On Clustered Ad hoc Networks:

Link-State Clustering Algorithm and Energy Performance Study

Chao Gao & **Riku Jäntti**

Department of Computer Science

University of Vaasa

Finland

Outline of the Talk

- Motivation
- LSCA for heterogenous networks
  - System model
  - Clustering algorithm
  - Performance analysis
  - Overhead comparison
- LSCA for homogenous networks
  - Scalability
- Conclusion & Future Work

Motivation

- Energy performance is one of the most critical issues of wireless ad hoc networks and sensor networks.
- Network overheads usually take significant amount of energy, especially when the network scale grows.
- Flat ad hoc network routing protocols are not applicable to large-scale networks.
- Dividing the whole network into clusters will results in much less overhead.
- We propose a Link-State Clustering Algorithm (LSCA) that can be applied to either heterogenous or homogenous networks.

LSCA for Heterogenous Network – system model

- We assume a network, in which there exists two kinds of nodes: Heavy-weight Nodes (HN) and Light-weight Nodes (LN).
- HNs have higher battery capacity than that of LNs.
- A HN has two stages of transmit power (and thus the radio range): the higher one P_{TxH} used for intercluster and the lower one P_{TxL} for its slaves.
- All the nodes use CSMA/CA MAC protocol.
- All the nodes are uniformly and randomly distributed. Mobility is considered in this model.

Furthermore, we can clusterize any heterogeneous ad hoc network in a way that all the mobile nodes with their **battery capacity** E_b greater than a **threshold** E_{bth} as HN and those with battery capacity less than Ebth as LN.

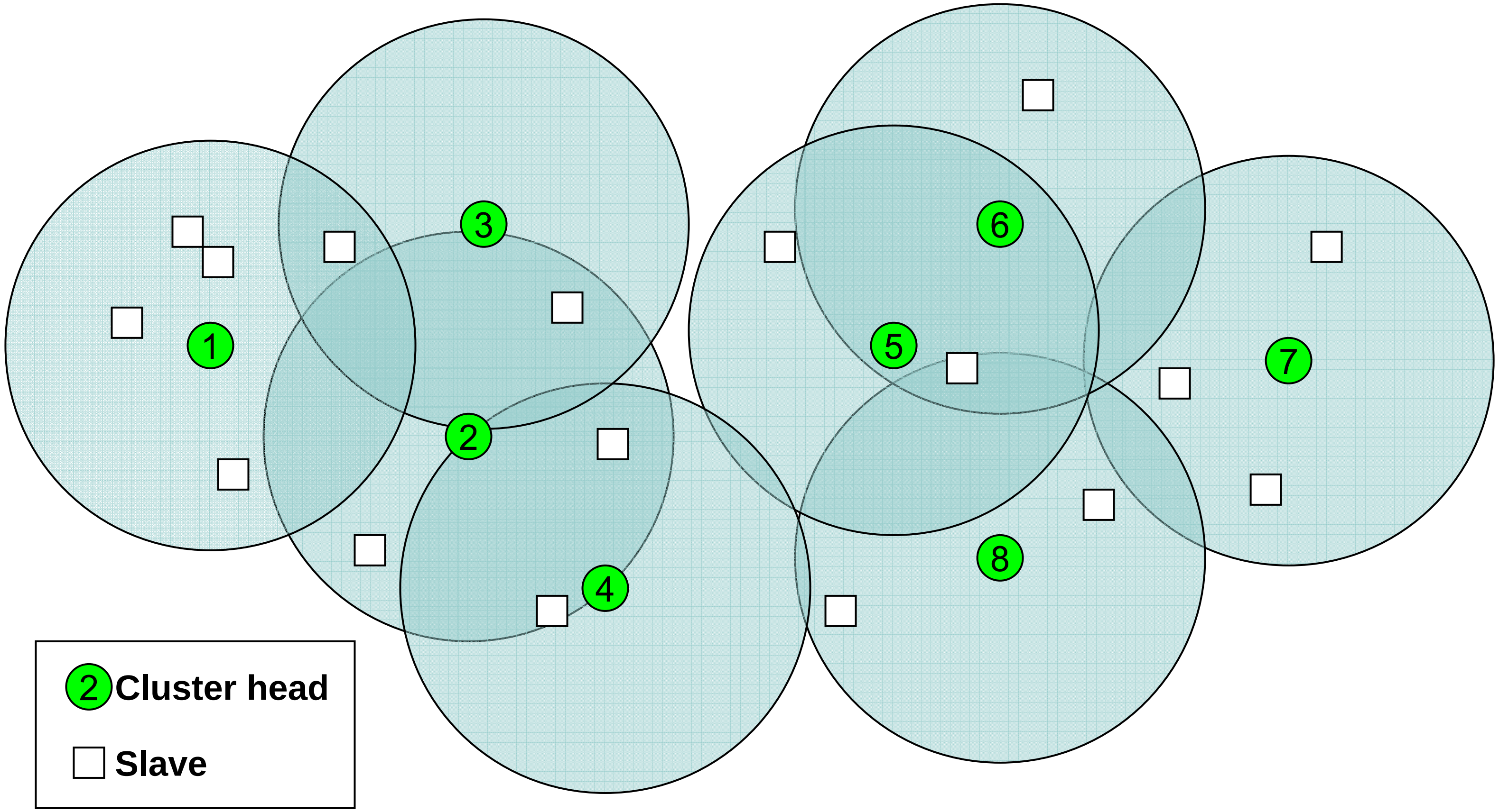
Clustering Algorithm for Heterogenous Network

- Each HN will act as Cluster Head (CH). A CH contains a predetermined **Cluster ID** (CID) and a **Slave Table** (ST).
- The CID will be broadcast and shared by all its slaves.
- A HN periodically broadcasts **BEAcon for Clustering** (BEAC) containing its CID with transmit power P_{TxL}.
- The period to re-broadcast BEAC is T_{BEAC}, which can be either fixed or variable.
- A LN should always be a slave of one and only one HN.

Clustering Algorithm for Heterogenous Network (Cont.)

- **Cluster Forming** A LN sets itself as clusterless when powered on, i.e., CID=UNKNOWN. Upon the reception of the first BEAC, it marks itself as a SL of the corresponding HN and sends back a **Beacon Reply** (BREP). The HN will add it to ST. The LN also records the SNR of the received BEAC, denoted as Γ. Γ represents the **link state**.
- **Cluster Updating** If a LN has received a new BEAC from another HN, it will compare the link state with the previous one. If $\Gamma_{new} > \Gamma_{old}+\Delta$, it updates its HN by sending two packets: a BREP to inform the new HN, and a **Slave Cancel** (SCAN) to inform the old HN to remove it from the slave table. Δ is chosen large enough to prevent the link fluctuating.

Clustering Example



Performance Analysis - cluster head population

- The network connectivity of a clustered network consists of two parts:
  - the connectivity of clusterheads, and
  - the coverage of clusterheads should cover the whole service area.
- Gupta *et al* in [18] asserted that that if *n* nodes are placed in a disc of unit area in $\Re^2$ and each node transmits at a power level so as to cover an area of $r^2 = (\log n + c(n))/n$, then the resulting network is asymptotically connected with probability 1 if and only if $c(n) \to +\infty$. Penrose has shown that the longest edge M_n of a minimum spanning tree of *n* points randomly distributed in unit area satisfies

$$\Pr(n\pi M_n^2 - \log(n) < b) = e^{-e^{-b}}$$

Performance Analysis
- cluster head population (cont.)

- Hence, by setting $r = \sqrt{(b+\log(n))/n\pi}$ the connectivity probability becomes $e^{-e^{-b}}$. Now it would be easy to compare how much more power would be needed to keep the clusterhead-based backbone network connected with the same probability as the flat network:

$$R_{flat} = \sqrt{(b+\log(N))/N\pi}$$

$$R_{cluster} = \sqrt{(b+\log(N_H))/N_H\pi}$$

Network Overhead Comparison

- **In a fixed area, we compare the overhead of a flat AODV with that of a clustered network. Both network have same number of nodes and generate same amount of traffic.**
- **A Slave always sends data packets to its CH. Routing among CHs is also AODV.**
- **The analysis shows that the network overhead is dramatically reduced in the clustered network model.**

Flat AODV Routing Cost

- **The energy of one routing procedure can be approximated as**

$$\begin{aligned} E_{AODV} &= \sum_{i,i\neq D} (E_{tRREQ} + \sum_{j\in N_i} E_{rRREQ}) + E_{RREP} \\ &= \sum_{1}^{N-1} (E_{tRREQ} + \overline{N} E_{rRREQ}) + E_{RREP} \\ &\approx (N-1)(E_{tRREQ} + \overline{N} E_{rRREQ}) \end{aligned}$$

N : No. of nodes

$\overline{N}$: Average neighbor nodes

E_{tRREQ} : Energy of transmitting a RREQ

E_{rRREQ} : Energy of receiving a RREQ

Clustered Overhead

- **The overhead of a clustered network consists of two parts: routing overhead and clustering overhead.**

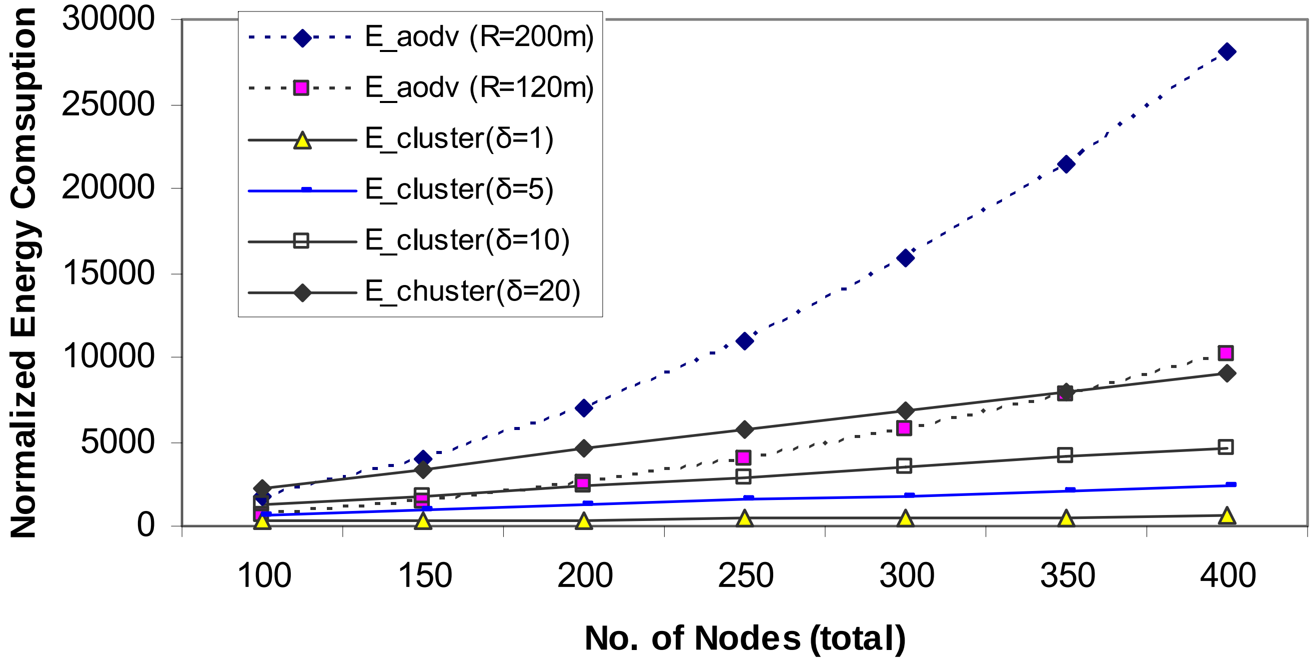
$$E_C = E_{AODV_{N_c}} + \delta N_c (E_{tBEAC} + \overline{N}_s E_{rBEAC})$$

$$\begin{aligned} N_c &: \text{No. of cluster heads} \\ \overline{N}_s &: \text{Average number of slaves} \\ E_{tBEAC} &: \text{Energy of transmitting a BEAC} \\ E_{rBEAC} &: \text{Energy of receiving a BEAC} \end{aligned}$$

- δ **stands for the average number of clustering events between any two routing events:**

$$\delta = \frac{\lambda_c}{\lambda_r}$$

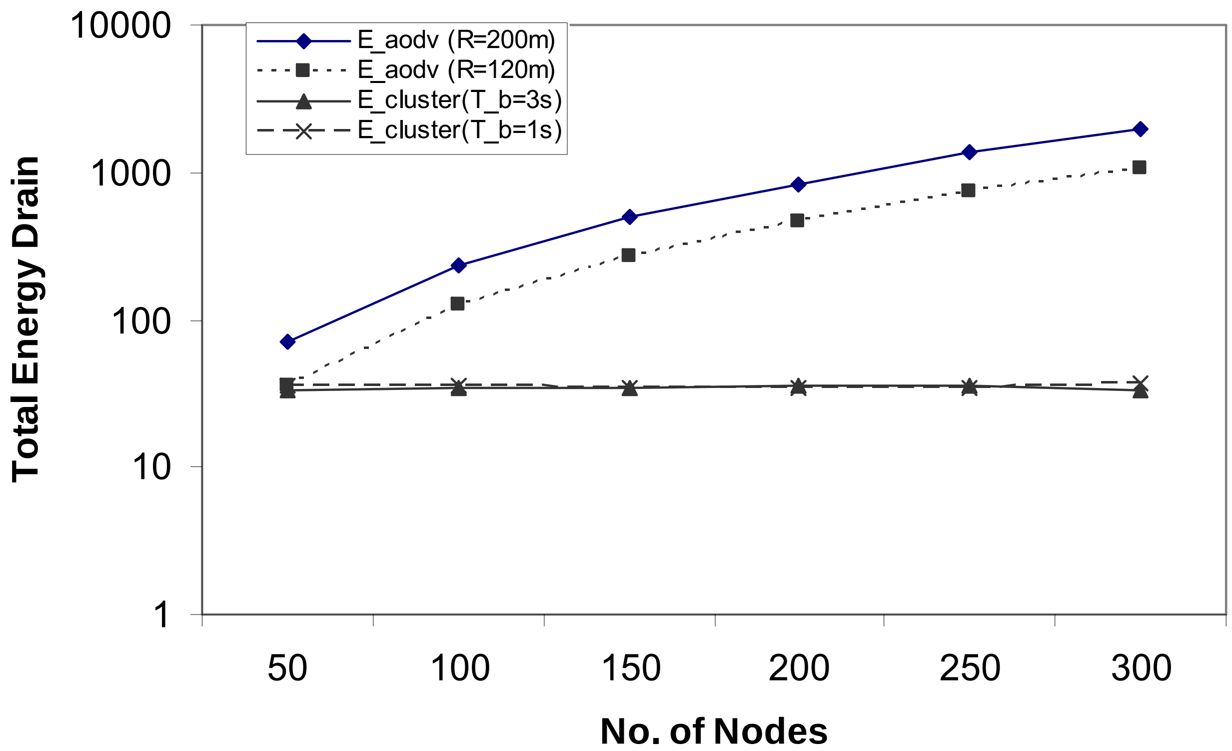
Analysis Results: E_{aodv} vs. $E_{cluster}$ at differrent δ



Simulation Settings

- **Area:** 600x600 (m^2)
- **No. of nodes:** 100
- **No. CHs in clustered mode:** 30
- **AODV radio range:** 200 m
- **CH to slave radio range:** 120 m
- **Mobility (Mean speed):** 2.5 m/s
- **Traffic:** 40 random generated CBR
- **Simulation time:** 50 sec.
- **Clustering Period:** 3 sec.
- **AODV Tx power:** 800 mW
- **CH-CH Tx power:** 800 mW
- **CH-Slave Tx Power:** 300 mW
- **Receive power:** 400 mW

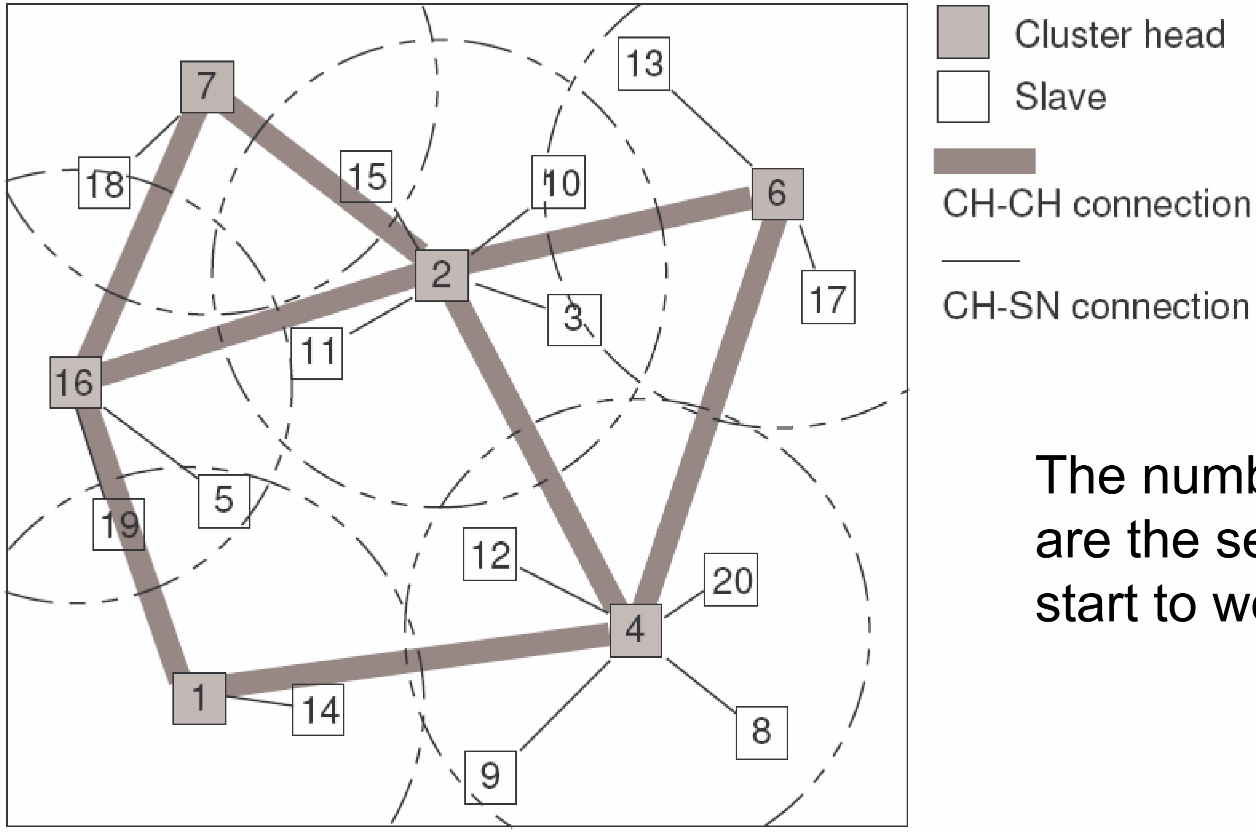
Simulation Results



Clustering for Homogeneous Network

- Three issues must be considered.
- **Cluster Forming** When a node is powered-on, it marks itself clusterless and sets up a waiting timer T_w and starts to monitor the radio channel for BEAC. We set $T_w > T_{BEAC}$ so that nodes have higher priority to be a SL. If no beacon is heard within T_w, the node mark itself as a CH.
- **Cluster Head Re-electing** A SL embeds its budget γ_b in BREP packets. If a node is set as CH, it starts a timer T_h for acting as CH. When T_h expires, it selects its slave that has highest γ_b as the next CH and sends it a packet to notify it.
- **Cluster Head Canceling** If a CH hears a BEAC from another CH, it will set itself as a slave and send a BREP to the other cluster head.

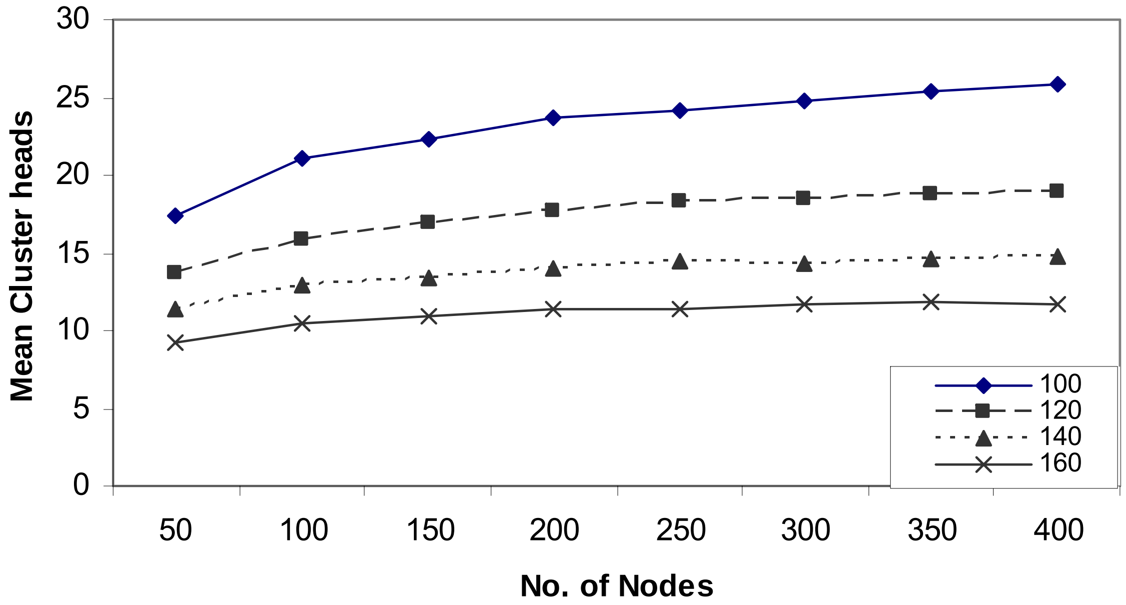
Cluster Forming Example



The numbers on nodes are the sequence they start to work.

Scalability

- **Using the different Clustering range r_c, the population of CH can be controlled.**



An empirical formula can be draw out for the number of cluster heads

$$N_H \approx 2.4N/\pi r^2$$

Conclusion

- In this paper we proposed a clustering algorithm based on the link state between cluster heads and slaves.
- This algorithm can be applied to both pre-determined heterogeneous networks (LSAC-he), and homogeneous networks construct a virtual backbone (LSAC-ho).
- The simulation results show that the overhead energy consumption of a flat ad hoc network is a dominating factor of overall energy drain.
- The algorithm is scalable.

Future Work

- End-to-end packet delivery delay.
- Mobility impacts.
  - A proper rebroadcasting period of beacon depends on the mobility of the nodes.
  - An optimal rebroadcasting period is desired to minimize the clustering overhead when the connectivity of the network is kept.
- Comparison with other types of clustering algorithms.
- Effect of ***chain reaction***.

Thank you!