CNRL
Computer Networking
Research Laboratory

Colorado State University
Knowledge to Go Places

An Enhanced Top-Down Cluster and Cluster Tree Formation Algorithm for Wireless Sensor Networks

H. M. N. Dilum Bandara, Anura P. Jayasumana
dilumb@engr.Colostate.edu, Anura.Jayasumana@Colostate.edu

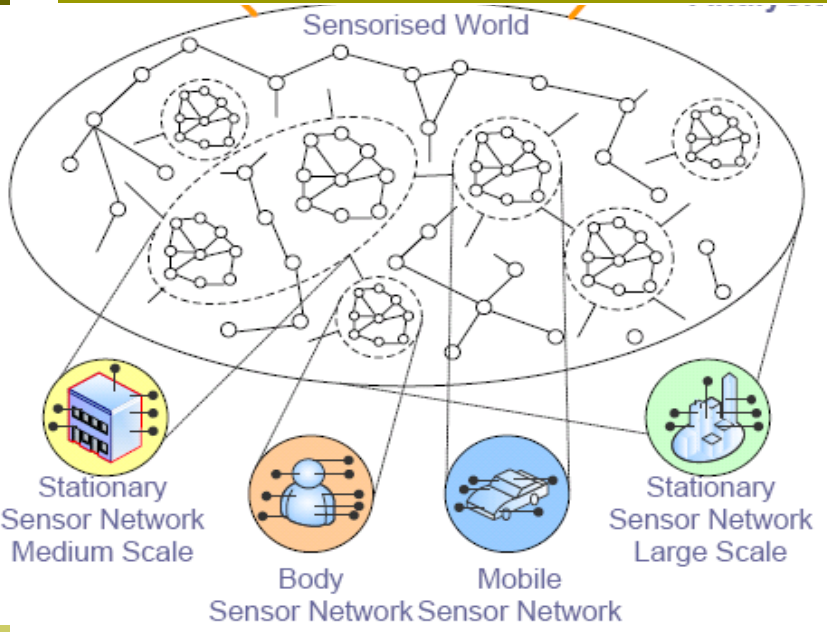
Department of Electrical and Computer Engineering,
Colorado State University, USA.

Outline

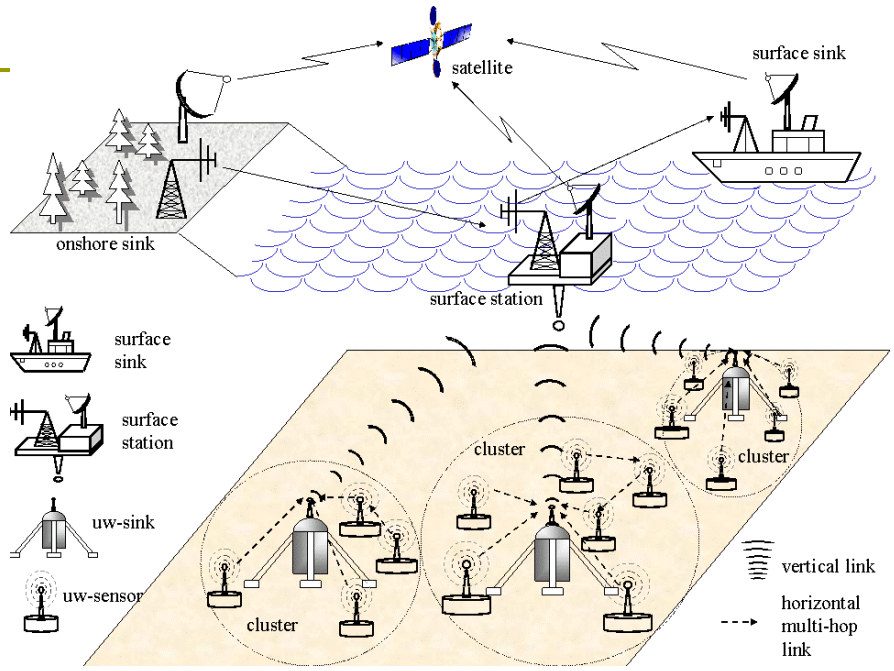
- Wireless Sensor Networks (WSN)
- Motivation
- GTC – Generic Top-down Clustering algorithm
- Control of cluster & cluster tree characteristics
- Simulation results
  - Simulator
- Conclusions & future work

Wireless Sensor Networks (WSN)

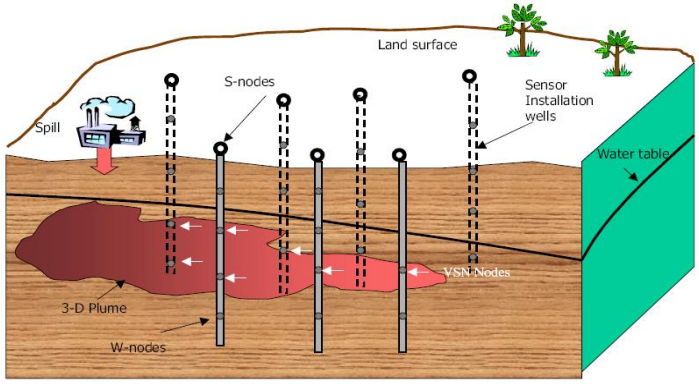
Clustering



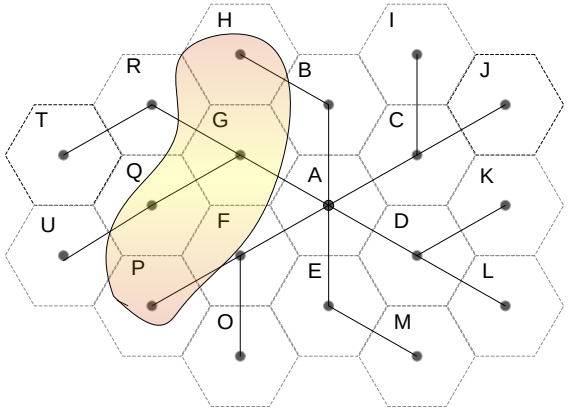
e-SENSE project [1]

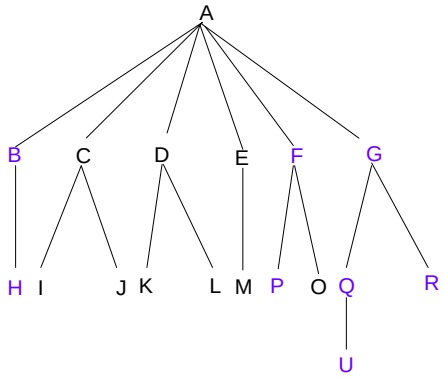


Underwater Acoustic Sensor Networks project [2]



Plume tracking





[1] www.ist-esense.org
[2] http://www.ece.gatech.edu/research/labs/bwn/UWASN/work.html

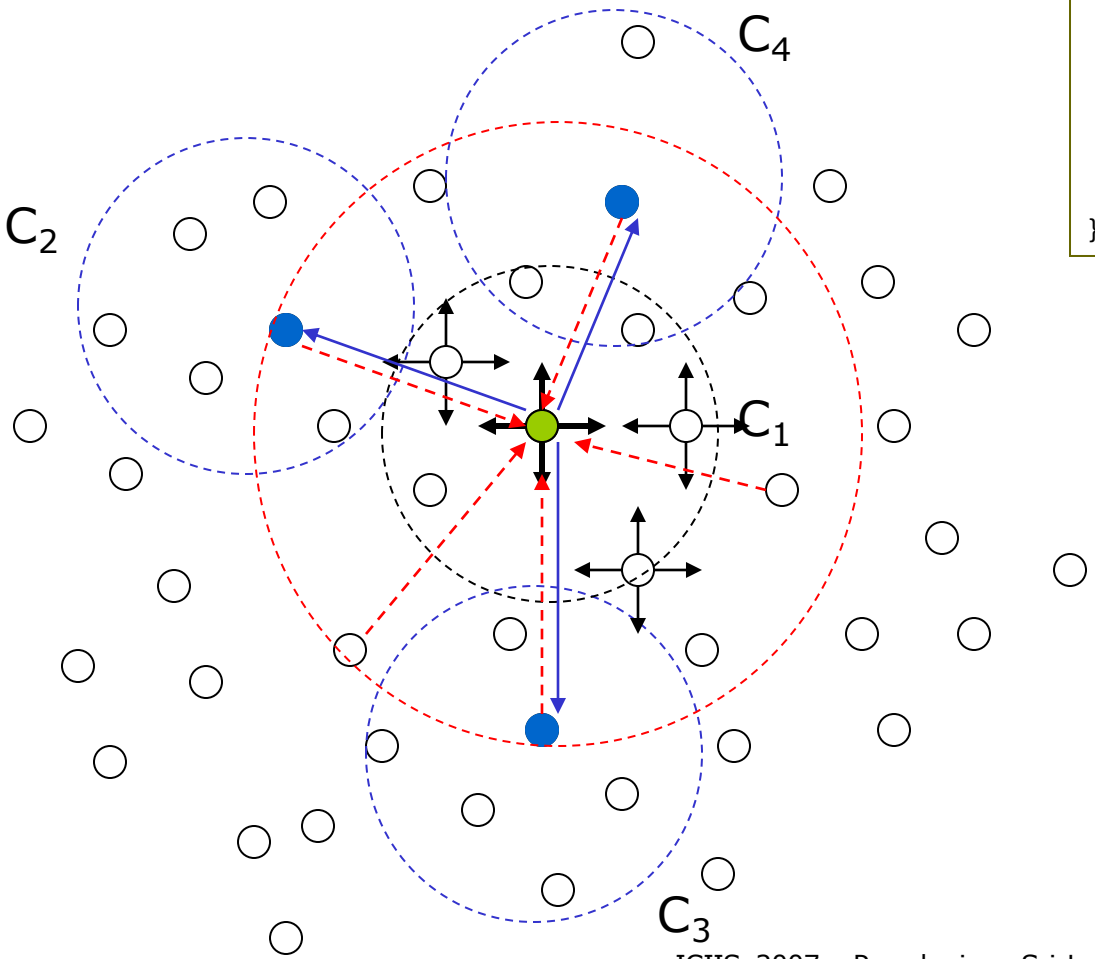
Motivation

- Some structure is required in future large scale WSNs, even if they are randomly deployed
  - Ease of administration
  - Better utilization of resources
  - Simplified routing
- An algorithm that is independent of
  - Neighbourhood information
  - Location awareness
  - Time synchronization
  - Network topology
- Top-down clustering allow better control
  - Controlled cluster size, controlled tree formation, hierarchical naming, etc.
- An algorithm that supports the existence of multiple WSNs in the same physical region

Generic Top-down Clustering (GTC) algorithm

```
Form_cluster(NID, CID, T, N, MaxHops, TTL) {
      Wait(T)
      Broadcast_cluster(NID, CID, MaxHops, TTL)
      ack_list ← Receive_ack(CNID, hops, timeout, P1, P2)
      For i = 1 to N {
            CCHi ← Select_candidate_CH(TTL, ack_list, P1, P2)
            CIDi ← Select_next_CID()
            Ti ← Select_delay()
            Request_form_cluster(CCHi, CIDi, Ti, N, MaxHops, TTL)
      }
}
Join_cluster() {
      Listen_broadcast_cluster(NID, CID, MaxHops, TTL)
      If(hops ≤ MaxHops & MyCID = 0)
            MyCID ← CID, MyCH ← NID
      Send_ack(CNID, Hops)
      TTL ← TTL -1
      If(TTL > 0)
            Forward_broadcast_cluster(NID, CID, MaxHops, TTL)
      Else {
            Listen_form_cluster(CCH, CID, T, N, MaxHops, TTL, timeout)
            Form_cluster(CCH, CID, T, N, MaxHops, TTL)
      }
}
```

Cluster formation

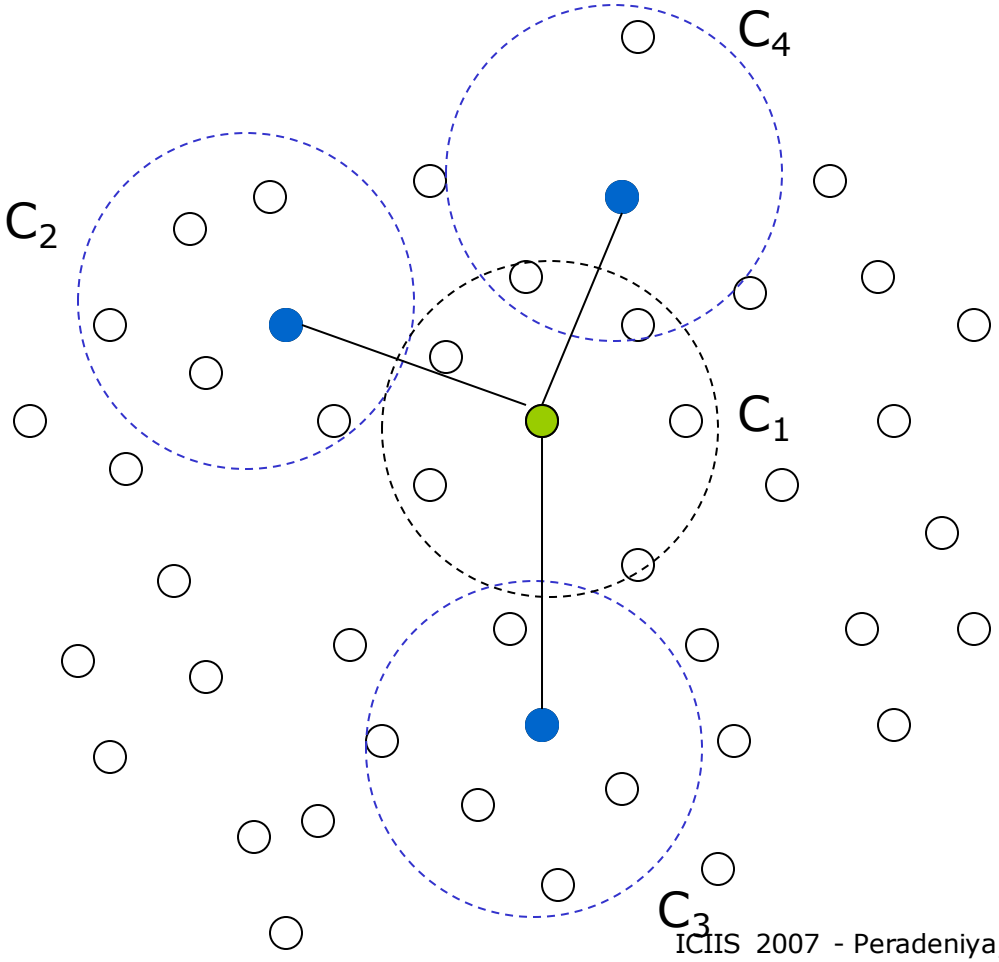


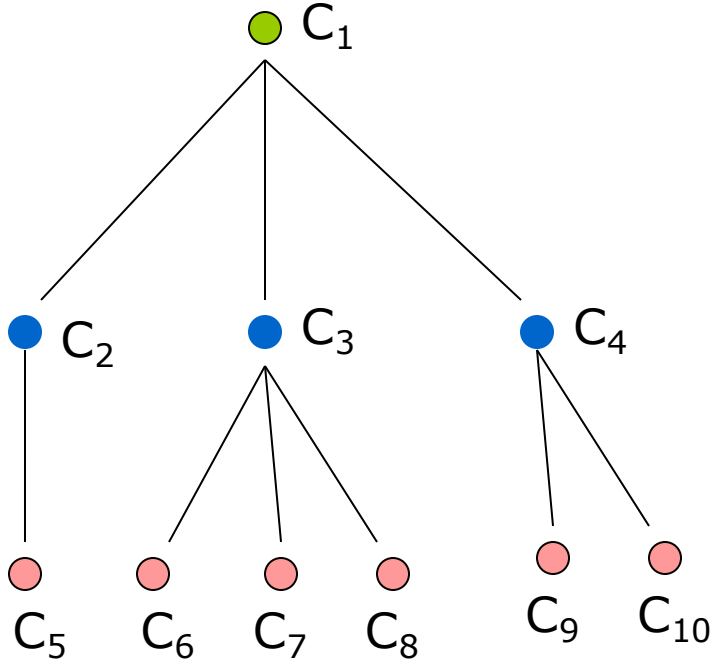

```
Form_cluster(NID, CID, T, N, MaxHops, TTL) {
    Wait(T)
    Broadcast_cluster(NID, CID, MaxHops, TTL)
    ack_list ← Receive_ack(CNID, hops, timeout, P1, P2)
    For i = 1 to N {
        CCHi ← Select_candidate_CH(TTL, ack_list, P1, P2)
        CIDi ← Select_next_CID()
        Ti ← Select_delay()
        Request_form_cluster(CCHi, CIDi, Ti, N, MaxHops, TTL)
    }
}
```

```
Join_cluster() {
    Listen_broadcast_cluster(NID, CID, MaxHops, TTL)
    If(hops ≤ MaxHops & MyCID = 0)
        MyCID ← CID, MyCH ← NID
    Send_ack(CNID, Hops)
    TTL ← TTL -1
    If(TTL > 0)
        Forward_broadcast_cluster(NID, CID, MaxHops, TTL)
    Else {
        Listen_form_cluster(CCH, CID, T, N, MaxHops, TTL,
timeout)
        Form_cluster(CCH, CID, T, N, MaxHops, TTL)
    }
}
```

Cluster tree formation

- Cluster tree is formed by keeping track of parent & child relationships





Control of cluster & cluster tree characteristics

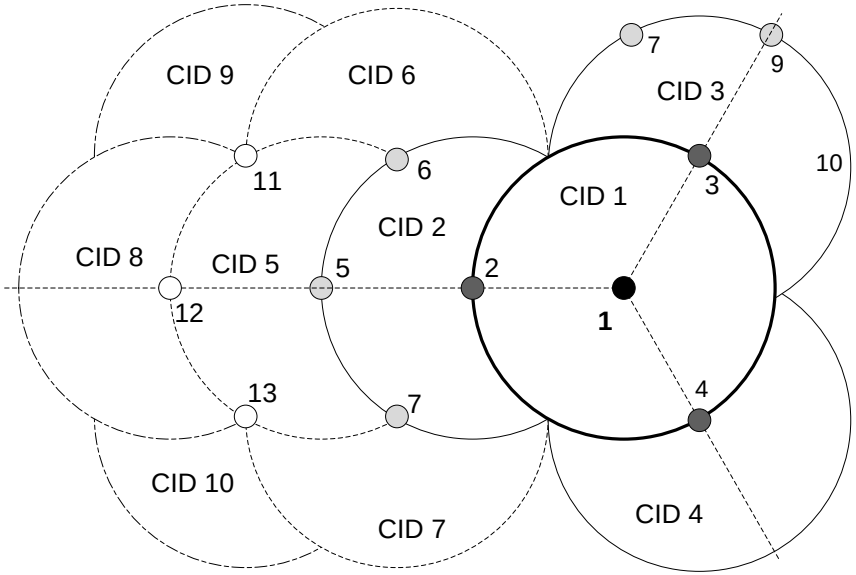
- By varying parameters of the algorithm clusters & cluster tree with desirable properties can be achieved
- Parameters that can be varied
  - *MaxHops* – Maximum distance to a child node within a cluster
  - *TTL* – No of hops to propagate the cluster formation broadcast
  - *N* – No of candidate cluster heads
  - T_i – Time delay before forming cluster *i*
  - CID_i – New cluster ID

Controlling *MaxHops* & *TTL*

- *MaxHops* determine the size of a cluster
  - $MaxHops = 1$ – Single-hop clusters
  - $MaxHops \geq 2$ – Multi-hop clusters

- Two variants of the GTC algorithm
  1. Simple Hierarchical Clustering (SHC)
     - $TTL = MaxHops$
     - New clusters heads are selected from nodes that are within the parent cluster
     - This is similar to the IEEE 802.15.4 clustering
  2. Hierarchical Hop-ahead Clustering (HHC)
     - $TTL = 2 \times MaxHops + 1$
     - New clusters heads are selected from nodes that are outside the parent cluster

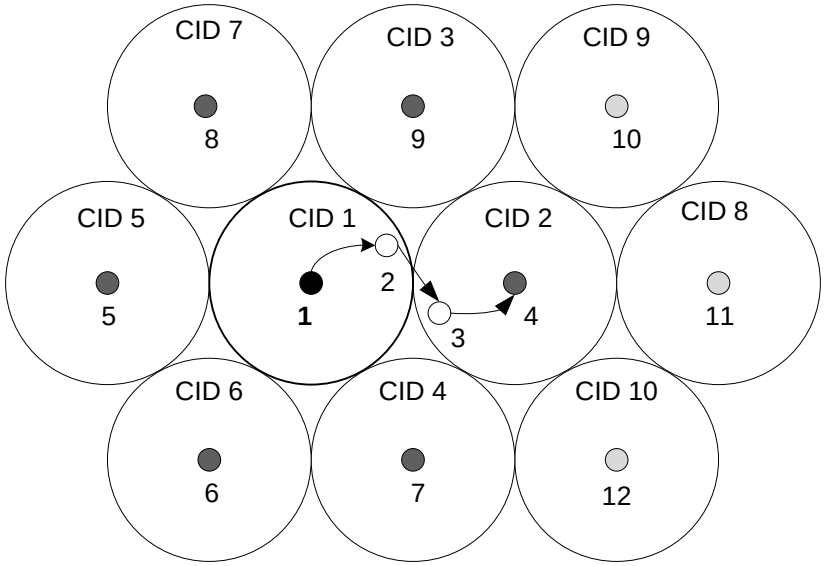
Ideal SHC & HHC clusters



SHC – Simple Hierarchical Clustering

MaxHops = TTL = 1

N = 3



HHC – Hierarchical Hop-ahead Clustering

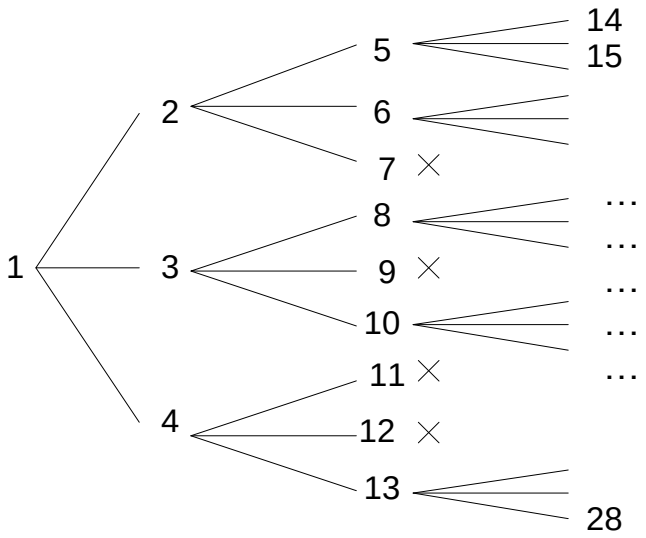
MaxHops = 1

TTL = 3

N = 6

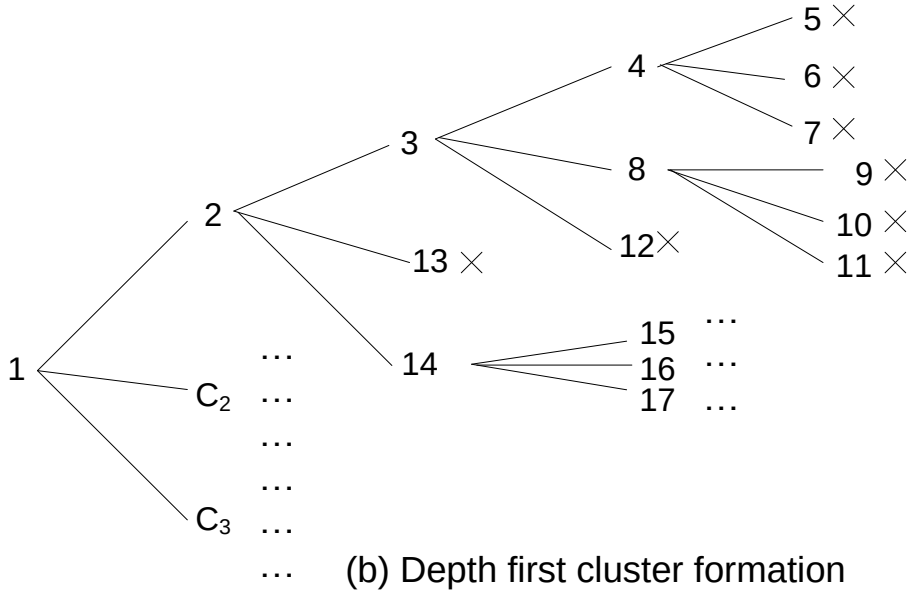
Controlling time delay (T)

- Each candidate cluster head waits sometime before forming a cluster
- This delay prevents collisions
- By varying time delay shape of the cluster tree can be controlled
  - Breadth-first, depth-first or some scheme in between



(a) Breadth first cluster formation

$T_L(i) < T_L(i+1)$

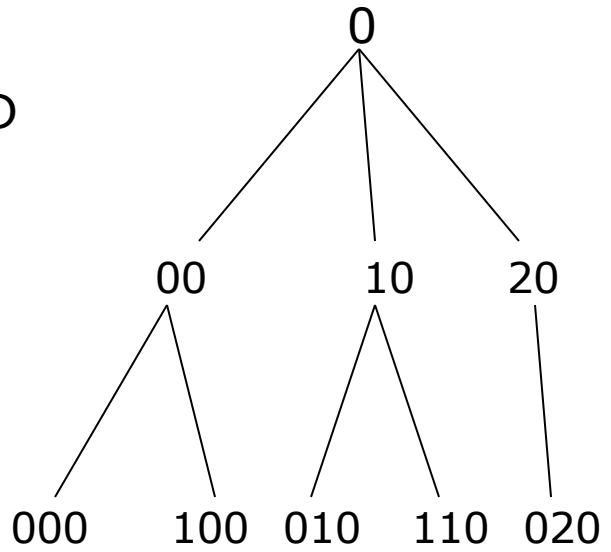


(b) Depth first cluster formation

$T_B(i) < T_B(i+1)$

New cluster ID

- New cluster ID can be assigned
  - as a sequence of numbers – 1, 2, 3
    - Root node must assign cluster ID
  - based on node ID of the candidate cluster head
    - *CID = NID*
    - Parent cluster heads can assign cluster ID
  - based on hierarchical naming
    - Parent cluster heads can assign cluster ID
    - Simplified routing
    - Much easier with the top-down approach

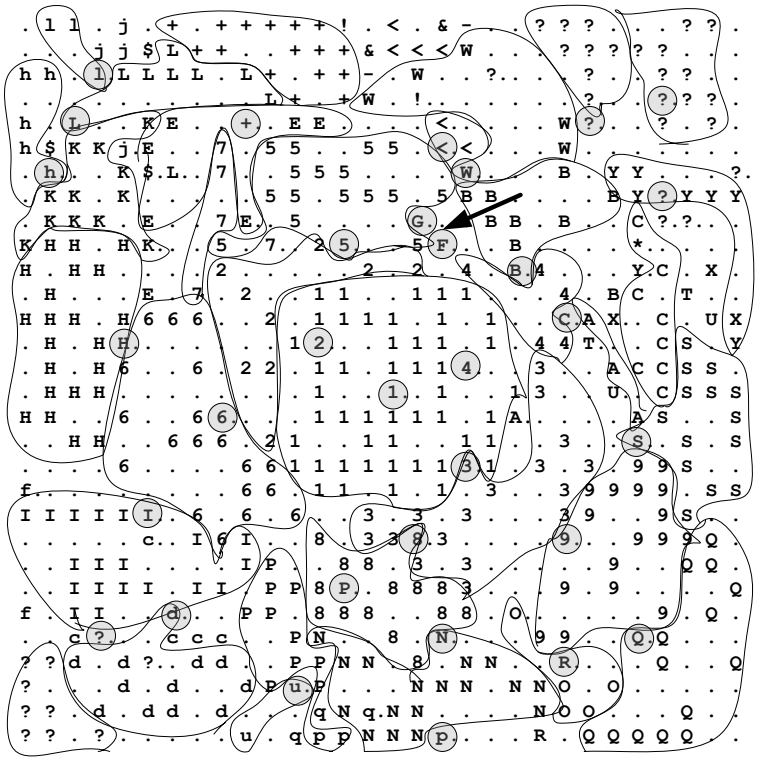


Simulation Results

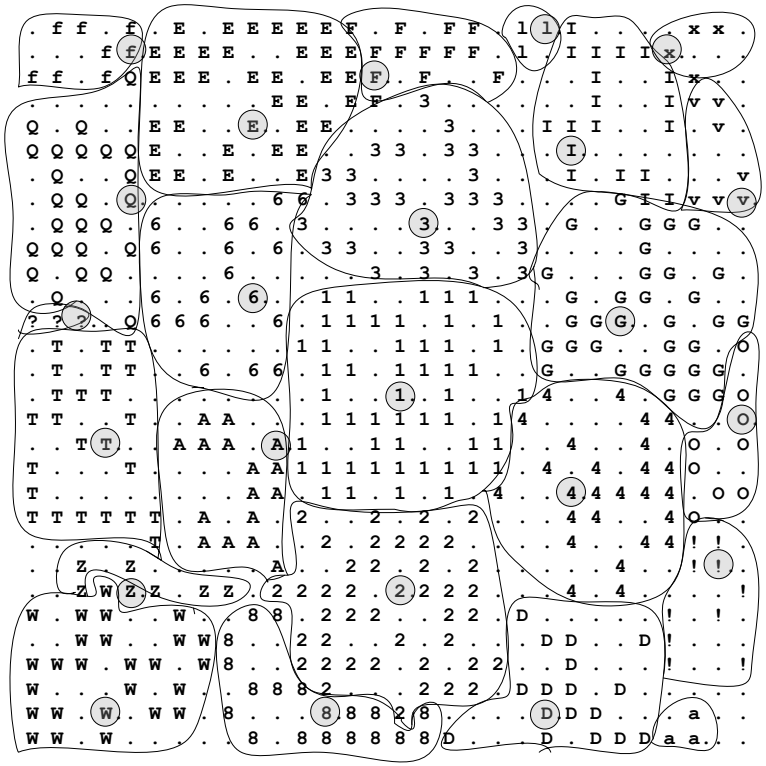
Simulator

- A discrete event simulator was developed using C
- Nodes were randomly placed on a 100×100 square grid with a given probability
  - e.g. 1, 0.5 & 0.25
- 100 sample runs based on pre-generated networks were considered
- *N* was selected such that *N=3* for SHC & *N=6* for HHC
- Circular communication model
  - Within clusters - Multi-hop
  - Cluster head to cluster head - Single-hop
- Assumptions
  - Nodes were homogeneous
  - Stationary
  - Fixed transmission range

Physical shape of the clusters



(a) SHC clusters



(b) HHC clusters

Grid size = 30×30
D = 450
R = 30

For (a):
MaxHops = 1
TTL = 1
N = 3

For (b):
MaxHops = 1
TTL = 3
N = 6

* CHs are indicated by circles

* Cluster heads are highlighted with circles

- HHC produce more circular & uniform clusters

Why clusters needs to be circular?

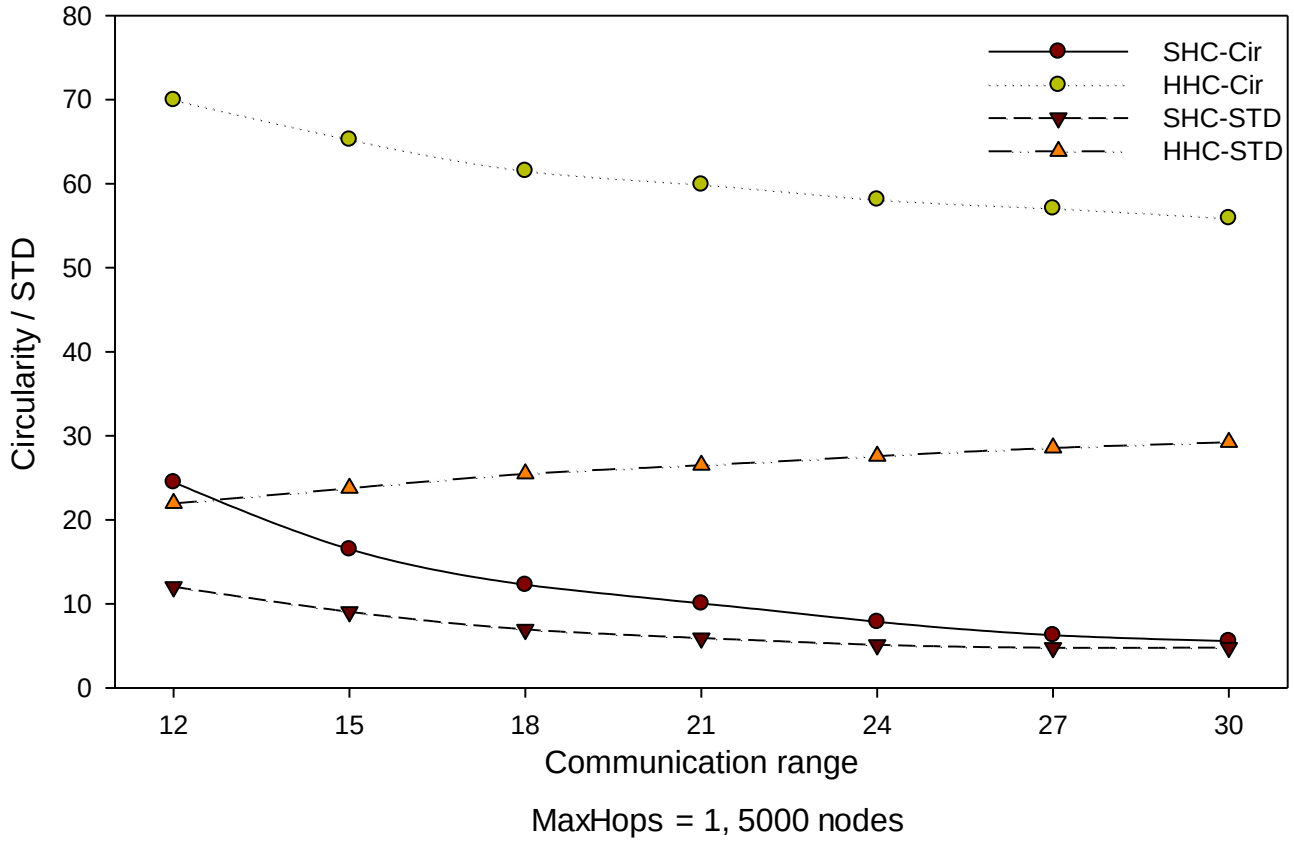
- Efficient coverage of the sensor filed[3]
  - Minimum number of clusters
  - Reduce the depth of the cluster tree
  - Better load balancing
- Topology becomes more predictable
- Reduce intra-cluster signal contention
- Aggregation is more meaningful when cluster head is in the middle

- Measuring circularity
  - Maximum Achievable Circularity (MAC)

$$MAC_i = \frac{No\ of\ nodes\ in\ cluster\ i}{Total\ no\ of\ nodes\ in\ the\ range\ of\ CH_i} \times 100$$

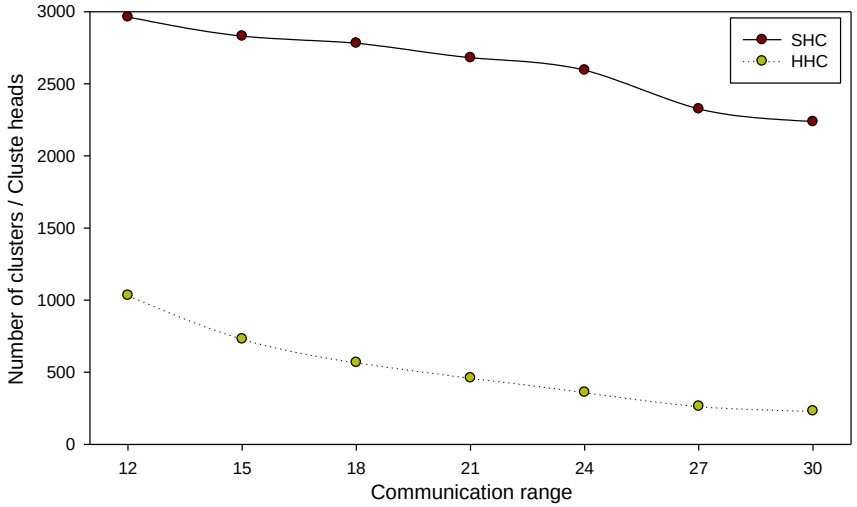
[3] M. Demirbas, A. Arora, V. Mittal and V. Kulathumani, "A fault-local self-stabilizing clustering service for wireless ad hoc networks", IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 9, Sept. 2006, pp. 912-922

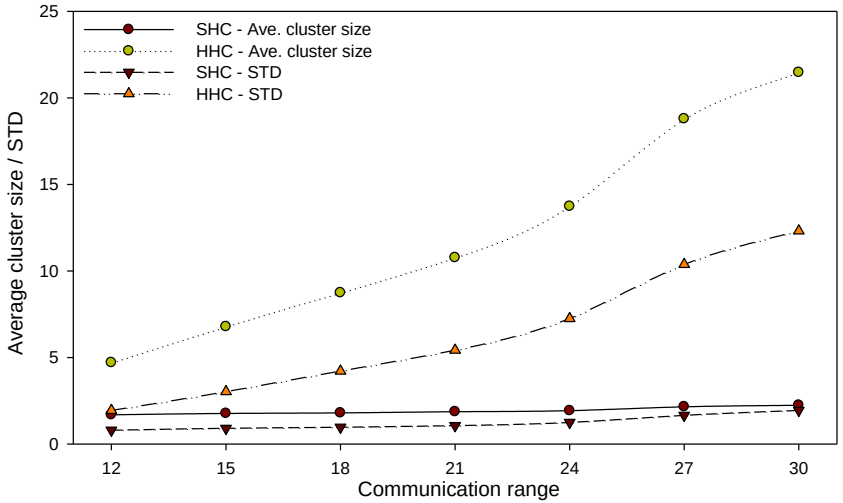
Circularity



- HHC produce more circular clusters than SHC

No of nodes/Clusters

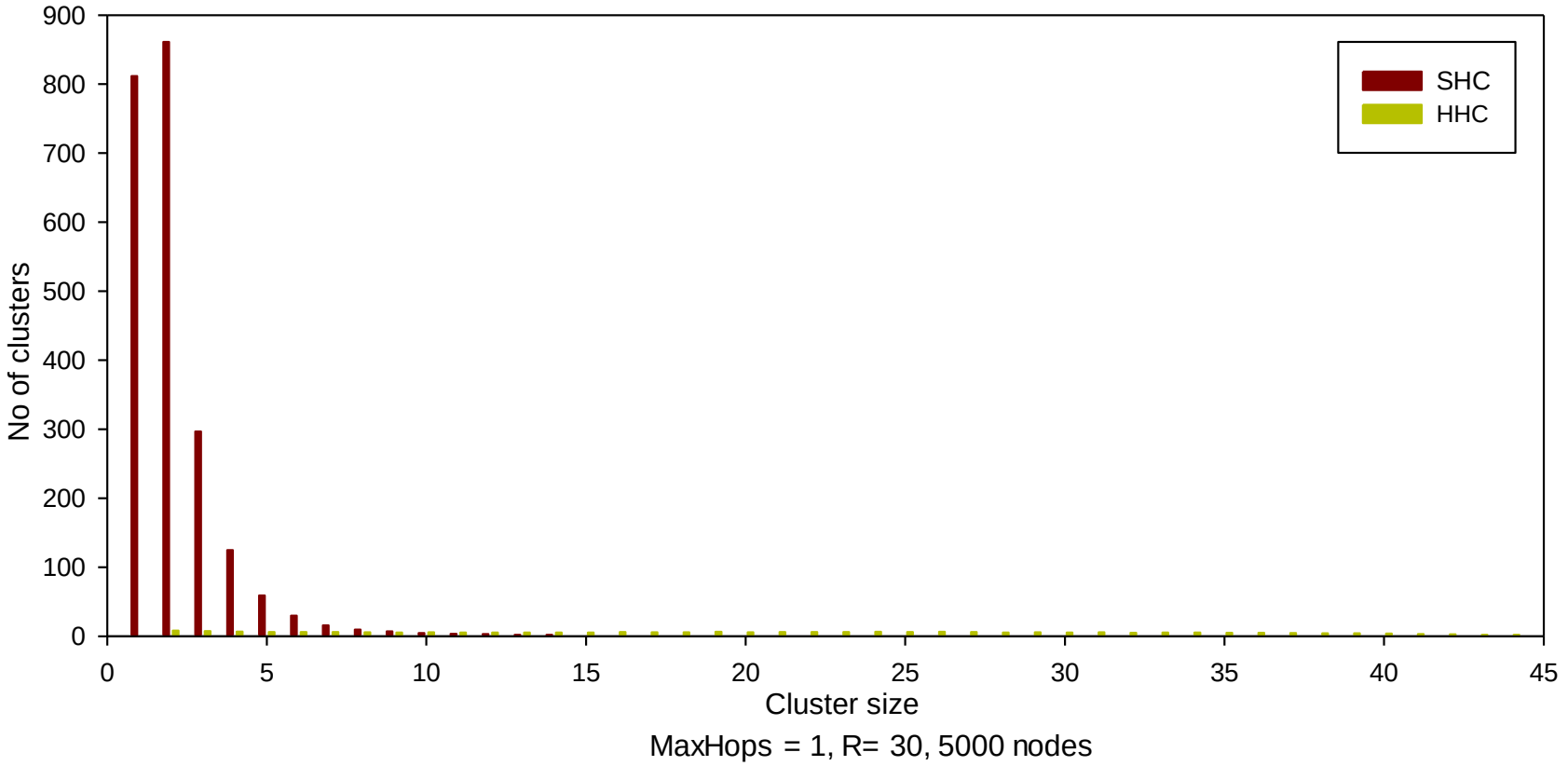




MaxHops = 1, 5000 nodes

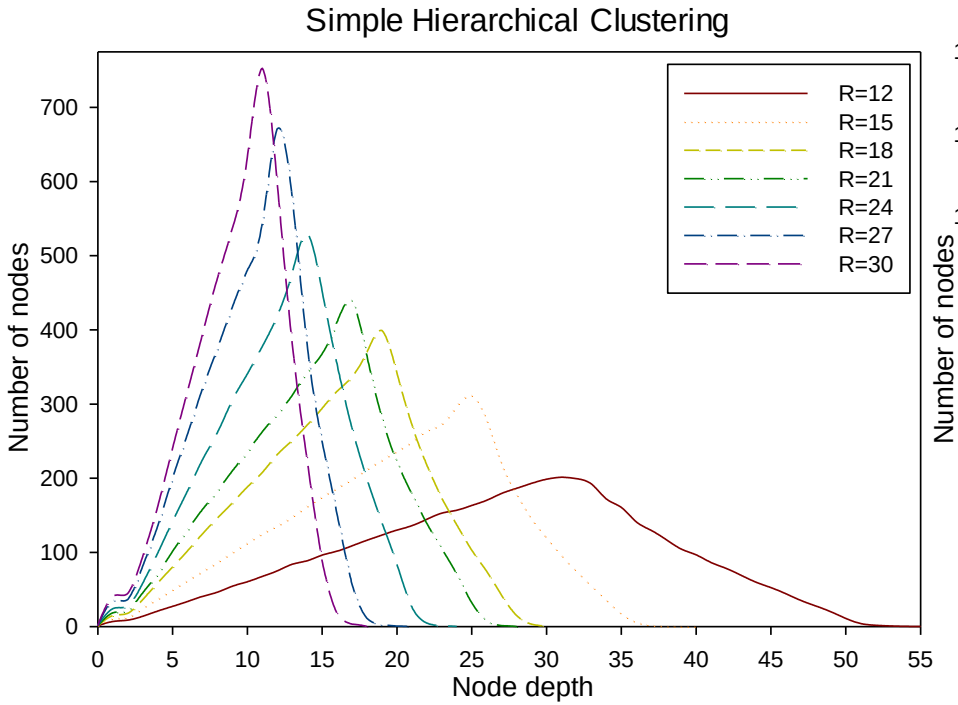
- HHC produces lesser number of clusters
- HHC produces much larger clusters than SHC
- High STD in HHC is due to smaller clusters at the edge of the sensor field
- Larger clusters are formed as the communication range is increased

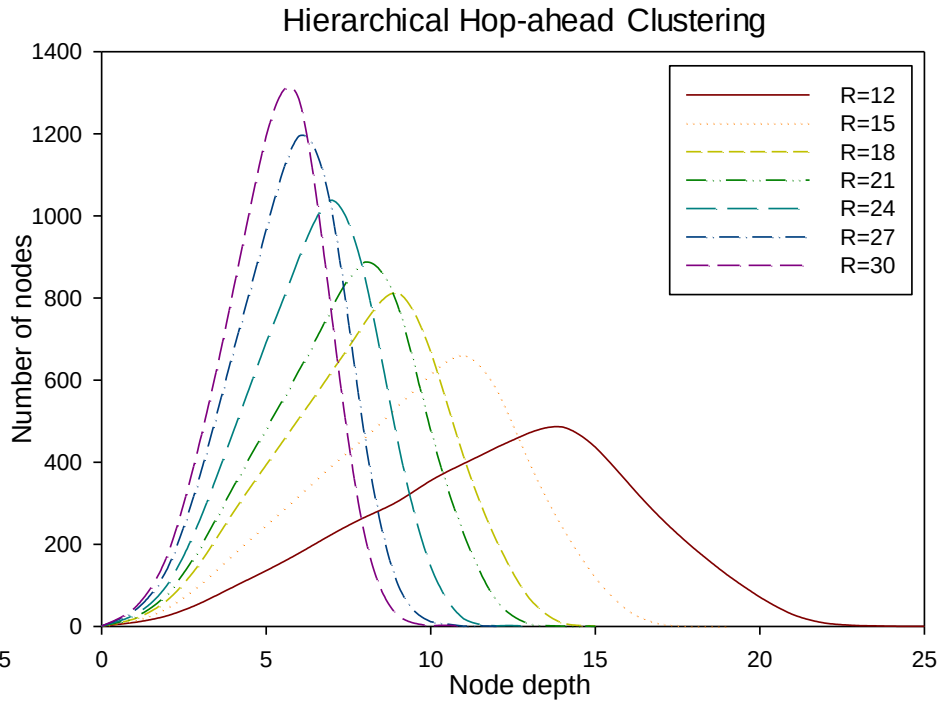
Node distribution



- HHC produces smaller number of large clusters
- SHC produces larger number of small clusters

Node depth distribution - Breadth-first tree formation

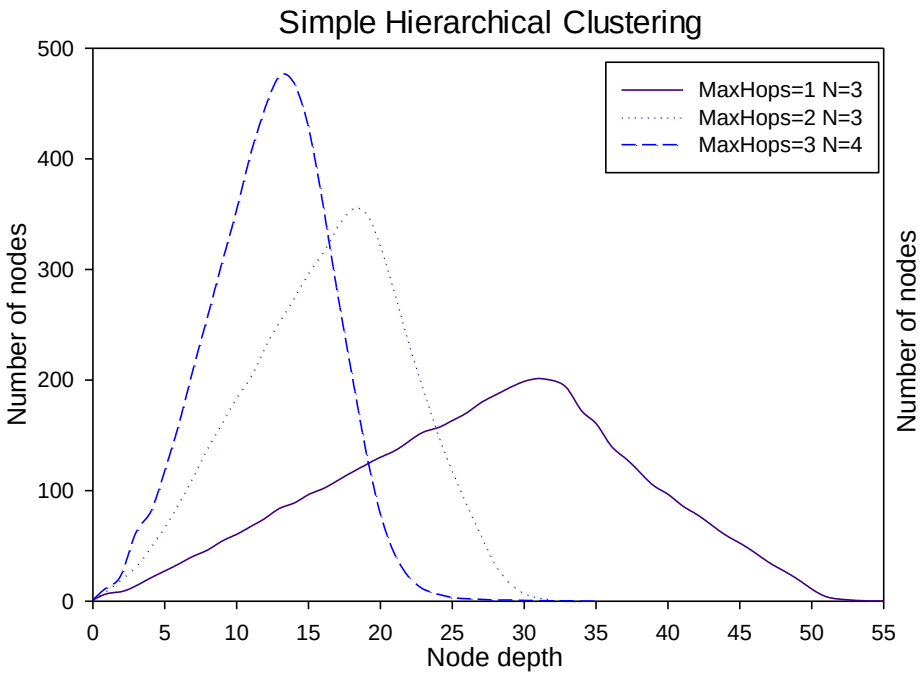


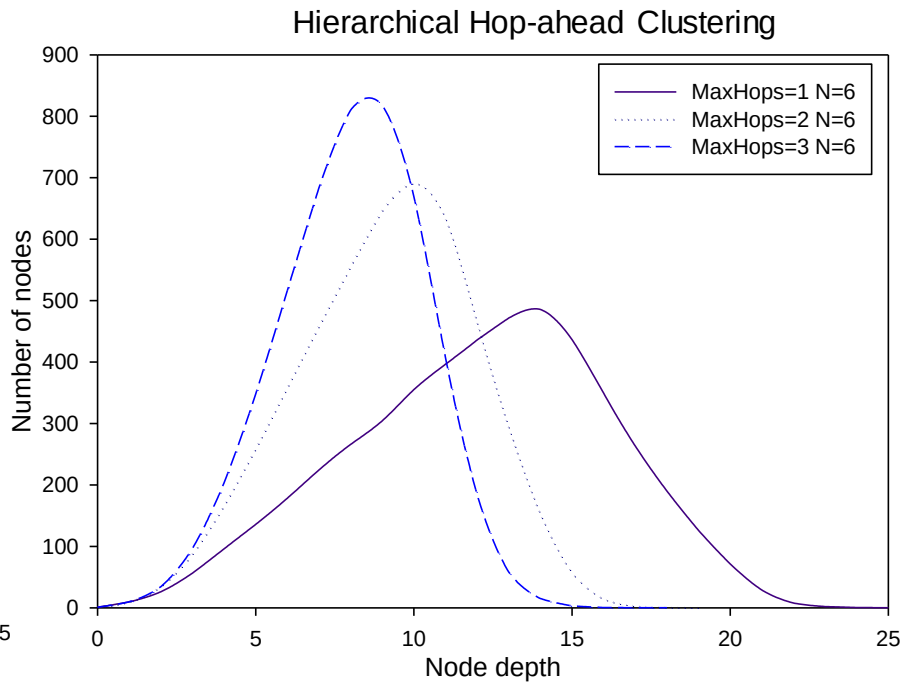


MaxHops = 1, 5000 nodes

- Nodes in HHC have a lower depth than SHC
- Depth reduces as the communication range increases

Node depth distribution - Multi-hop clusters (breadth-first tree formation)

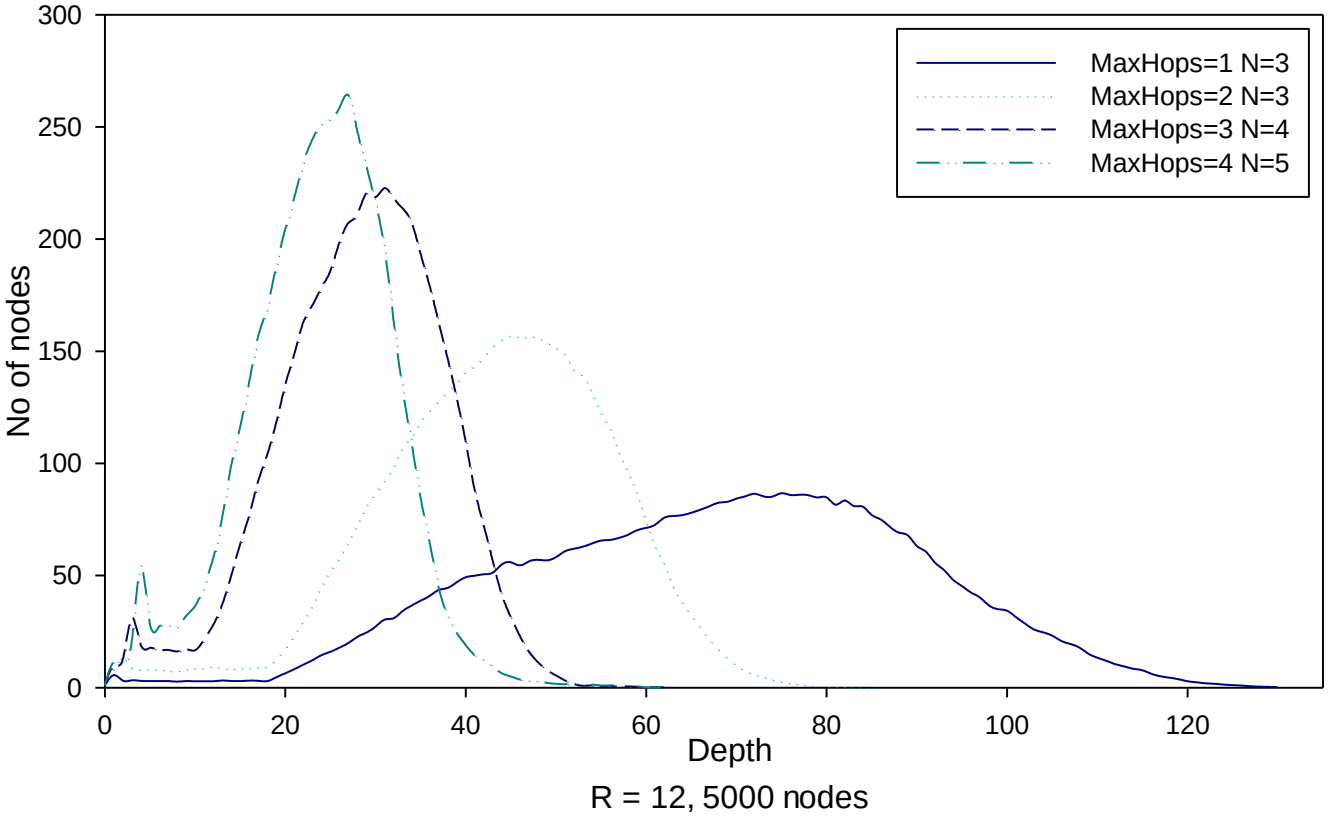




R = 12, 5000 nodes

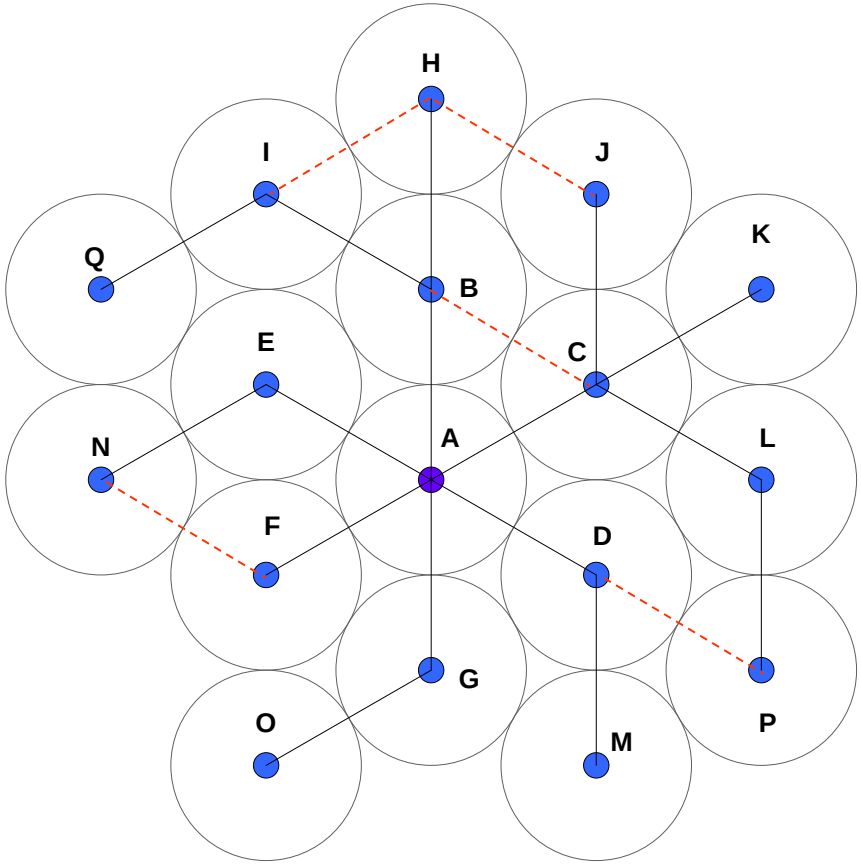
- Depth reduces as the *MaxHops* increases

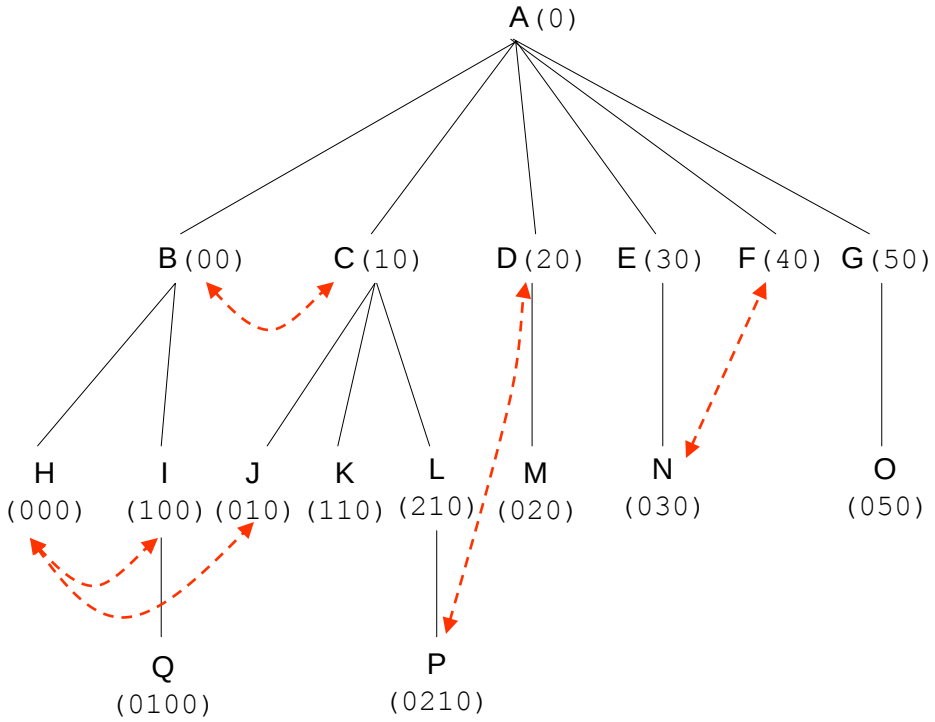
Node depth distribution - Depth-first tree formation



- Depth reduces as the *MaxHops* increases

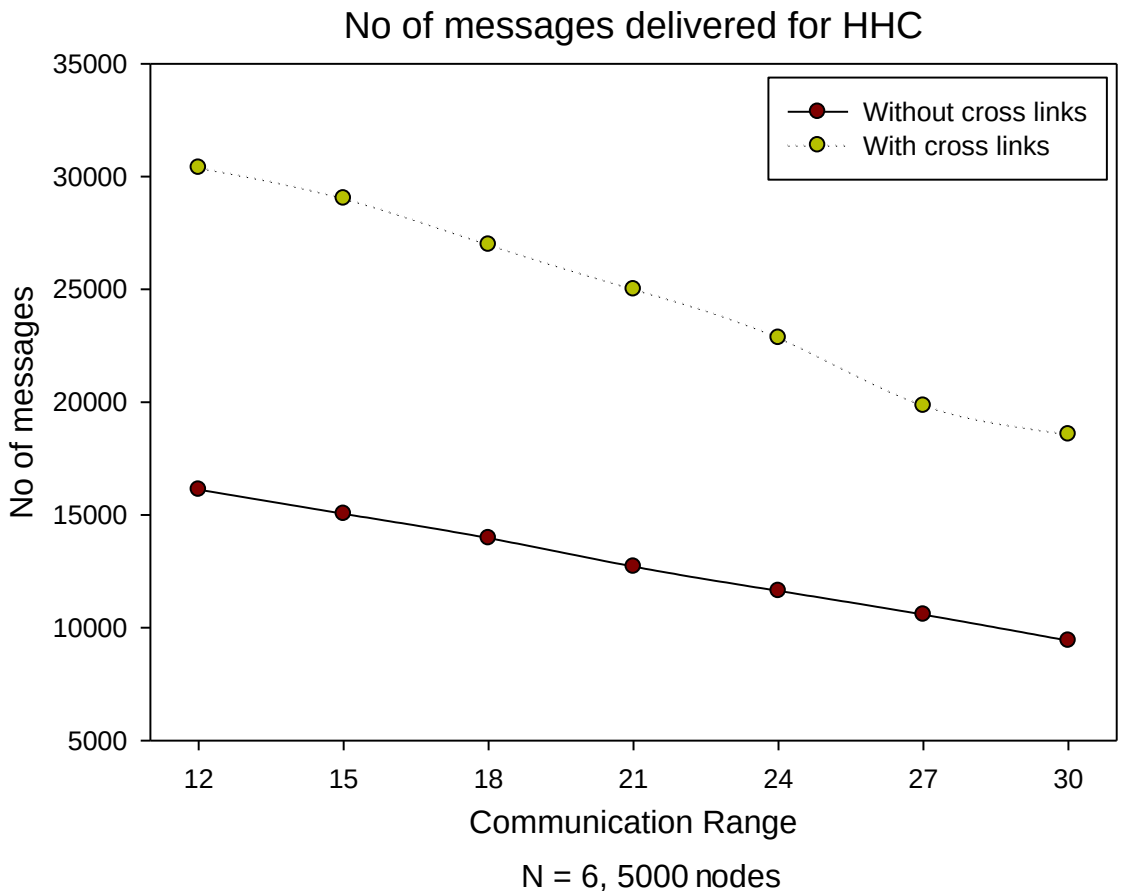
Hierarchical routing





- Routing through cross links
  - Reduce burden on the root node
  - Lower latency

Hierarchical routing & routing with cross links



- Routing with cross links significantly increase the number of messages delivered

Conclusions & future work

- The proposed algorithm is independent of neighbourhood information, location awareness, time synchronization & network topology
- Algorithm scales well into large networks
- The HHC outperforms SHC

- We are currently working on
  - Further optimizing clusters after they are formed
    - Balancing the cluster tree
    - Further reducing node depth
  - Energy aware routing that will further increase the number of messages delivered
    - Increased network lifetime
  - Determining suitable parameter values (*MaxHops*, *TTL*, *N*, *T*, etc.) for optimum performance of the algorithm.

Q&A ...?

Thank you ….