Key Pre-distribution Based Secure Backbone Formation in Wireless Sensor Networks

Dilum Bandara[1], Anura P. Jayasumana[1,2], and Indrajit Ray[2]

[1]Department of Electrical and Computer Engineering,
[2]Department of Computer Science,
Colorado State University, CO 80523, USA.

Anura.Jayasumana@Colostate.edu

Supported in part by a grant from Army Research Office (ARO)

Outline

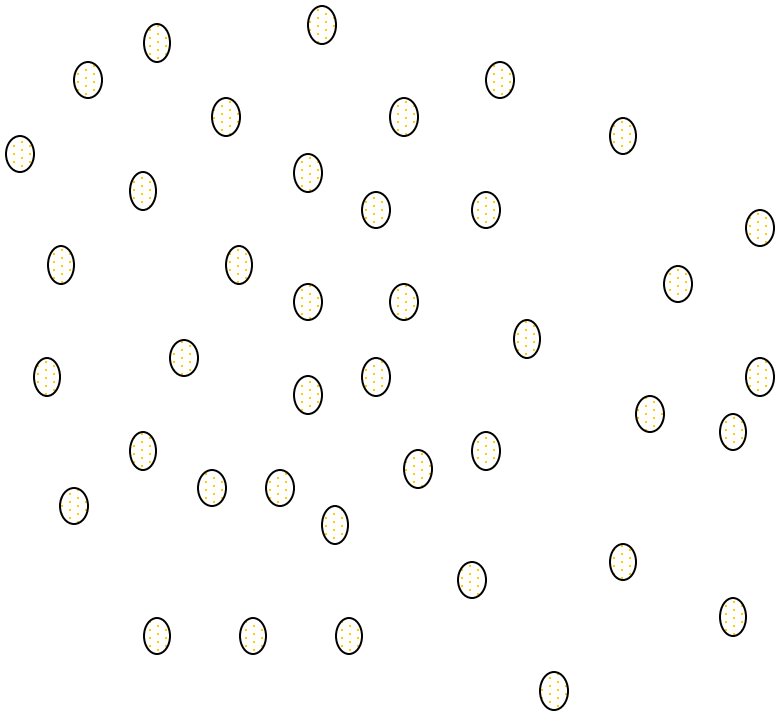
- Motivation
- Key distribution in WSNs
- Extended Generic Top-down Clustering algorithm
- Performance analysis
- Conclusions & future work

Motivation

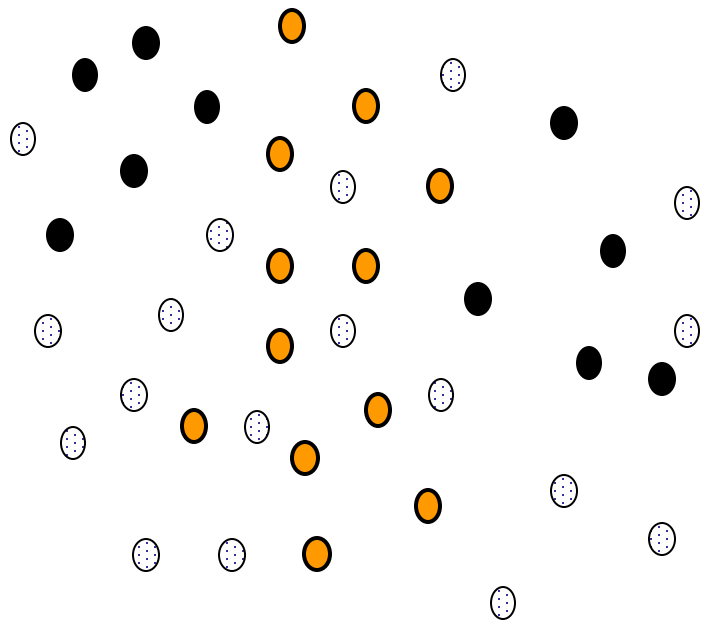
- Wireless Sensor Networks
- Virtual Sensor Networks
  - Perform different tasks
  - Deployed in the same geographical region
  - Involve dynamically varying subset of sensors nodes or users
  - Better resource efficiency through collaboration and resource sharing

Why dedicated WSNs?

- Limited sensing, processing and communication capabilities of the nodes
- Severe power constraints
- Cost



Virtual Sensor Networks



VSN-1 Nodes

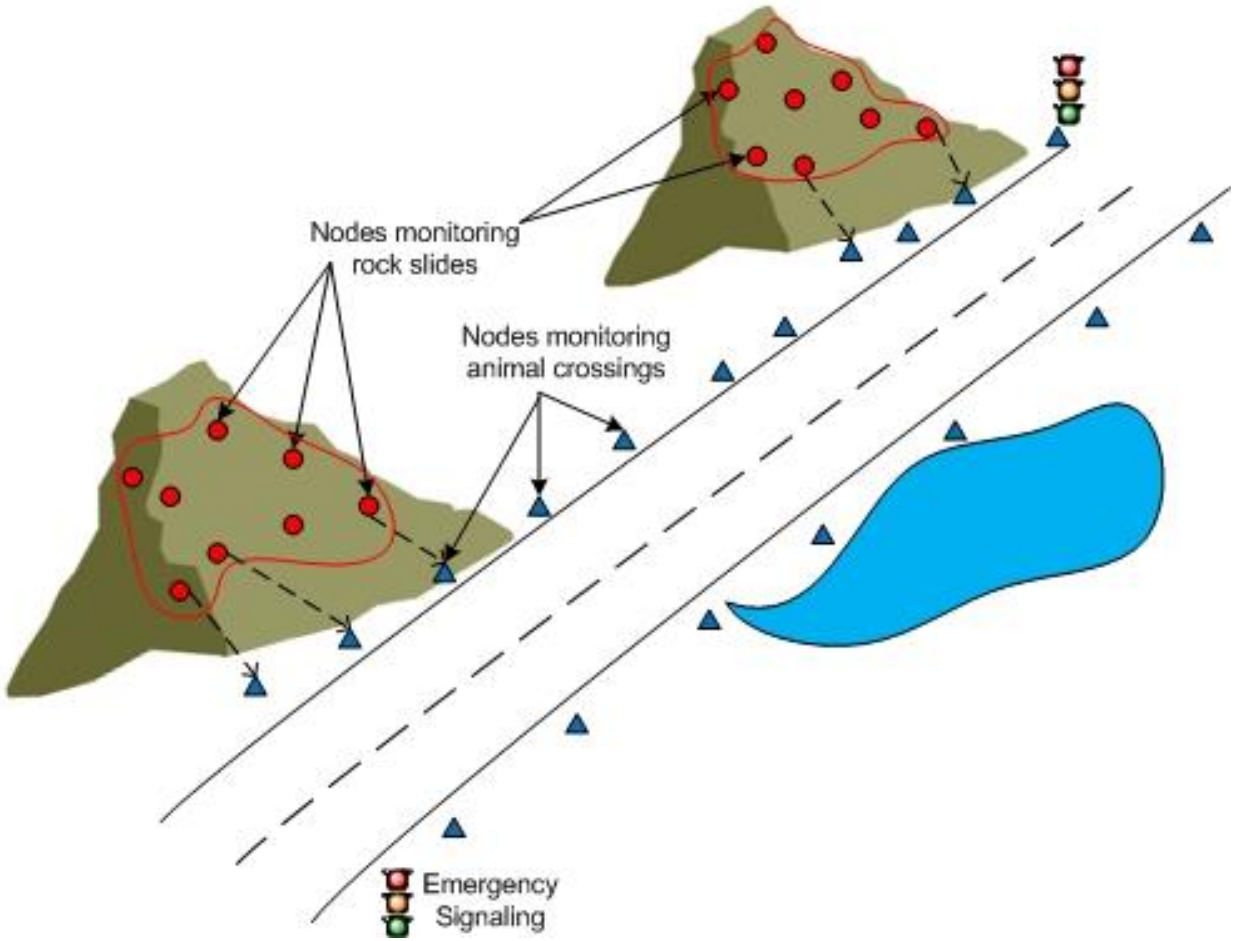
VSN-2 Nodes

Other Nodes

- VSN – formed by a subset of nodes dedicated to a certain task or an application
- Other nodes in physical network provide support functions to create, maintain and operate the VSN
- Multiple VSNs on a single WSN
- Membership in VSN may be dynamic

Jayasumana, Han, & Illangasekare, "Virtual Sensor Networks," Proc. ITNG'07

Ex 1: Geographically overlapped applications

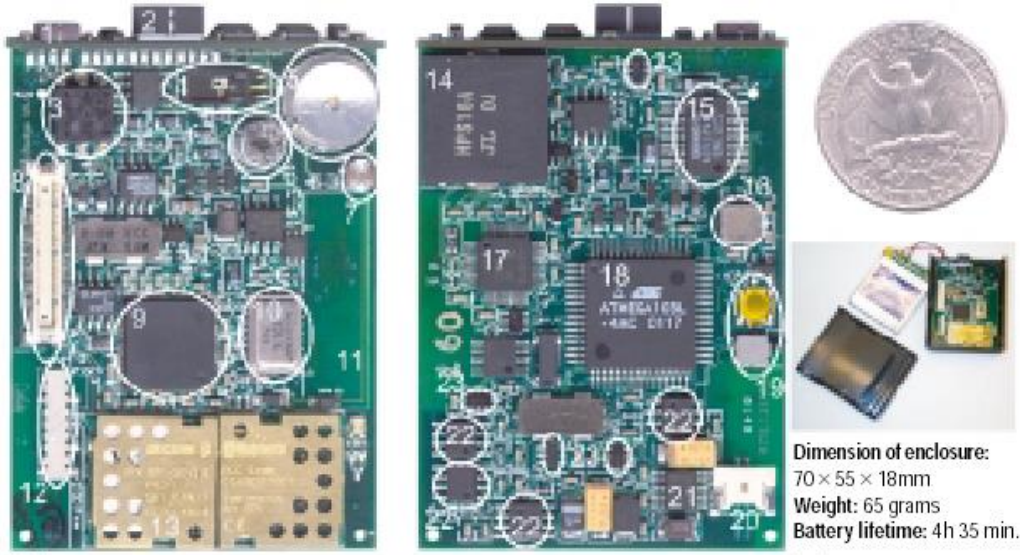


SenseApp 2008

Ex 2: Multi-functional sensor networks

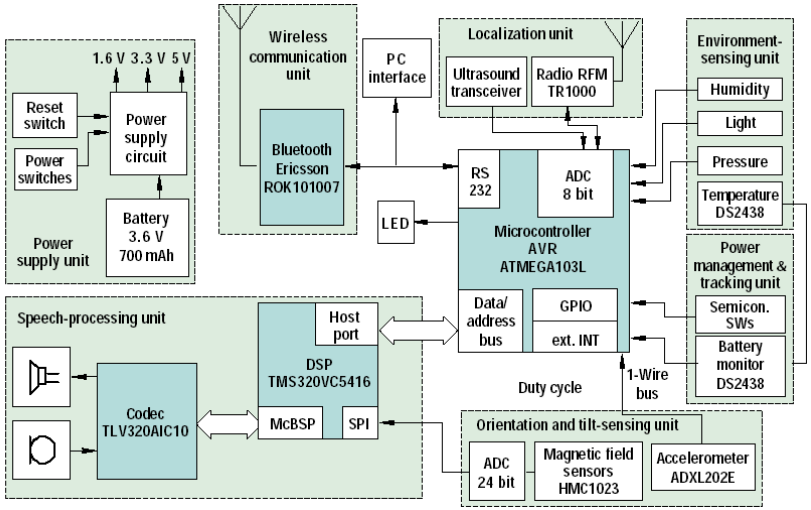
- One physical sensor network for different functions
  - Each node equipped with multiple sensors
  - Multiple applications

SmartKG iBadge platform (NESL/UCLA)



47 × 68 × 7 mm (1.85 × 2.78 × 0.28")

1. Accelerometer forx, y-axis
2. Magnetic field sensor
3. Pressure sensor
4. Humidity sensor
5. Ultrasound tranceiver
6. Microphone
7. Light sensor
8. Connector (SW download)
9. DSP
10. RFM radio (for localization)
11. PCB antenna for RFM radio
12. Blue tooth antenna
13. Blue tooth module
14. Loudspeaker
15. ADC magnetic field sensor
16. Accelerometer for x-axis
17. Codec chip
18. Microcontroller
19. Switches (Power, Reset)
20. Battery connector
21. Power supply
22. Battery monitors
23. Switches to functional units

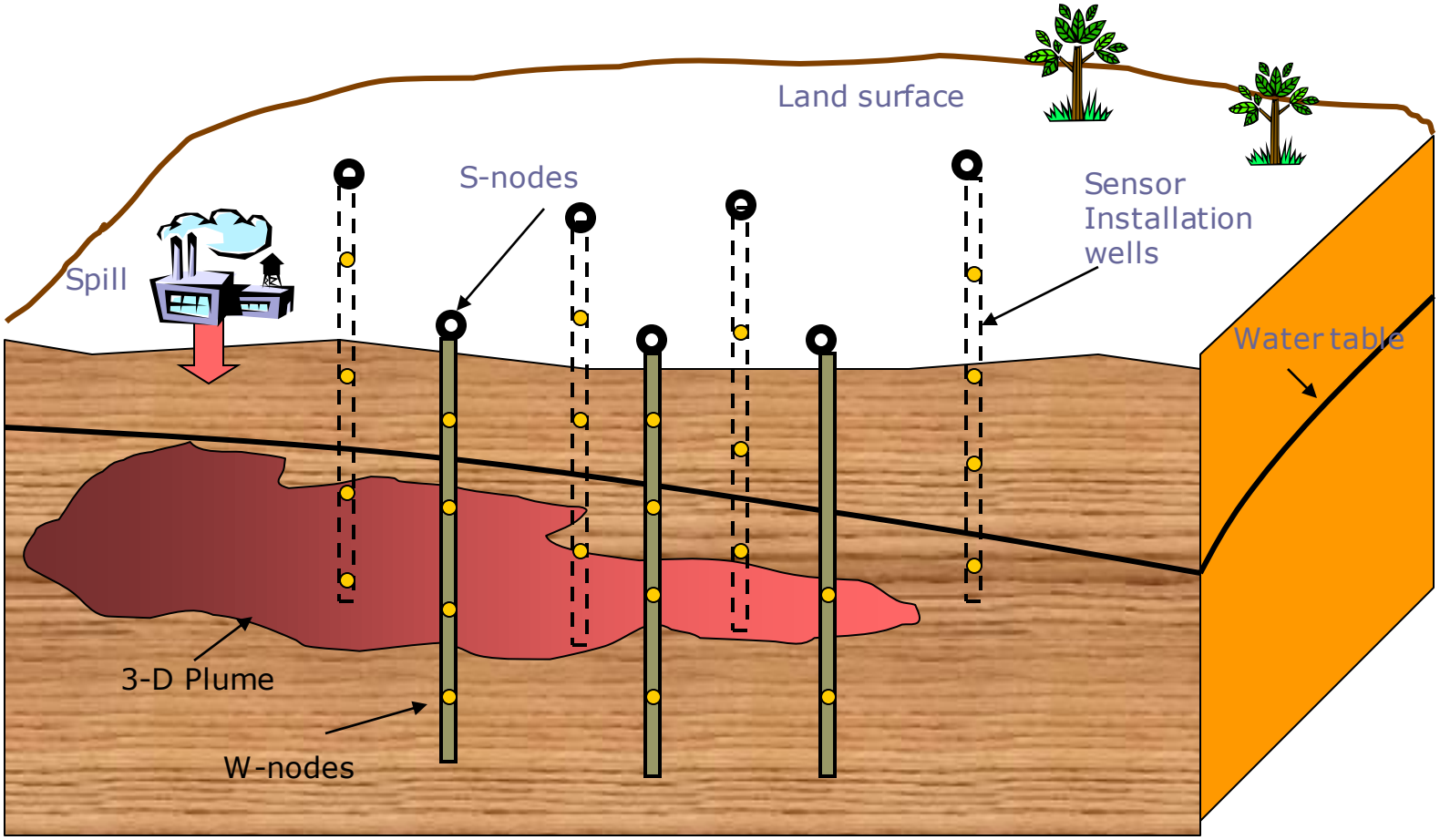


A. Savvides, Yale (ISPN/SPOTS 2005)

Ex 3: Dedicated applications

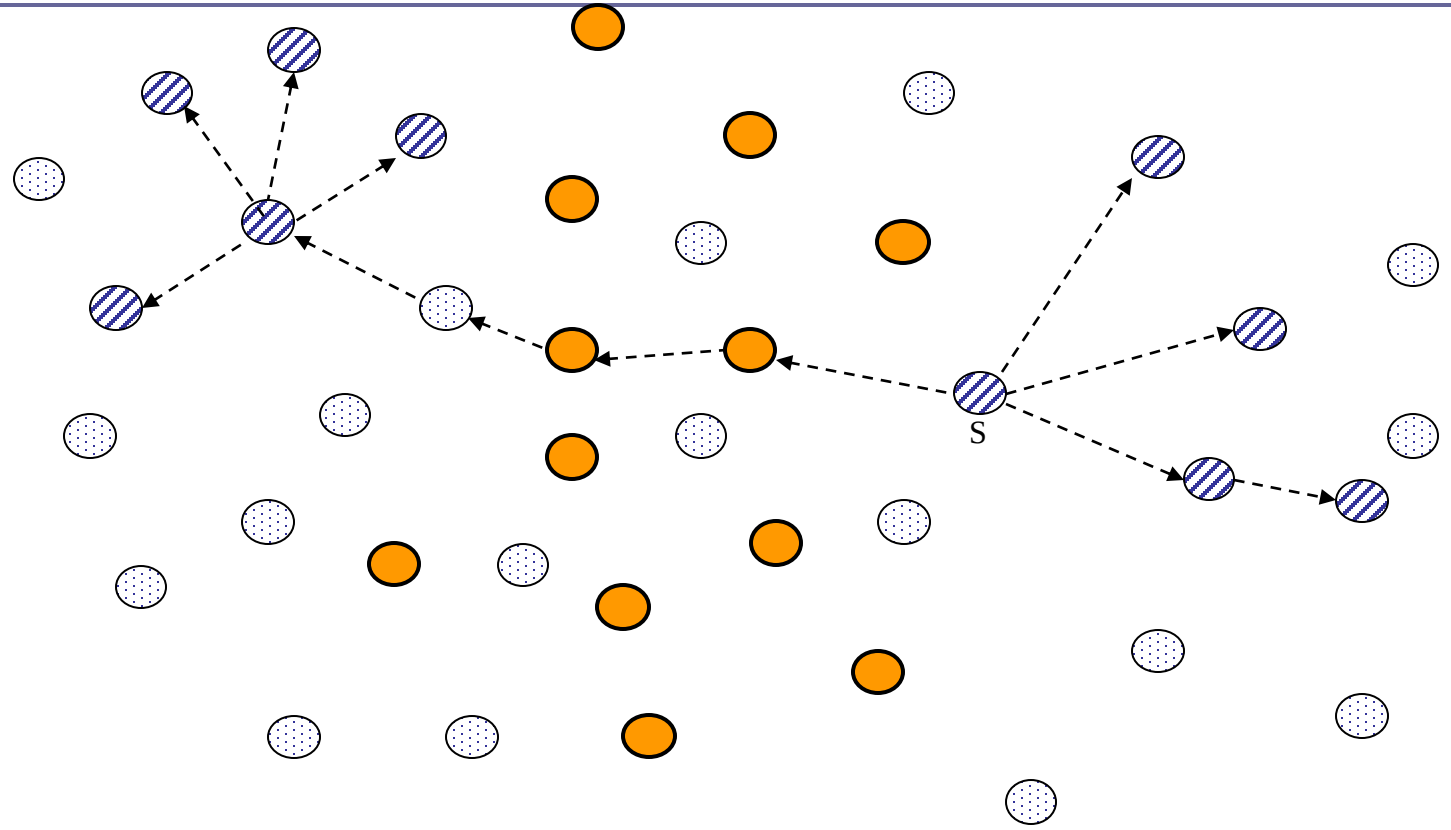
- There are some dedicated applications that will benefit from the VSN concept as well

Ex 3: Three-D plume tracking *(TDTP)*(CSU/COSM)



Jayasumana & Illangasekare, 2005

Virtual Sensor Networks



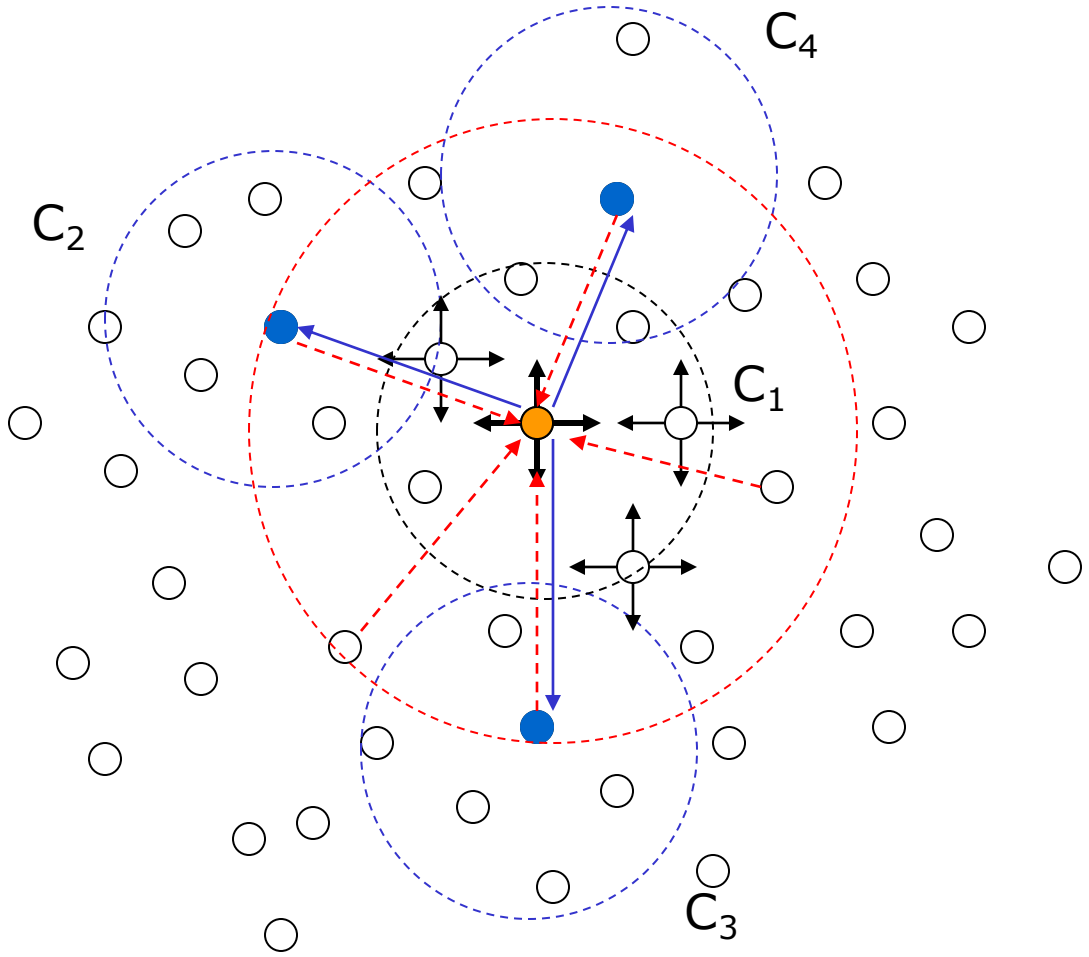
Motivation

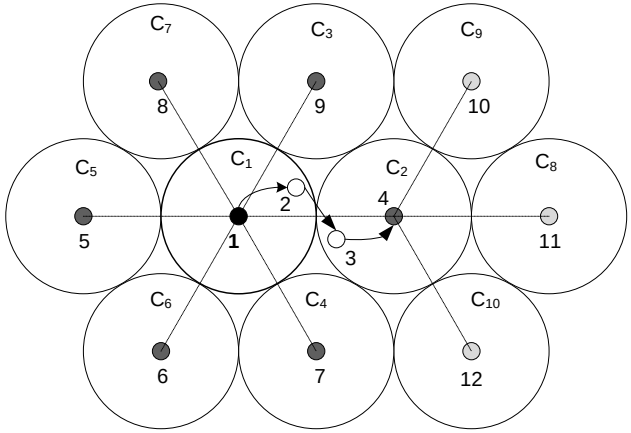
- Future collaborative/large-scale WSNs require some structure
- Security and privacy becomes critical
- Secure backbone
  - Dynamic distribution of cryptographic keys
  - Enhance secure upper layer functions

Key distribution in WSNs

- Dynamic key assignment based on PKI is not practical
- ECC based key distribution – *Du et al. (2007)*
  - Use several resourceful nodes – form backbone
  - Tamper proof nodes, location aware, geographic routing
- Key assignment through a trusted base station - *Shehab et al. (2005), Ibriq et al. (2007)*
  - High overhead, single point of failure
- Key pre-distribution
  - Random distribution – *Eschenauer et al. (2002), Chan et al. (2003)*
  - Combinatorial design – *Lee et al. (2005), Chakrabarti et al. (2006)*
  - Deployment knowledge based - *Simonova et al. (2003), Du et al. (2004)*
- Cluster membership is meaningless if nodes don't share at least one common key

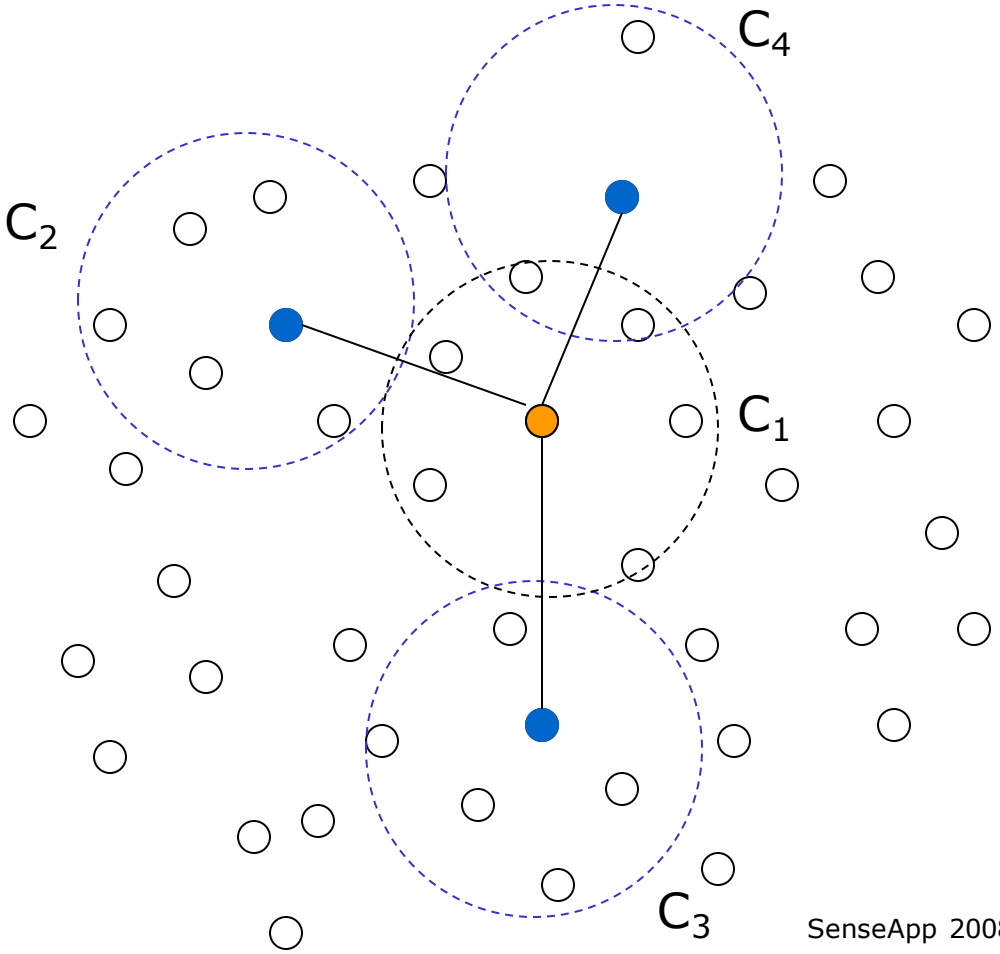
Generic Top-down Cluster & cluster tree formation (GTC) algorithm

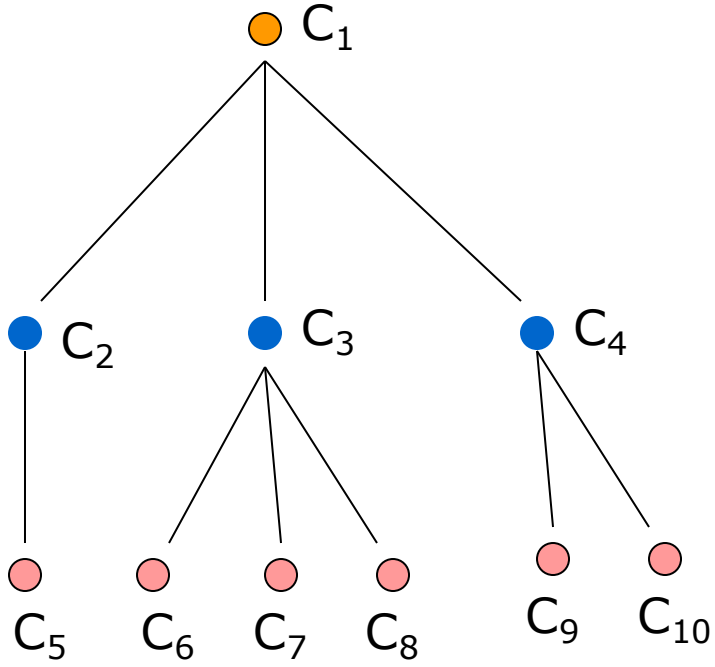




GTC - Cluster tree formation

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





- Individual links are secured through shared keys

Extended GTC

```
Form_Cluster(NID_CH, CID, delay, n_CCHs, hops_max, TTL_max, depth, keyIDs_CH)
1
2
                                    timeout_ACK)
5                   IF(ack_list = NULL)
6                   Join_Cluster()
7                   FOR i = 1 TO n_CCHs
8                   CCH_i ← Select_Candidate_CHs(ack_list)
9                   CID_i ← Select_Next_CID(i)
10                  delay_i ← Select_Delay(i)
11                  depth_i ← depth + 1
12                  Rqst_Form_Cluster(CCH_i, CID_i, delay_i, n_CCH, hops_max,
TTL_max, depth_i)
```

Form_Cluster(NID_{CH}, CID, delay, n_{CCHs}, $hops_{max}$, TTL_{max}, depth, $keyIDs_{CH}$)

```
Join_Cluster()
13                  Lstn_Bcast_Cluster(NID_CH, CID, hops_max, TTL_max, TTL, depth,
16IF (hops ≤ hops_max AND my_CID = 0)

17                  IF(Common_Keys(my_keyIDs, keyIDs_CH) ≠ NULL)
18                                  my_CID ← CID
                                    my_CH ← NID_CH
22                  IF(TTL > 0)
23                  Wait(Random(time_backoff))
24                  Fwd_Bcast_Cluster(NID_CH, CID, hops_max, TTL_max, TTL,
keyIDs_CH)
25                  IF (hops ≤ hops_max)
26                                  Exit()
27                  ELSE
28                  IF(Common_Keys(my_keyIDs, keyIDs_CH) ≠ NULL)
29
                    IF(Wait_Lstn_Neighbors(Random(time_backoff)) = FALSE)
30                  Send_ACK(my_NID, hops, p1, p2, my_keyIDs)
31                  IF(Lstn_Form_Cluster(CCH, CID, delay, n_CCHs, hops_max,
                                    TTL_max, depth, timeout_CCH) = TRUE)
32                  Form_Cluster(my_NID, CID, delay, n_CCHs, hops_max, TTL_max,
                                    depth, my_keyIDs)
33                  Exit()
34                  Join_Cluster()
```

IF(Common_Keys(my_keyIDs, $keyIDs_{CH}$) ≠ NULL)

- Each Cluster Head (CH) broadcast its list of key IDs
- Cluster members and candidate CHs respond only if they share a key(s)
- If not, find a CH with a common key(s)
- Intermediate nodes don't need to have common a key(s)

Extended GTC (cont.)

- GTC algorithm is extended to form a secure cluster tree
- Provisioning for secure communication
  - Integral goal of the cluster formation process
  - Reduced overhead and overall improvement in efficiency
  - Ensure cluster tree is fully connected
- Extended GTC algorithm
  - Independent of the pre key-distribution scheme and network topology
  - No prior neighborhood information, location awareness, or time synchronization
  - Form uniform and circular clusters
  - Control breadth and depth of the cluster tree

Outline

- Motivation
- Key distribution in WSNs
- Extended Generic Top-down Clustering algorithm
- Performance analysis
- Conclusions & future work

Performance analysis – Simulator

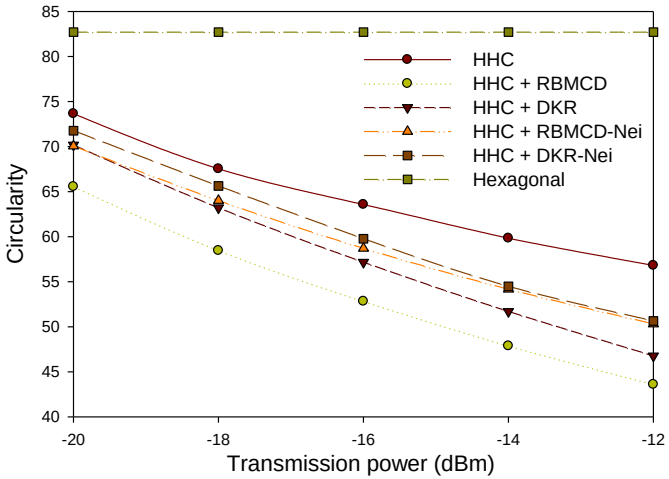
- 2 key pre-distribution schemes
  - Deployment Knowledge based Random key distribution (DKR)
    - *Du, Deng, Han, Chen, and Varshney (2004)*
    - 120 keys per node and key pool of 100,000
    - Shares 4-5 keys with its neighbors
  - Random Block Merging in Combinatorial Design (RBMCD)
    - *Chakrabarti, Maitra, and Roy (2006)*
    - 120 keys per node and key pool of 4,470
    - Shares 3-4 keys with its neighbors

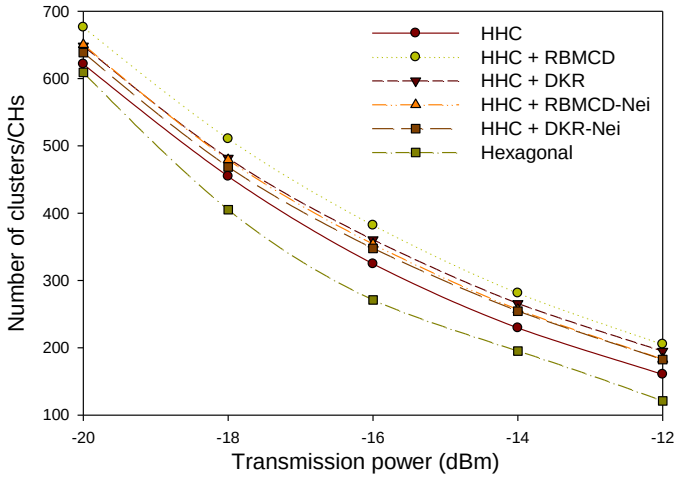
Simulator (cont.)

- Discrete event simulator was developed using C
- 5000 nodes in a circular region with a radius of 500m
  - 2-D Gaussian based node distributed to facilitate DKR
- 100 samples, each with a different random seed
- Log-distance path-loss model with a fading factor of 2.2

$$Circularity = \frac{1}{m}\sum_{i=1}^{m}\frac{no\ of\ \ nodes\ in\ cluster\ \ i}{no\ of\ \ nodes\ in\ the\ range\ of\ \ CH_i} \times 100$$

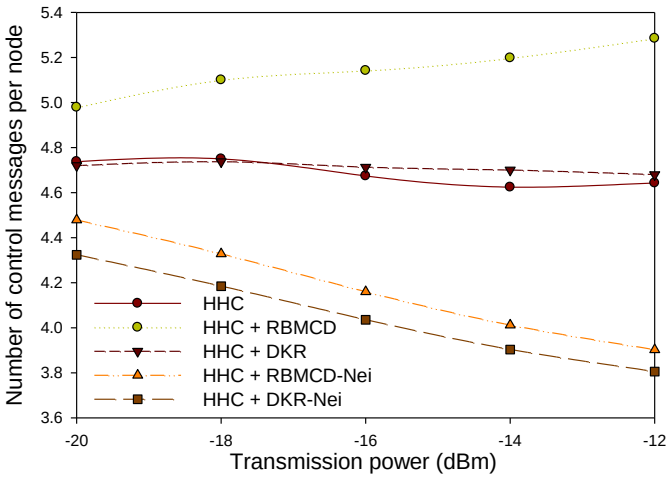
Performance analysis – Cluster characteristics

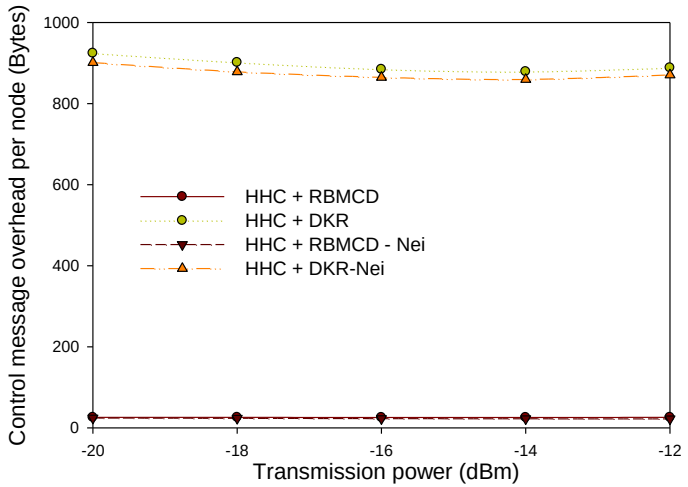




- Neighborhood information improves cluster characteristics
- Lower local connectivity of RBMCD is affecting cluster characteristics
- Circularity reduces with transmission range or density

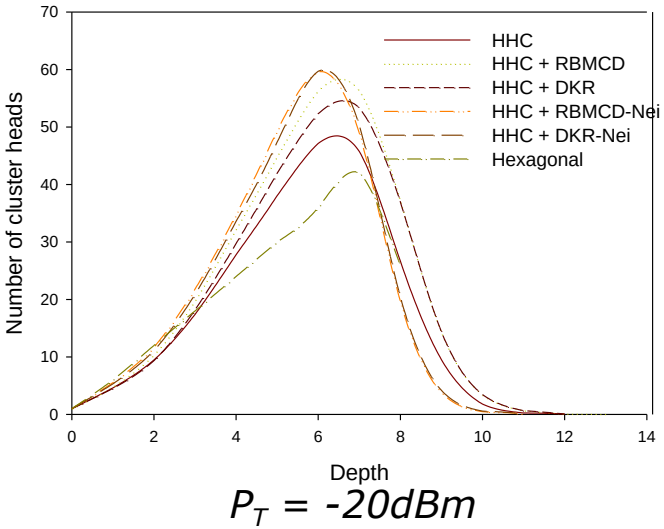
Performance analysis – Backbone formation overhead



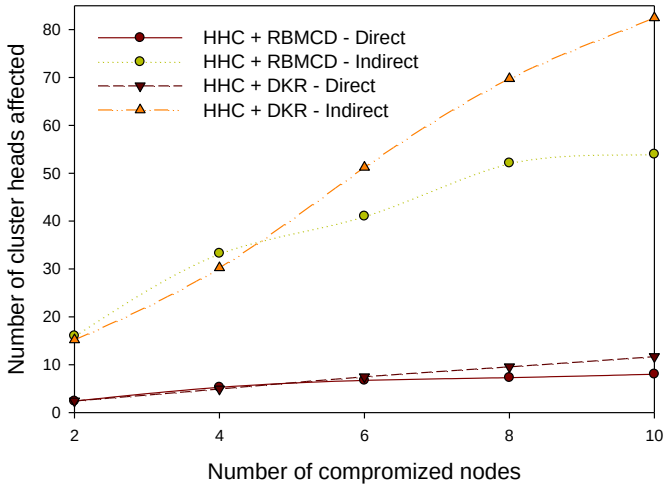


- RBMCD - Lower local connectivity increases overhead
- Neighborhood information reduce overhead
- Key ID distribution mechanism has a significant impact
  - RBMCD – 1 key ID per group of keys (4 groups with 30 keys)
  - DKR – 1 key ID per key (120 keys)

Performance analysis – Backbone and compromised nodes



$P_T = -20dBm$



$P_T = -20dBm$

- Direct impact of node compromise is not significant
- Indirect impact is significant
  - Depends on which node(s) got compromised
  - Disasters, if occur closer to the root node – particularly in DKR like schemes

Summary & future work

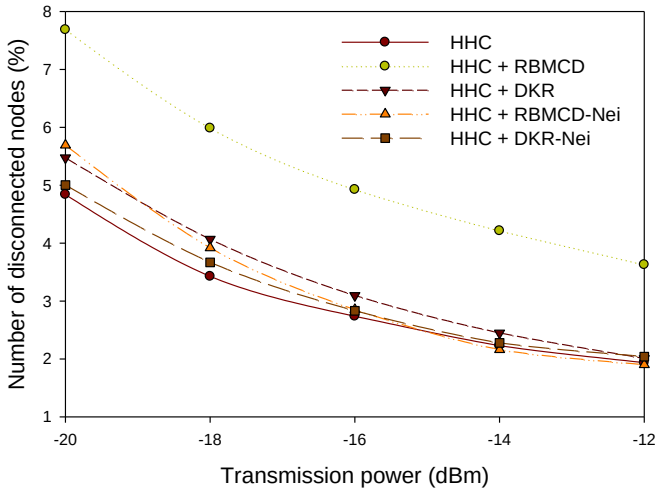
- Secure backbone formation is an integral part of GTC
- Facilitates secure key distribution and communication
- Independent of the key pre-distribution scheme
  - Better the local connectivity → better the results
  - Overhead is determined by key ID sharing mechanism
- Algorithm retains most of the desirable characteristics, while building the secure backbone
- Node compromise is a major issue in hierarchical WSNs

- Future work
  - More robust schemes for hierarchical WSNs
  - Dynamic key distribution scheme for collaborative WSNs

Questions ?

Thank You…

Disconnected nodes



- Generally 1-5% of the nodes are disconnected in GTC
  - Due to random node placement, collisions during cluster formation phase, etc.
  - Increase with transmission power
  - Developed a 2-step cluster and tree optimization phase
- Lower local connectivity of the key pre-distribution scheme can further disconnect nodes