Tree Bound on Probabilistic Connectivity of Underwater Sensor Networks

Md Asadul Islam and Professor Ehab S. Elmallah

University of Alberta

{*mdasadul,elmallah*}*@ualberta.ca*

August 24, 2014

Overview

Why UWSNs?

UWSNs fuelled by many important underwater sensing applications and services such as

- Scientific applications
- Industrial applications
- Military and homeland security applications
- Humanitarian applications

Challenges of the underwater communication

Radio Communication

- suffer strong attenuation in salt water
- short distances (6-20 m) and low data rates (1 Kbps)
- require large antennas and high transmission power

Optical Communication

- strongly scattered and absorbed underwater
- limited to short distances (40 m)

Acoustic Communication

- suffers from attenuation, spreading, and noise
- very long delay because of low propagation speed.
- it is most practical method upto now

Node Deployment Strategy

Static Deployment

- nodes attached to underwater ground, anchored buoys, or docks
- are not subject to move

Semi-mobile Deployment

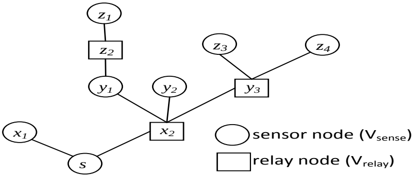
- nodes attached to a free floating buoy
- subject to small scale movement

Mobile Deployment

- composed of drifters with self/noself mobile capability
- are subject to large scale movement
- maintaining connectivity is important to perform localization, routing etc.

Node Locality Sets

- $V = V_{sense} \cup V_{relay}$ the set of nodes in a given UWSN
- The geographic area considered rectangles of a superimposed grid layout.
- At time T, each node x can be in any one of a possible set of grid rectangles denoted $\mathrm{Loc}(x) = \{x[1], x[2], \ldots\}$.
- Node x can be grid rectangle $x[i]$ with a certain probability $p_x(i)$.
- Truncate some locality sets of low probability for convenience thus, $\sum_{x[i] \in \mathrm{Loc}(x)} p_x(i) \leq 1$, if $\mathrm{Loc}(x)$ is truncated.



x_1	0.2, 0.5, 0.1	0.07, 0.05, 0.08
x_2	0.25, 0.3	0.16, 0.29
y_1	0.2, 0.3, 0.1	0.1, 0.15, 0.15
y_2	0.25, 0.35	0.25, 0.15
y_3	0.16, 0.18, 0.14	0.3, 0.22
z_1	0.6, 0.2	0.1, 0.1
z_2	0.5, 0.25, 0.2	
z_3	0.25, 0.36	0.14, 0.25
z_4	0.7, 0.15, 0.15	
s	0.18, 0.1, 0.03	0.34, 0.1, 0.14

Figure: Network with Probabilistic locality set

Node Reachability

- node x can reach node y if the acoustic signal strength from x to y (and vice versa) exceeds a certain threshold value.
- we set $E_G(x[i], y[j]) = 1$ iff the two nodes x and y can reach each other if they are located anywhere in their respective rectangles $x[i]$ and $y[j]$.
- connectivity between x and y is ignored if they can reach each other at some (but not all) pairs of points in their respective rectangles.
- ignoring connectivity in such cases results in computing lower bounds on the network connectivity, as required.

Problem Definition

We define four probabilistic connectivity problem. They are

- A-CONN problem
- S-CONN problem

the A-CONN and S-CONN problems

Definition (the *A-CONN* problem)

Given a probabilistic network G with no relay nodes, compute the probability $Conn(G)$ that the network is in a state where the sink node s can reach all sensor nodes.

Definition (the *S-CONN* problem)

Given a probabilistic network G with no relay nodes, and a required number of sensor nodes $n_{req} \leq |V_{sense}|$, compute the probability $Conn(G, n_{req})$ that the network is in a state where the sink node s can reach a subset of sensor nodes having at least n_{req} sensor nodes.

Network State

- A probabilistic graphs arises when network is in some particular network states.
- A state S of V can be specified by $\{v_1[i_1], v_2[i_2], ..., v_n[i_n]\}$
- If locations are independent, we have $Pr(S) = \prod_{v_\alpha \in V} p_{v_\alpha[i_\alpha]}$.
- In the *A-CONN* problem, a state is **operating** if the sink s can reach all sensor nodes in V_{sense} .
- Similarly, in the *S-CONN* problem, a state S is **operating** if the sink node s can reach a n_{req} sensor nodes.

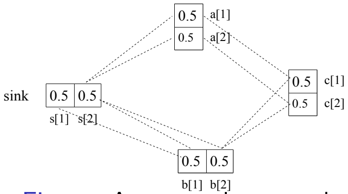


Figure: An example network

k-Trees and Partial k-Trees

Definition

For a given integer $k \geq 1$, the class of k-trees is defined as follows

1. A k-clique is a k-tree.
2. If G_n is a k-tree on n nodes then the graph G_{n+1} obtained by adding a new node adjacent to every node in k-clique of G_n.

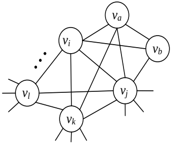


Figure: a fragment of a 3-tree

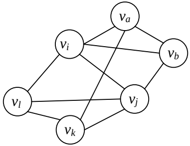


Figure: a fragment of partial 3-tree

A partial k-tree is a k-tree possibly missing some edges and a *k-perfect elimination sequence* (k-*PES*) of G is an ordering $(v_1, v_2, \ldots, v_r)$ of $v(G)$

Example

For the graph G in figure , $(v_a, v_b, v_i, v_j, v_k, v_l)$ is a 3-*PES*.

Probabilistic Connectivity of Tree Network

- Notation and Definition
- Pseudo-code for $\mathrm{S\text{-}CONN}$ problem on tree network
- Running time Analysis

Notation and definition

- *type*(x): *type*$(x) = 0$ and 1 if x is a relay node and a sensor node respectively.
- $n_{sense}(X)$: # of sensor nodes in a given subset of nodes $X \subseteq V$.
- $n_{relay}(X)$: # of relay nodes in a given subset of nodes $X \subseteq V$.
- $n(X) = n_{sense} + n_{relay}$.
- $n_{sense,min}(X)$: The minimum number of sensor nodes in a given subset $X \subseteq V$. So,

$n_{sense,min}(X) = max(0, n_{req} - n_{sense}(\overline{X}))$

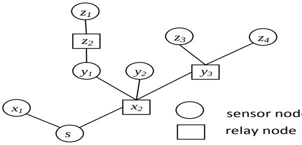


Figure: A tree network

Example

In figure , consider node x_2. $n(V_{x_2}) = 8$ where $n_{sense}(V_{x_2}) = 5$ and $n_{relay}(V_{x_2}) = 3$. Assuming $n_{req} = 5$ in an instance of the S-CONN problem then $n_{sense,min}(V_{x_2}) = 3 = n_{req} - n_{sense}(\{s, x_1\}) = 5 - 2$.

SR-CONN problem

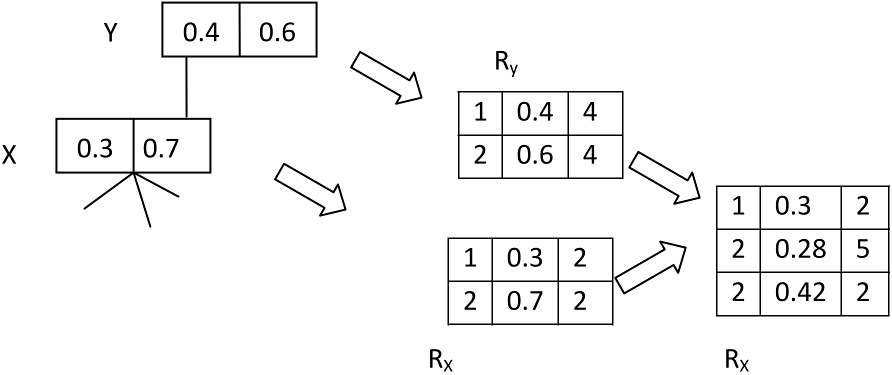


Figure: Table merge in Tree networks and $n_{req} = 5$

Pseudo-code for function Conn

Function $\mathrm{Conn}(G, T, n_{req})$
Input: the *SR-CONN* problem where G has a tree topology T with no relay leaves
Output: $Conn(G, n_{req})$

1. **foreach** (node x and a valid location index i)
   set $R_x(i, type(x)) = 1$
2. **while** (T has at least 2 nodes)
   {
3. Let y be a non-sink leaf of T, and $x = parent(y)$
4. **foreach** (key $(i, count) \in R_y$) $R_y(i, count) *{=} p_y(i)$
5. set $R'_x = \phi$
6. **foreach** (pair of keys $(i_x, count_x) \in R_x$ and $(i_y, count_y) \in R_y$)
   {
7. $count = \min(n_{req}, count_x + count_y)$
8. **if** ($count < n_{sense,min}(\{x\} \cup V_y \cup_{z \in DCH(x)} V_z)$) **continue**
9. $R'_x(i_x, count) \mathrel{+}= R_y(i_y, count_y) \times R_x(i_x, count_x) \times E_G(x[i_x], y[i_y])$
   }
10. set $R_x = R'_x$; remove y from T
   }
11. return $\sum_{s[i] \in \mathrm{Loc}(s)} R_s(i, n_{req}) * p_s(i)$

Running time

Let n be the number of nodes in G, and ℓ_{max} be the maximum number of locations in the locality set of any node.

Theorem

Function Conn solves the $\mathrm{SR\text{-}CONN}$ problem in $O(n \cdot n_{req}^2 \cdot \ell_{max}^2)$ time

Proof. We note the following.

- Step 1: storing the tree T require $O(n)$ time.
- Step 2: the main loop performs $n - 1$ iterations. Each of Steps 3, 5, and 10 can be done in constant time.
- Step 4: this loop requires $O(n_{req} \cdot \ell_{max})$ time.
- Step 6: this loop requires $O(n_{req}^2 \cdot \ell_{max}^2)$ iterations. Steps 7, 8, and 9 can be done in constant time.

Thus, the overall running time is $O(n \cdot n_{req}^2 \cdot \ell_{max}^2)$ time.

Theorem

Function Conn solves the $\mathrm{AR\text{-}CONN}$ problem in $O(n \cdot \ell_{max}^2)$ time

Simulation Results for A-CONN problem

- Test Networks
- Running Time
- Connectivity for different partial k-trees
- Effect of subgraph selection method

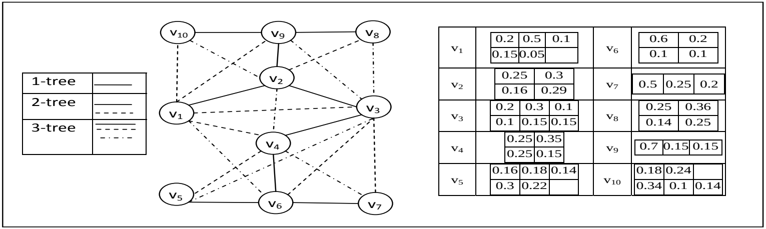


Figure: Network G_{10}

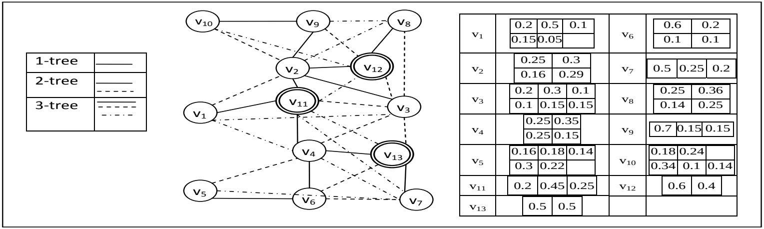


Figure: Network $G_{10,3}$

Connectivity for different partial k-trees

k	Network G_{10}	Network G_{12}	Network G_{15}
1	0.62	0.336	0.82
2	0.71	0.36	0.99
3	0.75	0.37	1

Table: Connectivity lower bounds using different partial k-trees

Running Time

k	Network G_{10}	Network $G_{10,3}$
1	90	110
2	1000	6000
3	6000	8000

Table: Running time in milliseconds

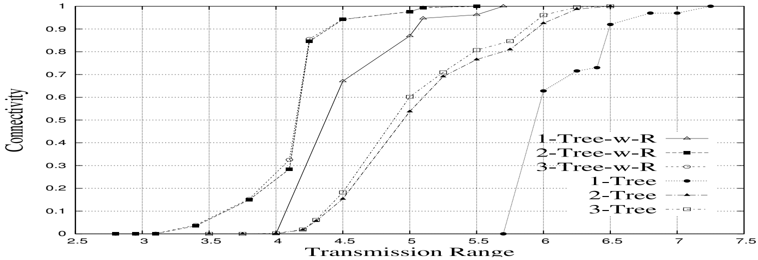
Effect of adding relay nodes

k	Network G_{10}	Network $G_{10,3}$
1	0.30	0.86
2	0.54	0.96
3	0.60	0.98

Table: Connectivity with respect to k

Effect of adding relay nodes for various node R_{tr}:

- Each obtained curve exhibits a notable monotonic increasing behaviour as R_{tr} increases.
- This behaviour is due to the appearance of more edges, and the potential increase in the probability of each edge as R_{tr} increases.
- Increasing R_{tr}, however, requires increasing node energy consumption.
- To achieve a desired $Conn(G)$ value, a designer may utilize the obtained curves to assess the merit of increasing R_{tr} versus deploying more relay nodes.



Concluding Remarks

- This thesis is motivated by recent interest in UWSNs as a platform for preforming many useful tasks.
- A challenge arises since sensor nodes incur small scale and large scale movements that can disrupt network connectivity.
- Thus, tools for quantifying the likelihood that a network remains completely or partially connected become of interest.
- the thesis has formalized 4 probabilistic connectivity problems, denoted $\mathrm{A\text{-}CONN}$, $\mathrm{S\text{-}CONN}$, $\mathrm{AR\text{-}CONN}$, and $\mathrm{SR\text{-}CONN}$.
- The obtained results show that all of the 4 problems admit polynomial time algorithms on k-trees (and their subgraph), for any fixed k.

Future Research Directions

- Investigating the applicability of our algorithm to some other classes of graph to be a worthwhile direction.
- It is interesting to analyze the delays incurred in typical data collection rounds.
- It is worthwhile to investigate area coverage assuming a probabilistic locality model of the nodes.

Thanks!