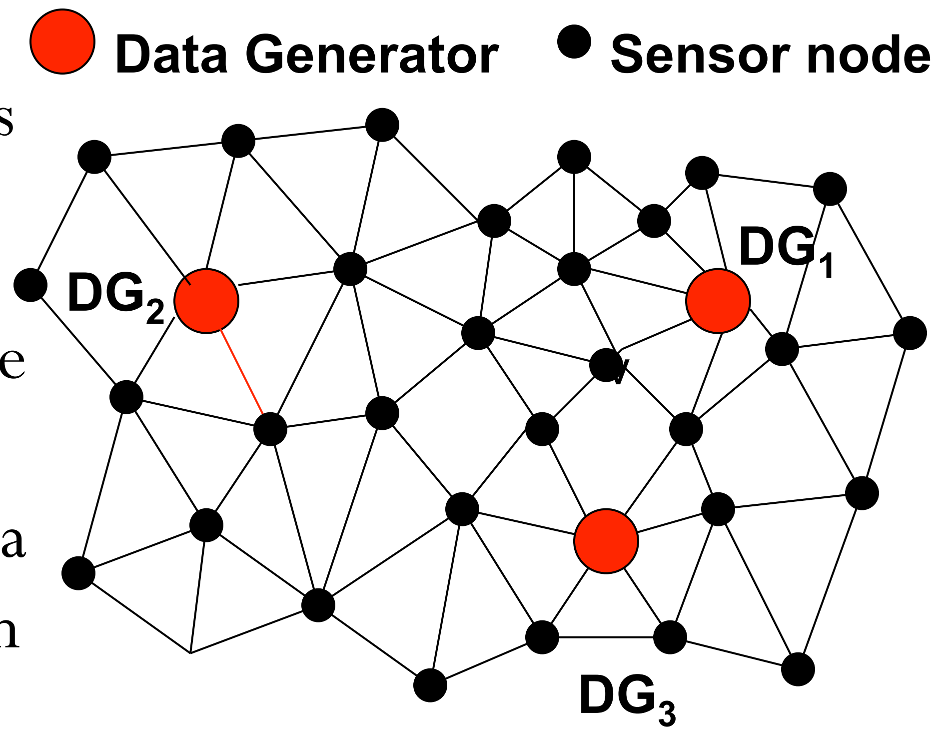
- Non-uniform data generation and limited storage capacity
- Data generators (DGs): storage-depleted
- Data preservation: offload overflow data from DGs to nodes with available storage
- Data from different DGs are of different importance



priority of $\mathbf{DG_1}$ ≥ priority of $\mathbf{DG_2}$ > priority of $\mathbf{DG_3}$

Challenge in Data Preservation

- Limited battery power
- Data preservation consumes battery power
- When not all the data can be preserved inside the network, how to ensure data preservation with maximum total priorities – data preservation with data priorities (DPP)



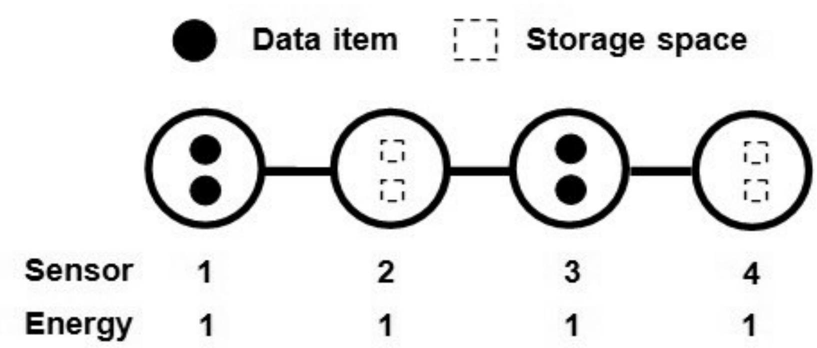
priority of **DG_1** ≥ priority of **DG_2** > priority of **DG_3**

Data Preservation With Priority (DPP)

- Sensor network graph $G(V, E)$
- Data generators: DG_1, DG_2, …, DG_k
  - v_i - DG_i' priority
  - d_i - number of overflow data DG_i needs to offload
- m_i - initial available free storage space of node i
- E_i - initial energy level of node i
- Sending/relaying/receiving a data item each costs 1 unit of energy

Objective of DPP

- Select a subset of data items to offload to maximize their total priorities



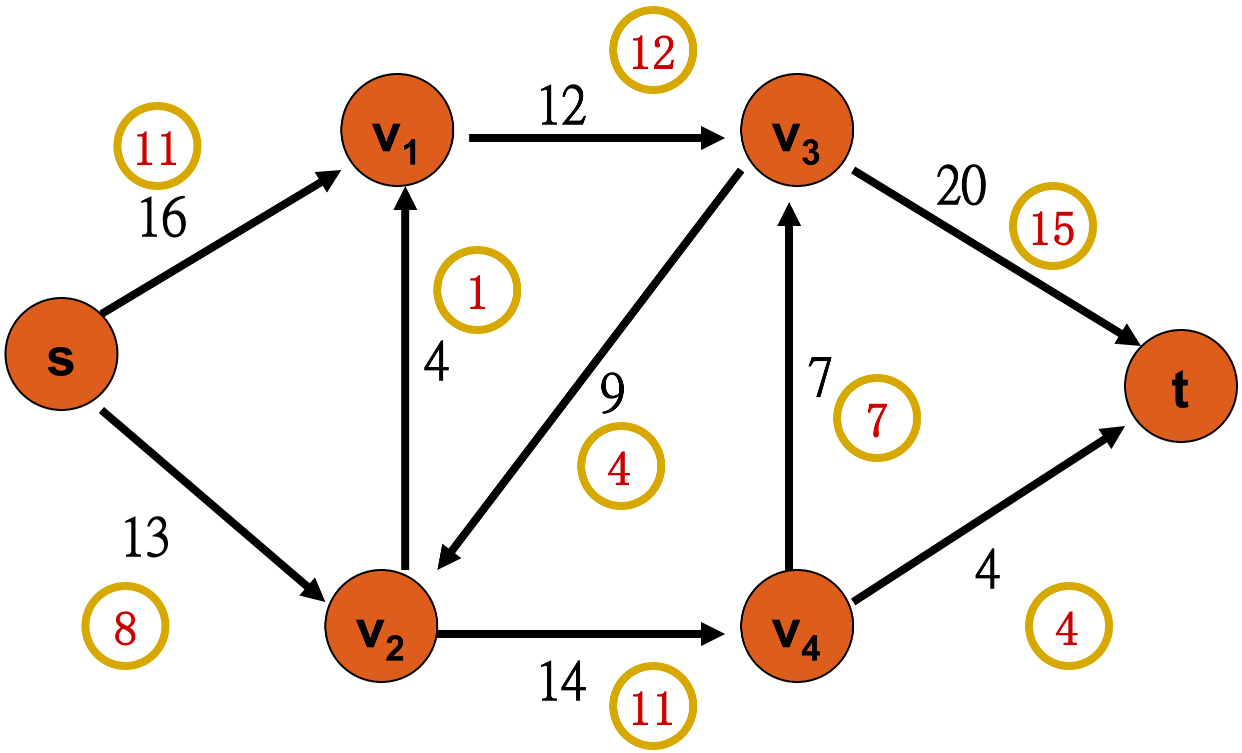
Priority of node 1: 2
Priority of node 3: 1

Total preserved priorities: 3

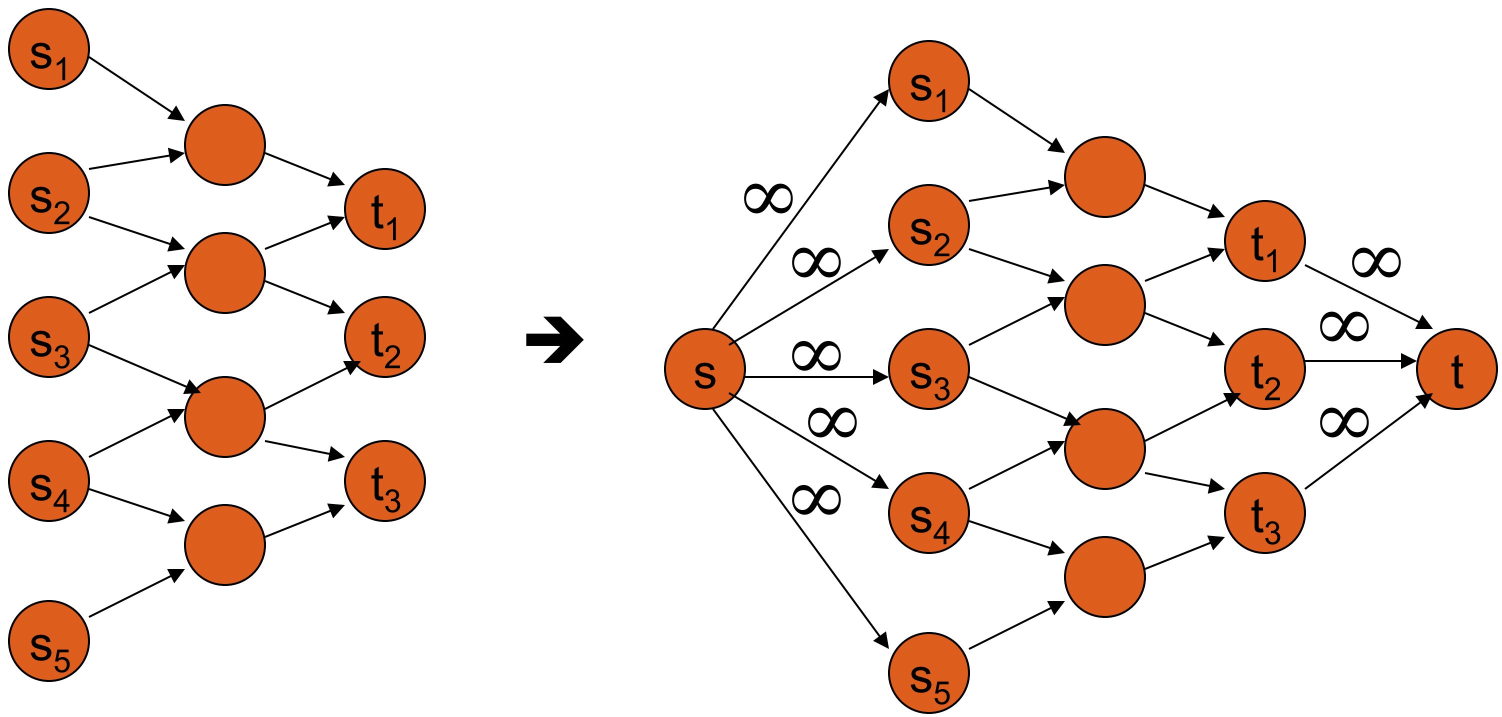
Fig.1. Illustration of the DPP problem.

Maximum Flow Problem

Given a flow network *G* with source *s* and sink *t*, find maximum amount of flow from *s* to *t*



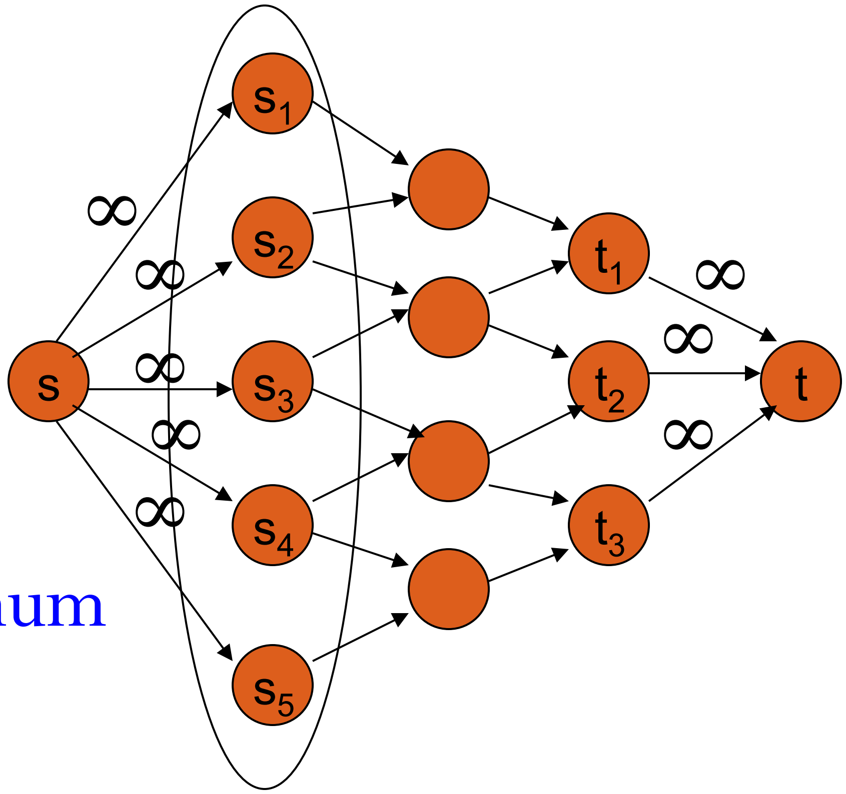
Multiple Sources and/or Sinks



Maximum Weighted Flow (MWF)

- v_i: weight of one net flow out of $s_i \in S$

$$v_1 \geq v_2 \geq \ldots \geq v_5$$



Find a flow with maximum total weight from s to t

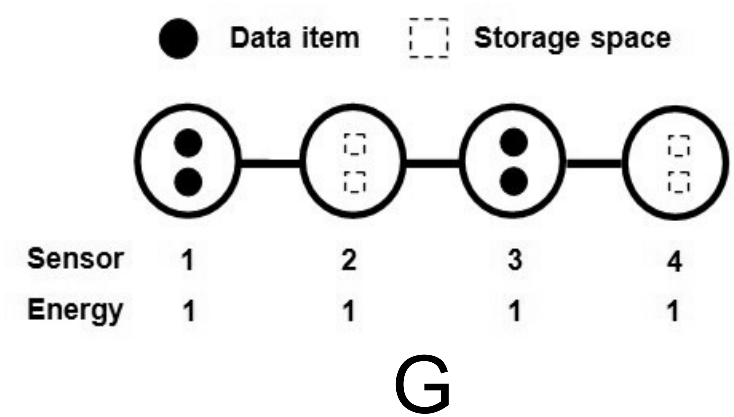
Priority-based Algorithm (PBA) for MWF

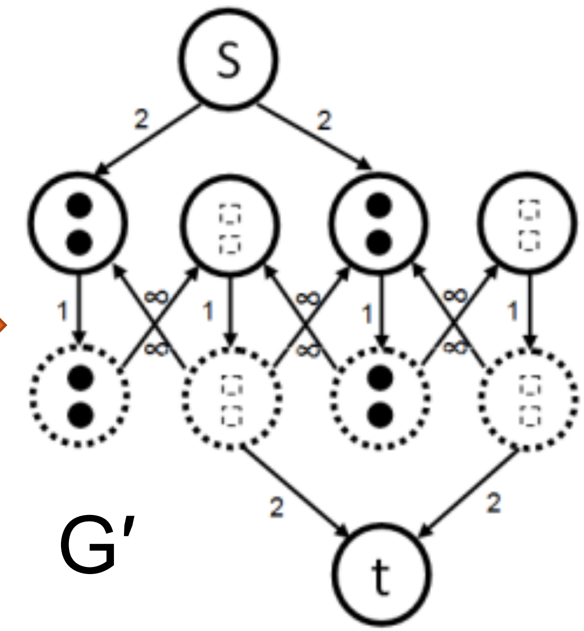
Find maximum flow (using Edmonds-Karp) in non-ascending order of source node's priority

- Optimality proof of PBA (omitted)
- Maximum weighted flow is a maximum flow, but not vice versa
- Time complexity: $O(knm^2)$

Optimal Algorithm for DPP

Transform G into G'





Priority of node 1: 2
Priority of node 3: 1

Total preserved priorities: 3

PBA on *G′* is an optimal algorithm for DPP on *G.*

A Heuristic Algorithm of DPP

- Offload data in non-ascending order of DG's priority:

  for each DG

  while (It can still off a data item from it to a non-DG node)

  Offload it to the closest non-DG node

- Time complexity: O(km + k<u>d</u>n), <u>d</u> is average number of data items of each DG

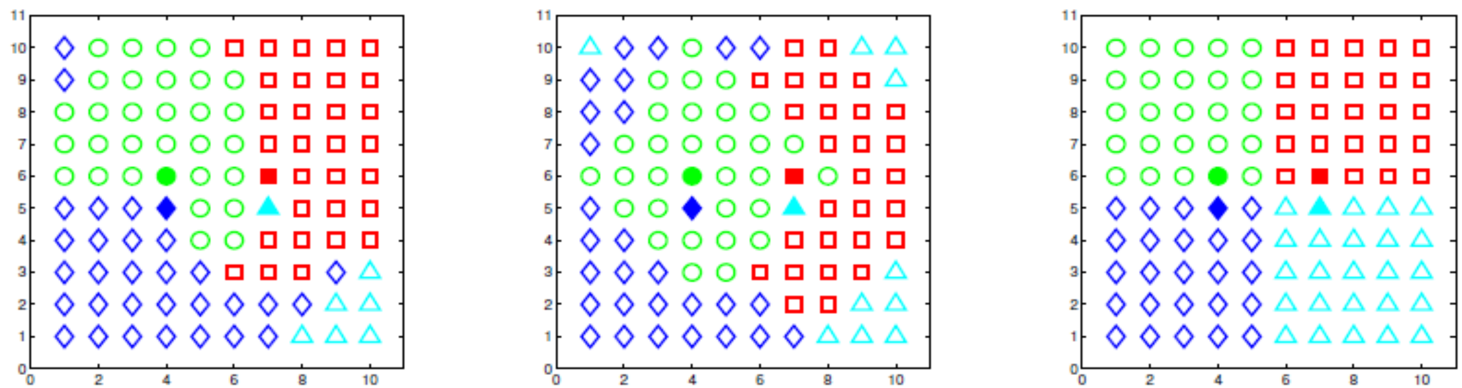
Performance Evaluation

- **Visual Performance Comparison**
  - Grid network: 10 × 10, 20 × 20
  - Initial energy level: 30 units

	Shape	Coordinate	Priority
DG_1	●	(4, 6)	8
DG_2	■	(7, 6)	6
DG_3	◆	(4, 5)	4
DG_4	▲	(7, 5)	2

  - Each DG has 30 data items to offload (50 data items in 20 × 20 grid)

Data Preservation Blocked by Storage Constraint



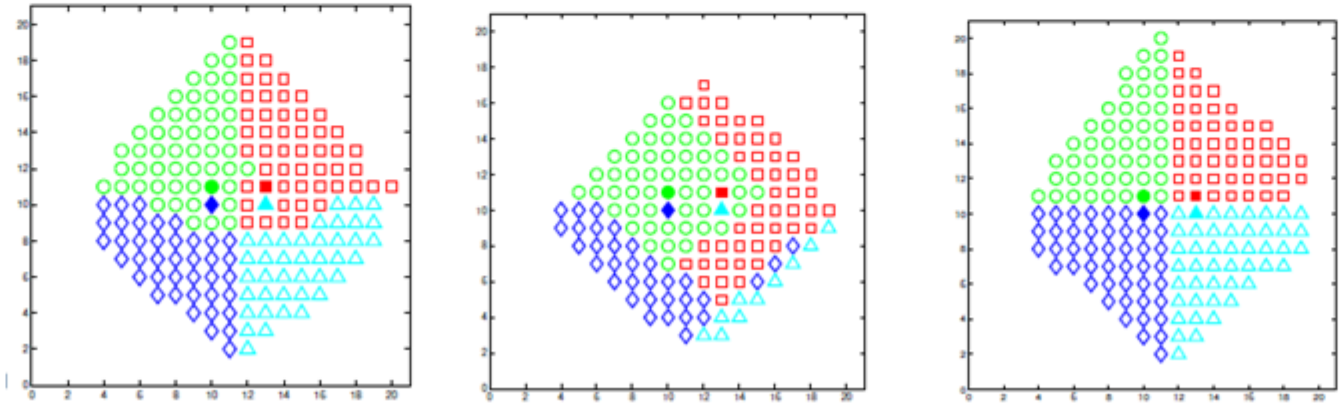
(a) Optimal. (b) Heuristic. (c) Edmonds-Karp.

Fig. 4. Data Preservation Blocked by Storage Constraint.

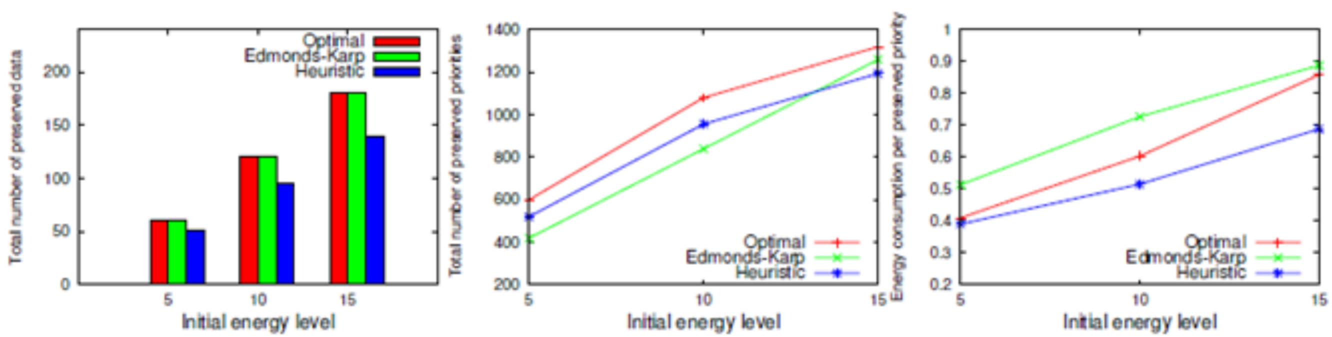
TABLE II
RESULTS OF VISUAL COMPARISON IN FIG. 4.

	Optimal	Heuristic	Edmonds-Karp
Number of Preserved Data	96	96	96
Total Preserved Priority	552	540	480

Data Preservation Blocked by Energy Constraint



(a) Optimal. (b) Heuristic. (c) Edmonds-Karp.



(a) Total number of preserved data. (b) Total number of preserved priorities. (c) Energy consumption per preserved priority.

Push-Relabel Algorithm

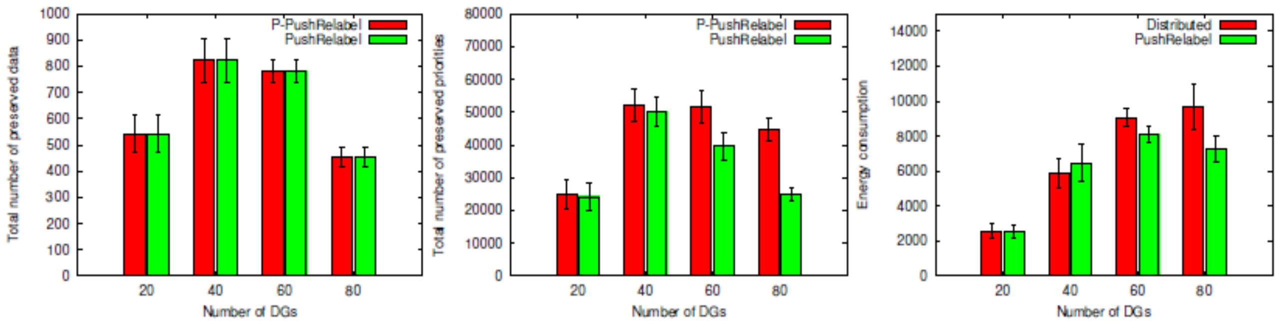
- "active" node - a node with "excess flow"
- "Relabel": increase the height of the "active" node to push excess flow
- "Push": send the excess flow to the neighbors
- Terminates when no "active" nodes left

Distributed Data Preservation with Data Priority

```
Each DG broadcasts its priority to the network;
for each DG in the non-ascending order of its priority
    s pushes maximum allowable data to this DG;
    while (there exists a node u with positive excess)
        Push-Relabel(u);
    end while;
end for
```

The distributed algorithm preserves maximum total priority. It runs in $O(kn^2)$ time and uses $O(n^2m)$ messages.

Distributed Algorithm Comparison



(a) Total number of preserved data.

(b) Total number of preserved priorities.

(c) Total energy consumption.

Fig. 8. Comparing Distributed and PushRelabel.

Conclusions

- Data preservation in sensor networks by considering data priorities (DPP)
- Maximum weighted flow (MWF), generalizing maximum flow problem
- Distributed algorithm

Future Works

- DGs of low priority discard their locally generated data
- General energy model
- Combining data preservation and data retrieving

THANKS!

btang@apu.edu