Self-Organization in Autonomous Sensor/Actuator Networks [SelfOrg]

Dr.-Ing. Falko Dressler

Computer Networks and Communication Systems
Department of Computer Sciences
University of Erlangen-Nürnberg

http://www7.informatik.uni-erlangen.de/~dressler/
dressler@informatik.uni-erlangen.de

Overview

- **Self-Organization**
  Introduction; system management and control; principles and characteristics; natural self-organization; methods and techniques

- **Networking Aspects: Ad Hoc and Sensor Networks**
  Ad hoc and sensor networks; self-organization in sensor networks; evaluation criteria; medium access control; ad hoc routing; data-centric networking; clustering

- **Coordination and Control: Sensor and Actor Networks**
  Sensor and actor networks; coordination and synchronization; in-network operation and control; task and resource allocation

- **Bio-inspired Networking**
  Swarm intelligence; artificial immune system; cellular signaling pathways

Clustering

- Introduction and classification
- k-means and hierarchical clustering
- LEACH and HEED

Clustering

- Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data

- A loose definition of clustering could be ***“the process of organizing objects into groups whose members are similar in some way”***

- A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters

Objectives

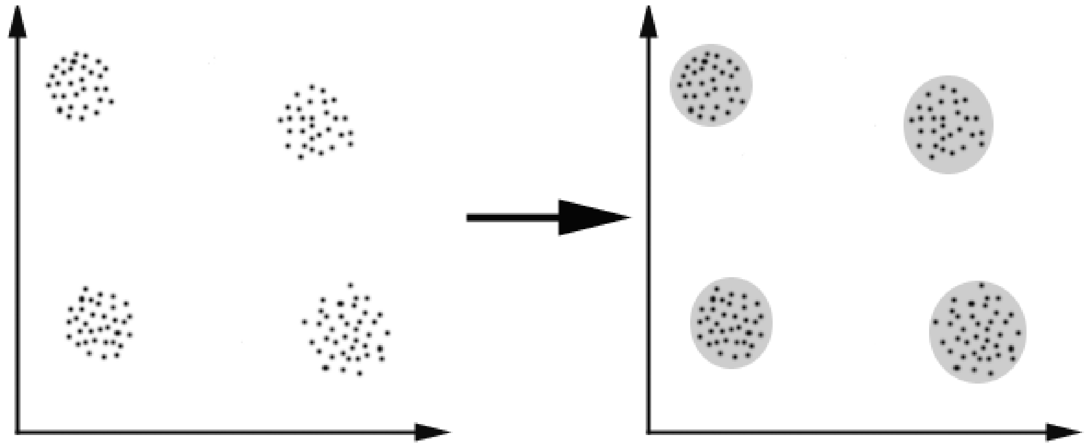
- ***Optimized resource utilization*** - Clustering techniques have been successfully used for time and energy savings. These optimizations essentially reflect the usage of clustering algorithms for task and resource allocation.

- ***Improved scalability*** - As clustering helps to organize large-scale unstructured ad hoc networks in well-defined groups according to application specific requirements, tasks and necessary resources can be distributed in this network in an optimized way.

Classification

- ***Distance-based clustering***: two or more objects belong to the same cluster if they are “close” according to a given *distance* (in this case geometrical distance). The “distance” can stand for any similarity criterion

- ***Conceptual clustering***: two or more objects belong to the same cluster if this one defines a concept common to all that objects, i.e. objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures



Clustering Algorithms

- ***Centralized***
  - If centralized knowledge about all local states can be maintained
    → central (multi-dimensional) optimization process

- ***Distributed / self-organized***
  - Clusters are formed dynamically
  - A cluster head is selected first
  - Usually based on some election algorithm known from distributed systems
  - Membership and resource-management is maintained by the cluster head
    → distributed (multi-dimensional) optimization process

Applications

- General
  - Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
  - Biology: classification of plants and animals given their features;
  - Libraries: book ordering;
  - Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
  - City-planning: identifying groups of houses according to their house type, value and geographical location;
  - Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones;
  - WWW: document classification; clustering weblog data to discover groups of similar access patterns.

- Autonomous Sensor/Actuator Networks
  - Routing optimization
  - Resource and task allocation
  - Energy efficient operation

Clustering Algorithms

- Requirements
  - Scalability
  - Dealing with different types of attributes
  - Discovering clusters with arbitrary shape
  - Minimal requirements for domain knowledge to determine input parameters
  - Ability to deal with noise and outliers
  - Insensitivity to order of input records
  - High dimensionality
  - Interpretability and usability

Clustering Algorithms

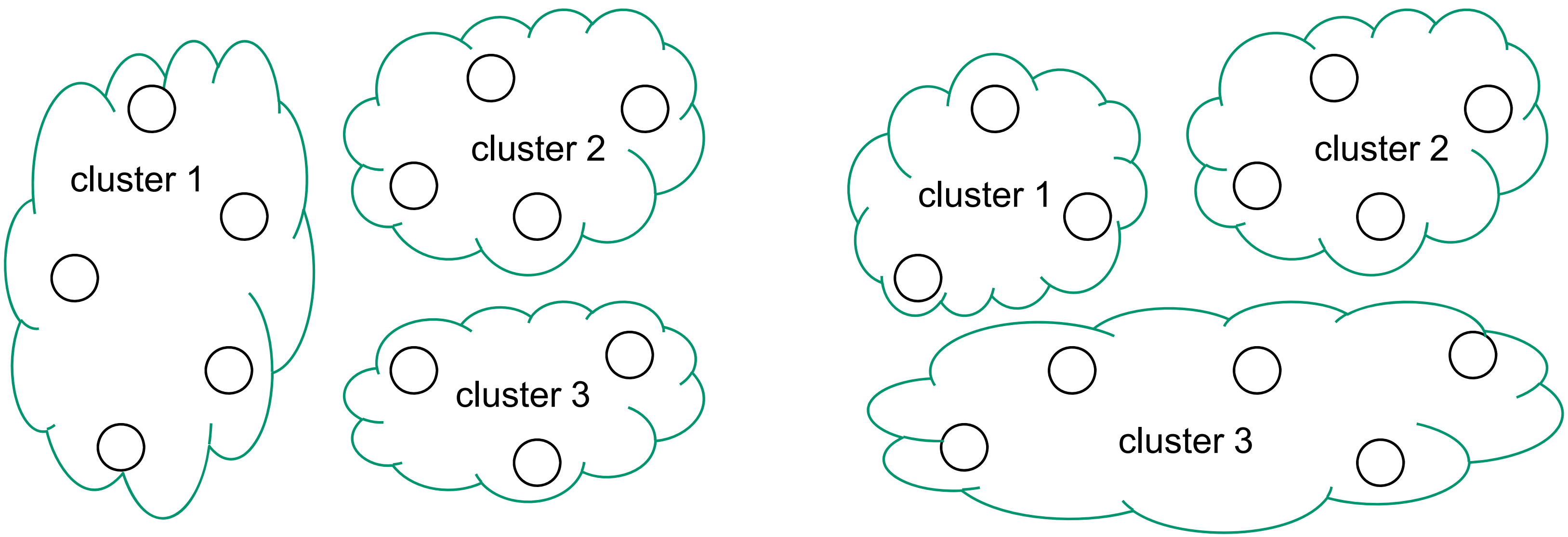
- Problems
  - Current clustering techniques do not address all the requirements adequately (and concurrently)
  - Dealing with large number of dimensions and large number of data items can be problematic because of time complexity
  - The effectiveness of the method depends on the definition of “distance” (for distance-based clustering)
  - If an obvious distance measure doesn’t exist we must “define” it, which is not always easy, especially in multi-dimensional spaces
  - The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways

Clustering Algorithms

- Classification
  - **Exclusive** – every node belongs to exactly one cluster (e.g. ***k-means***)
  - **Overlapping** – nodes may belong to multiple clusters
  - **Hierarchical** – based on the union of multiple clusters (e.g. ***single-linkage clustering***)
  - **Probabilistic** – clustering is based on a probabilistic approach

Clustering Algorithms

- Distance measure
  - The quality of the clustering algorithm depends first on the quality of the distance measure



Clustering variant (a)　　Clustering variant (b)

k-means

- One of the simplest unsupervised learning algorithms
- Main idea
  - Define k centroids, one for each cluster
    - These centroids should be placed in a cunning way because of different location causes different result, so, the better choice is to place them as much as possible far away from each other
  - Take each point belonging to a given data set and associate it to the nearest centroid - when no point is pending, the first step is completed and an early grouping is done
  - Re-calculate k new centroids as barycenters of the clusters resulting from the previous step
    - A new binding has to be done between the same data set points and the nearest new centroid
  - A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done, i.e. the centroids do not move any more

k-means

- The algorithm aims at minimizing an objective function, in this case a squared error function

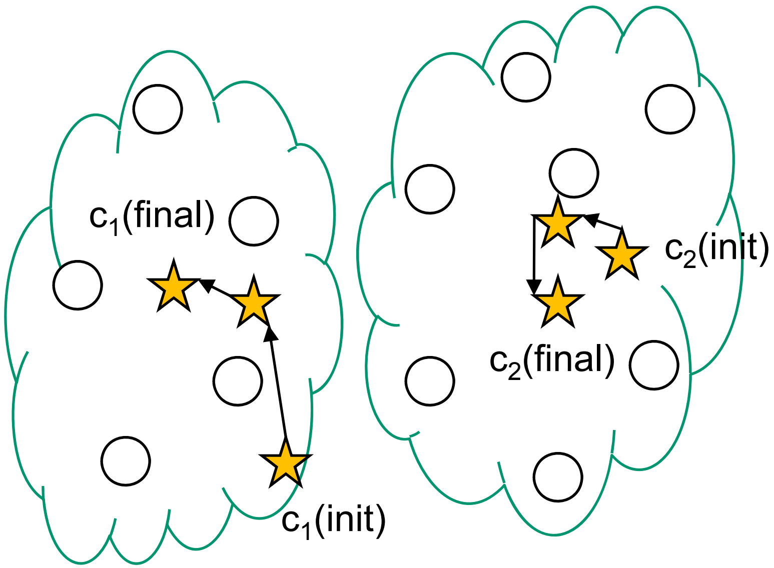
$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

- Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j
- The objective function is an indicator of the distance of the n data points from their respective cluster centers

k-means – algorithm

- Exclusive clustering of n objects into k disjunct clusters

- Initialize centroids c_j (j = 1, 2, …, k), e.g. by randomly choosing the initial positions c_j or by randomly grouping the nodes and calculating the barycenters
- **repeat**
  - Assign each object x_i to the nearest centroid c_j such that $\left\| x_i^{(j)} - c_j \right\|^2$ is minimized
  - Recalculate the centroids c_j as the barycenters of all $x_i^{(j)}$
- **until** centroids c_j have not moved in this iteration



Demo

Hierarchical Clustering Algorithms

- Given a set of N items to be clustered, and an N x N distance (or similarity) matrix, the basic process of hierarchical clustering is this:
  1. Assign each item to a cluster (N items result in N clusters each containing one item); let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain
  2. Find the closest (most similar) pair of clusters and merge them into a single cluster
  3. Compute distances (similarities) between the new cluster and each of the old clusters
  4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N (this results in a complete hierarchical tree; for k clusters you just have to cut the k-1 longest links)

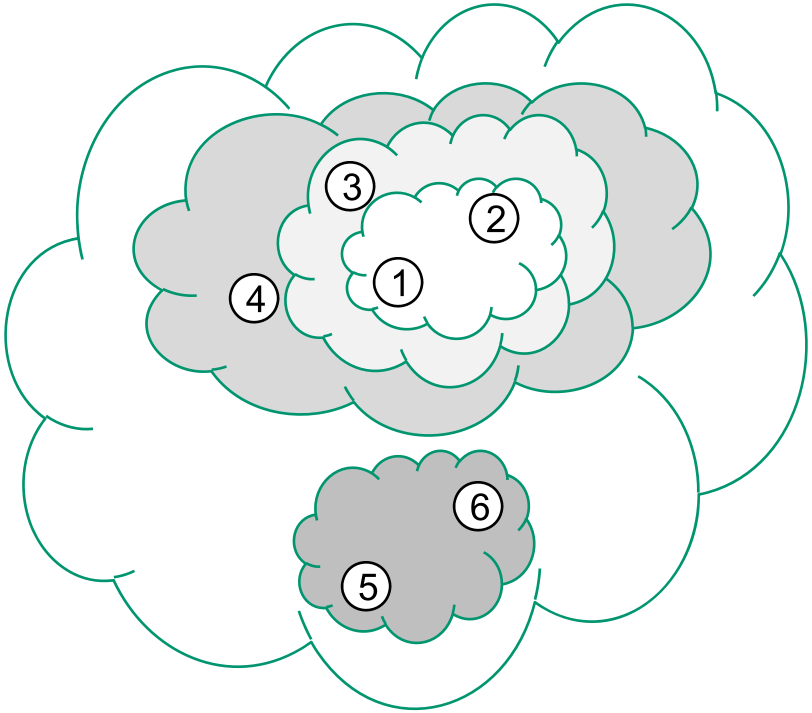
- This kind of hierarchical clustering is called ***agglomerative*** because it merges clusters iteratively

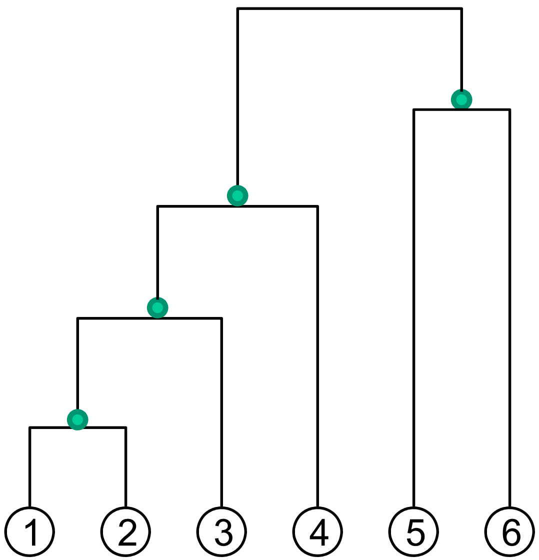
Hierarchical Clustering Algorithms

- Computation of the distances (similarities)
  - In ***single-linkage clustering*** (also called the *minimum* method), we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster
  - In ***complete-linkage clustering*** (also called the *diameter* or *maximum* method), we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster
  - In ***average-linkage clustering***, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster

- Main weaknesses of agglomerative clustering methods:
  - they do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - they can never undo what was done previously

Single-Linkage Clustering





Demo

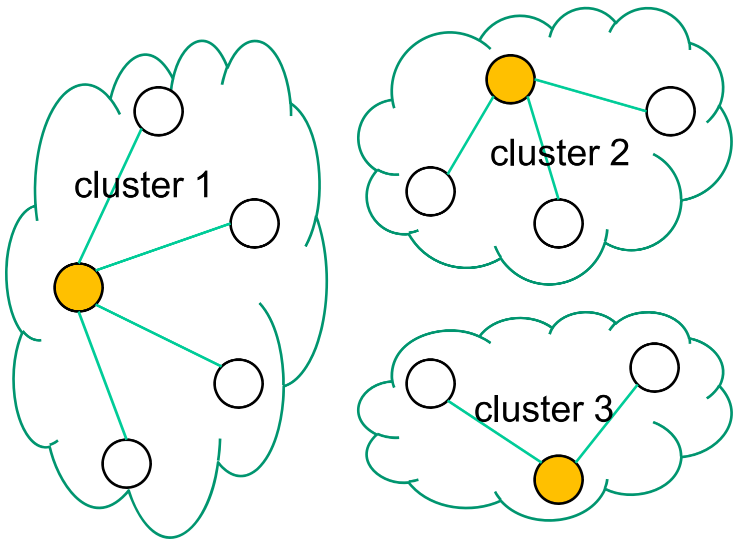
LEACH

- LEACH: Low-Energy Adaptive Clustering Hierarchy

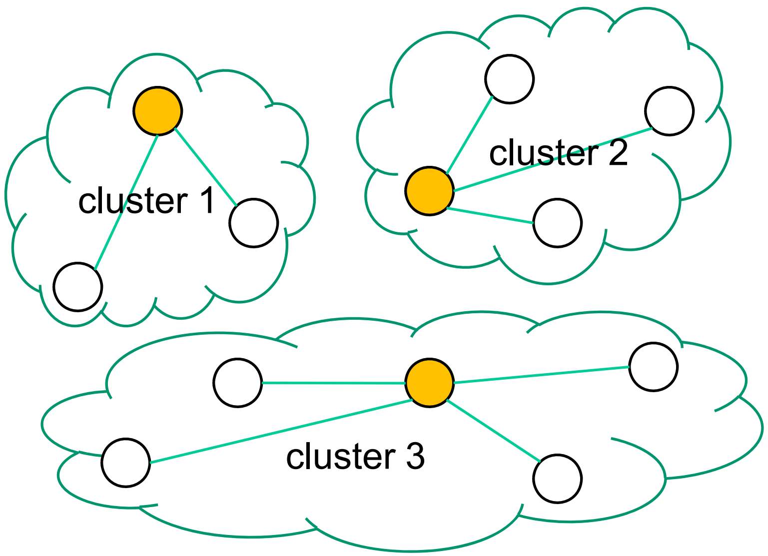
- Capabilities
  - ***Self-organization*** – Self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network. All nodes organize themselves into local clusters, with one node acting as the local base station or cluster-head
  - ***Energy distribution*** – Includes randomized rotation of the high-energy cluster-head position such that it rotates among the various sensors in order to not drain the battery of a single sensor
  - ***Data aggregation*** – Performs local data fusion to “compress” the amount of data being sent from the clusters to the base station, further reducing energy dissipation and enhancing system lifetime

LEACH

- Principles
  - Sensors elect themselves to become cluster-heads at any given time with a certain probability
  - The clusterhead nodes broadcast their status to the other sensors in the network
  - Each sensor node determines to which cluster it wants to belong by choosing the cluster-head that requires the minimum communication energy



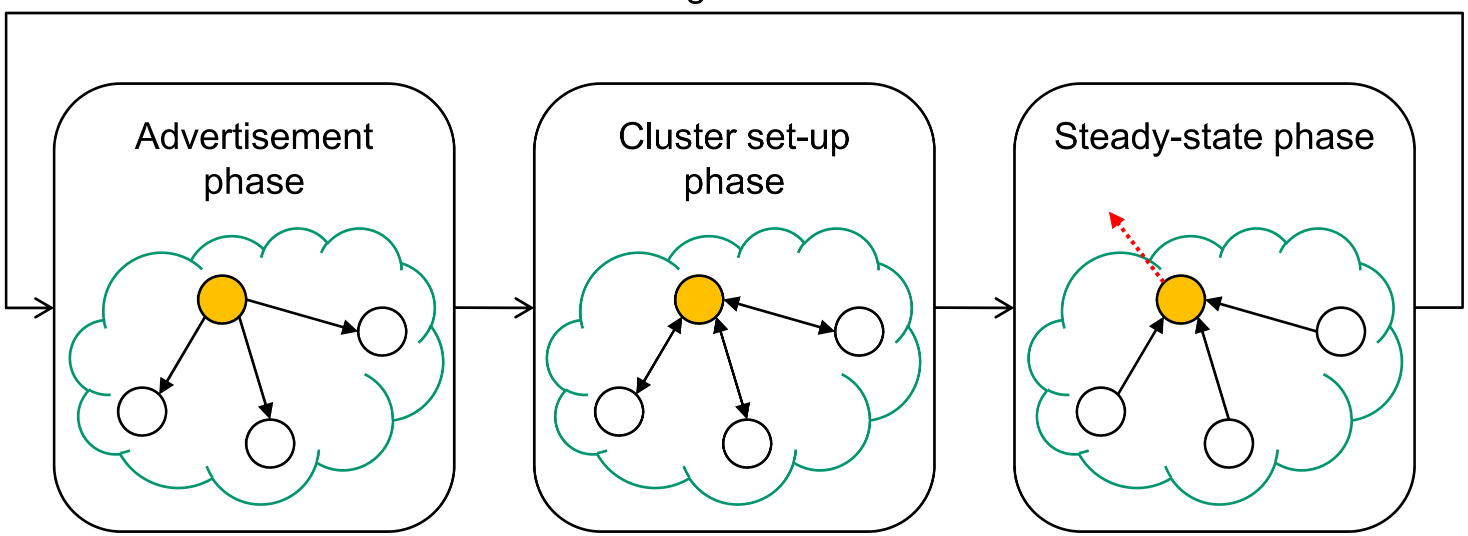
Clustering at time t_1



Clustering at time $t_1 + d$

LEACH

- Algorithm details
  - Operation of LEACH is broken into rounds
  - Cluster is initialized during the advertisement phase
  - Configuration during the set-up phase
  - Data transmission during the steady-state phase



LEACH

- Advertisement phase
  - Each node decides whether or not to become a clusterhead for the current round
    - Based on the suggested percentage of clusterheads for the network (determined a priori), and the number of times the node has been a clusterhead so far
    - The decision is made by the node n choosing a random number between 0 and 1; if the number is less than a threshold $T(n)$, the node becomes a cluster-head for the current round
    - The threshold is set as:

$$T(n) = \begin{cases} \dfrac{P}{1 - P \times \left(r \bmod \dfrac{1}{P} \right)} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

    - where P is the desired percentage of clusterheads (e.g., P = 0.05), r is the current round, and G is the set of nodes that have not been clusterheads in the last 1/P rounds
  - Using this threshold, each node will be a clusterhead at some point within 1/P rounds; the algorithm is reset after 1/P rounds

LEACH

- Clusterhead-Advertisement
  - Each node that has elected itself a cluster-head for the current round broadcasts an advertisement message to the rest of the nodes
  - All cluster-heads transmit their advertisement using the same transmit energy; the non-clusterhead nodes must keep their receivers on during this phase of set-up to hear the advertisements
  - Each non-clusterhead node decides the cluster to which it will belong for this round based on the received signal strength of the advertisement; tiebreaker: randomly chosen cluster-head

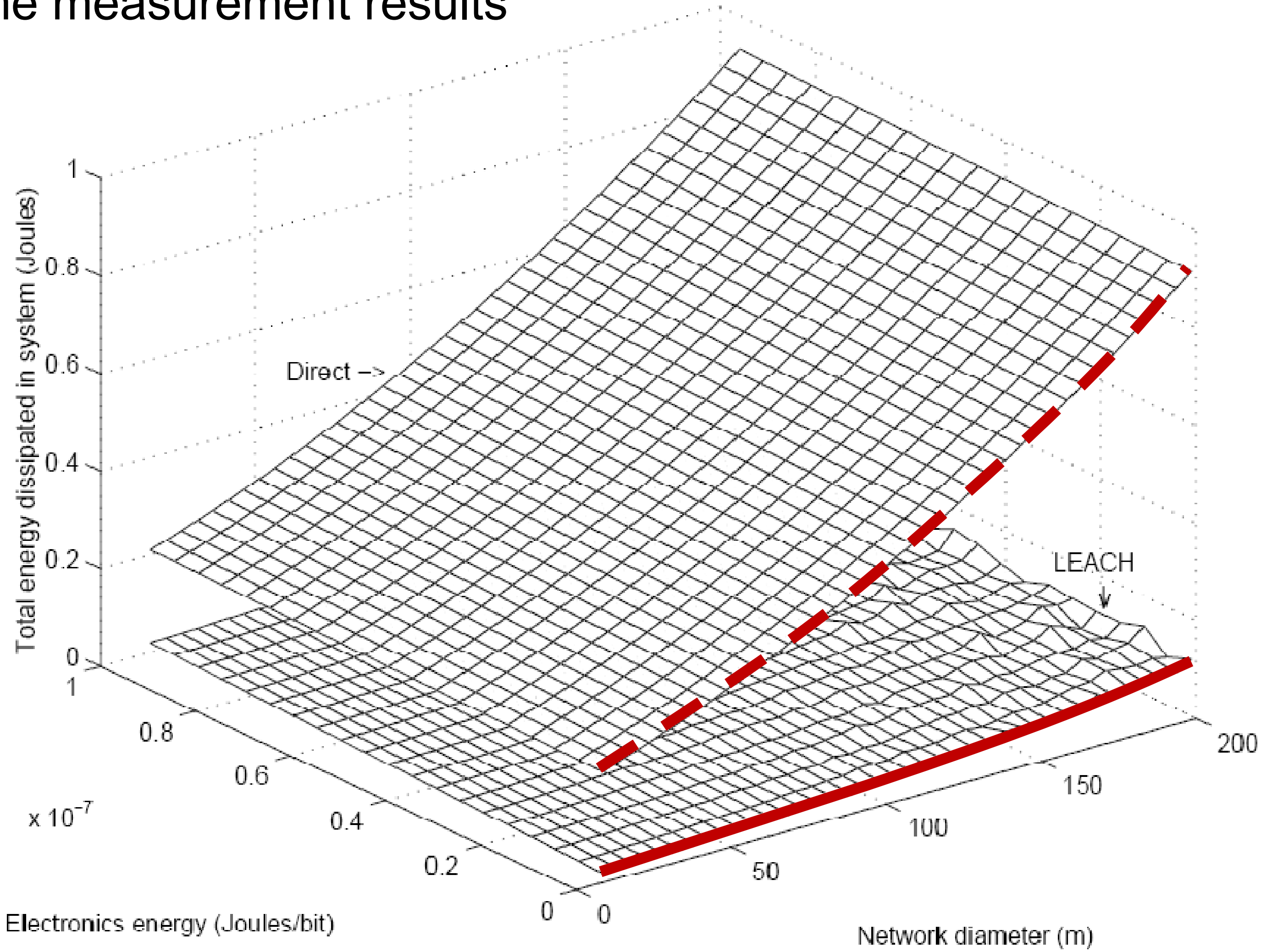
- Cluster set-up phase
  - Each node must inform the clusterhead node that it will be a member of the cluster by transmitting this information back to the cluster-head
  - The clusterhead node receives all the messages for nodes that would like to be included in the cluster; based on the number of nodes in the cluster, the clusterhead node creates a TDMA schedule that is broadcast back to the nodes in the cluster.

LEACH

- Steady-state phase
  - Assuming nodes always have data to send, they send it during their allocated transmission time to the clusterhead
  - This transmission uses a minimal amount of energy (chosen based on the received strength of the clusterhead advertisement)
  - The radio of each non-clusterhead node can be turned off until the node's allocated transmission time, thus minimizing energy dissipation in these nodes
  - The clusterhead node must keep its receiver on to receive all the data from the nodes in the cluster
  - The clusterhead is responsible to forward appropriate messages to the base station; since the base station is far away, this is a high-energy transmission
  - After a certain (a priori determined) time, the next round begins

LEACH

- Some measurement results



HEED

- HEED – Hybrid Energy-Efficient Distributed Clustering

- Similar to LEACH but incorporates the currently available remaining energy at each node for the (still probabilistic) self-election of clusterheads
- Three protocol phases to set-up the cluster structure: initialize, cluster set-up, and finalize
- Calculation of the probability CH_{prob} to become clusterhead based on the initial amount of clusterheads C_{prob} among all n nodes and the estimated current residual energy in the node $E_{residual}$ and maximum energy E_{max}

$$CH_{prob} = C_{prob} \times (E_{residual} / E_{max})$$

HEED

- ***Hybrid approach*** – clusterheads are probabilistically selected based on their residual energy

- ***Objective function for the reduced communication cost*** – average minimum reachability power defined as the mean of the minimum power levels required by all nodes within the cluster range to reach the clusterhead

- Capabilities
  - Simulation results demonstrate that HEED prolongs network lifetime
  - The operating parameters, such as the minimum selection probability and network operation interval, can be easily tuned to optimize resource usage according to the network density and application requirements.

Summary (what do I need to know)

- ***Clustering techniques***
  - Objectives and principles
- ***k-means and hierarchical clustering***
  - Algorithm
  - Advantages and limitations
- ***LEACH and HEED***
  - LEACH algorithm
  - Distribution of energy load, overhead
  - Improvements by HEED

References

- Y. P. Chen, A. L. Liestman, and J. Liu, "Clustering Algorithms for Ad Hoc Wireless Networks," in *Ad Hoc and Sensor Networks, Y. Xiao and Y. Pan, Eds.: Nova Science Publisher, 2004.*
- Y. Fernandess and D. Malkhi, "K-Clustering in Wireless Ad Hoc Networks," Proceedings of 2nd ACM Workshop on Principles of Mobile Computing, Toulouse, France, 2002, pp. 31-37.
- W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proceedings of 33rd Hawaii International Conference on System Sciences, 2000.
- J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm " *Applied Statistics, vol. 28 (1), pp. 100-108, 1979.*
- S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika, vol. 32 (3), pp. 241-254, September 1967.*
- O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Transactions on Mobile Computing, vol. 3 (4), pp. 366-379, October-December 2004.*