Quick Edge-Device Deployment for Connecting First Responders in Public Safety Networks

Ghazaleh Jowkar
School of Engineering and Technology
University of Washington
Tacoma, USA
jowkarg@uw.edu

Mohammadbagher Fotouhi
School of Engineering and Technology
University of Washington
Tacoma, USA
mfotouhi@uw.edu

Wei Cheng
School of Engineering and Technology
University of Washington
Tacoma, USA
uwcheng@uw.edu

***Abstract*—Keeping First Responders (FR) connected has the highest priority of communications in public safety networks. In this paper, we propose a method to quickly determine where to deploy Edge-Devices for connecting the first responders in the scenarios, where the infrastructure supports may be insufficient. As an FR can only carry very few numbers of edge-devices, our research goals are two folds: (i) Quickly decide where to deploy edge-devices and (ii) deploy as less number of edge-devices as possible. We model the problem as an Optimal (Minimal) CDS construction problem with time constraints. Specifically, we propose a novel real-time connected dominating set (CDS) construction algorithm for connecting first responders (FRs) in public safety networks. The proposed algorithm is a polynomial time approximate solution that can yield a solution in a given time, which is guaranteed by both the algorithm convergence speed and the designed stopping theory based rules. The approximation ratio to the optimal solution is jointly determined by the stopping rules, the required time, and the algorithm properties. The outputs from the algorithm can guide the FRs to deploy and/or select edge-devices in off-network environments to make them connected.**

***Index Terms*—Edge Computing; Public Safety Network; Connected Dominating Set, Stopping Theory**

I. Introduction

First responders may be in the environments, where there is no wireless connection among them and the base station. There is no fixed infrastructure or centralized management in wireless networks and the hosts may turn on, turn off or move around at any speed freely, thus the network topology changes dynamically, i.e. the routing protocols in such a network need to adapt quickly to the topology changes. The objective of this proposed technique is to design a relay or BS deployment or selection method, so that the FRs can quickly determine when and where to deploy relays or how to select relays to keep themselves connected in Public Safety Networks (PSN).

Each host can directly transmit messages to its neighbors within the transmission range. In this case, there is a link to every neighbor within the transmission range. If two hosts in the network are not within the transmission range of each other and they want to communicate, a multi-hop routing mechanism is required where some other hosts will relay the messages to the destination. All of these characteristics stimulate the use of the Connected Dominating Set (CDS) as a virtual backbone in the network.

We use a connected graph $G = (V, E)$ to represent an ad hoc wireless network. V is the set of mobile hosts in the network and E represents all the links in the network. We assume that all the hosts are deployed in a 2-D plane and their maximum transmission range are the same. Thus the resultant topology graph of the net- work is modelled as an undirected Unit Disk Graph (UDG). In the context of graph theory, we call a host as a node. A Dominating Set (DS) of a graph $G = (V, E)$ is a subset $V_{CDS} \subseteq V$ such that each node either belongs to V_{CDS} or is adjacent to at least one node in V_{CDS}. A CDS is a DS which induces a connected sub-graph. The nodes in the CDS are called the dominators, otherwise, dominatees. It is desirable to build a Minimum-sized Connected Dominating Set (MCDS). With the help of the CDS, the routing is easier since the search space for a route from a dominatee to another dominatee, is reduced only within the CDS. The construction of an MCDS in a UDG is proved to be NP-hard in [1].

In this paper we jointly utilize connected dominating set and stopping theory to design the solution. The theory of optimal stopping is concerned with the problem of choosing a time to take a given action based on sequentially observed random variables in order to maximize an expected payoff or to minimize an expected cost [2]. In our design, we partition the PSN area into several grids. We set the center of each grid as a candidate position for deploying relays or BS. In addition, if there exist some FRs and/or already deployed relays, they are also considered as candidate positions. We model the graph consisted of all the candidate positions and the edges among them representing the communication links. We design real-time minimal connected dominating set algorithm to connect FRs with minimal number of deployed/selected relays. As PSN may require real-time decision on relay deployment, we further utilized stopping theory to control the processing time within a required range. To construct and test our model we randomly generated Connected graphs $G(V, E)$ with different topology. Our experiments shows that we can solve for MCDS with in less than a given threshold time. We show that our model can find a CDS with small size nodes that can be as low as 5 percent of the input graph, and we can solve for MCDS without reaching the threshold time in 80 percent of the times.

II. Related works

Due to the fact that there is no fixed infrastructure or centralized management in ad hoc wireless networks, a Connected Dominating Set (CDS) of the graph representing the network is widely used as the virtual backbone of the network this idea was first introduced in [3] . Since finding a minimum CDS is NP-hard [1], variety of studies and models has been introduced to find an optimal solution for it. Algorithms in constructing CDS is divided to two categories, $Centralized$ algorithms that required network wide information and $Decentralized$ algorithms that requires local information only. and khuller [4] proposed two 2-phase centralized greedy algorithms to construct CDS in general graphs, which can be viewed as the process of growing a spanning tree T in several sequential rounds, this method is also called $MCDalgorithm$ has an $O(Log\delta)$ where δ is the maximum number of neighbors of a vertex. Wen et al [5] has proposed an algorithm considering the Maximal Independent Set (MIS) based on spanning tree where each vertex in MIS can be connected to the spanning tree with an additional vertex. This algorithm usually produces a larger CDS than MCDS. In [6] the CDS construction algorithm is based learning automata, and a near optimal solution is found based on a heuristic algorithm. A study of MCDS problem in Cognitive radio network is proposed in [7] where both Minimum CDS and Maximum network lifetime is considered where both centralized and localized algorithms are proposed. Extended Dominating Set (EDS) is studied in [8], and in [9] a Minimum Spanning Tree is constructed using PSO-Optimized Topology Control Scheme.

In the study of decentralized algorithms there are two main methods that are either Clustered-base or pure localized. Clustering algorithms are relatively slow convergence and have a constant approximation ratio. In cluster based algorithms first the network is divided into clusters then a $clusterhead$ is selected for each cluster. These $clusterheads$ are connected to form a CDS which is called MIS. [3], [10]–[12]. In localized algorithms status of each node depends on its $h-$hop topology (where h is small constant), in [13] set of hosts, which are called as $coordinators$, generate a CDS, and a host becomes a coordinator if it has two neighbors that are not directly connected.

III. System Model and Formulation

In this section we introduce the Connected Dominating Set (CDS) model along with our basic CDS construction method and the performance analysis. Then, a stopping theory based quick CDS construction method is proposed to satisfy the real-time requirement in PSN stopping Model. Finally, a real-time CDS construction framework is proposed along with the analysis of the factors that affect the algorithm's performance

A. CDS Model

Given a connected graph $G(V,E)$, where V is the node set in G and E is the edge set in G. A subgraph $G_{CDS}(V_{CDS},E_{CDS})$, is a Connected Dominating Set (CDS) of G iff G_{CDS} is connected and $\forall v_i \notin V_{CDS}$, $\exists v_j \in V_{CDS}$ s.t. $e_{i,j} \in E$. The objective of Minimal Connected Dominating Set (MCDS) problem is to find a G_{CDS} s.t. G_{CDS} is a CDS of G and $|V_{CDS}|$ is minimized.

To find a MCDS, we define the following graphs generated from node set V_M, $V_M \subseteq V$: (1) $G_M(V_M,E_M)$, $E_M = \{e_{i,j}$: $\forall i,j,\ v_i,v_j \in V_M$ and $e_{i,j} \in E\}$. (2) $\overline{G_M}(V,\overline{E_M})$, $\overline{E_M} = \{e_{i,j}$: $\forall i,j,v_i \in V_M$ and/or $v_j \in V_M$, and $e_{i,j} \in E\}$. The potential function f for a G_M is defined as $f(G_M) = |V| - N_C(G_M) - N_C(\overline{G_M})$, given a node set $D = \{v_d\} \subseteq V$, we define the profit of f after adding D to G_M as $\Delta f_D(G_M) = f(G_M \cup D) - f(G_M)$ Since the complexity of calculating $\Delta f_D(G_M)$ is greater than or equal to zero. Therefore, the potential function f is monotone non-decreasing. $f(G) = |V| - 2$, which is the maximal possible value for the potential function f given connected graph G, as $N_C(G) = 1$, $N_C(\overline{G}) = 1$, and $G \cup D = G$ (We use $N_C()$ to denote the number of connected components in a graph). Then, G_M is a connected dominating set of G iff $N_C(G_M) = 1$ and $N_C(\overline{G_M}) = 1$. Therefore, for constructing CDS via adding nodes one by one to G_M, we can stop when $f(G_M) = |V|-2$. As f is monotone non-decreasing, to reduce the number of nodes in V_M (for the best approximation to MCDS), the best strategy is to select the node D that can yield the maximal $\Delta f_D(G_M)$ in each step. The time complexity of brute-force searching for the best D in each step will totally be $O(|V|^2)$ times the time for calculate $\Delta f_D(G_M)$, denoted by $T_{\Delta f_D(G_M)}$ which is not sufficient for real time processing of of a network of FRs. We, therefore, will propose a stopping theory based strategy for quickly selecting the approximate optimal D in each step while considering the UEs' differences in Sec.III-B.

B. Stopping Model

To design a greedy method for constructing a CDS without brute-force search for the best D, which yields maximal $\Delta f_D(G_M)$ in each step, we could chose the v_i that has the following properties in priority order: (1) maximal potential dominating contribution degree of a node v_i for G_M , (2) bigger potential connecting contribution degree of a node v_i for G_M, and (3) bigger potential robust contribution degree of a node v_i for G_M .

Note that this greedy method cannot guarantee that the maximal $\Delta f_D(G_M)$. Therefore, if time allows, we would check more nodes for their $\Delta f_D(G_M)$ in each step so that the probability of reaching/approximating to the maximal $\Delta f_D(G_M)$ is maximized. Then, the next question is how many nodes we should check in each step before making a decision. This is where the Stopping Theory starts to chat in. In stopping theory, the basic rule is that checking new node should be stopped if the current profit is greater than the expected profit of checking more nodes.

Considering the time constrain and probability of finding the maximal $\Delta f_D(G_M)$ in each step, we define the profit function p will be, $p(v_i) = Time_{factor}(v_i) \times \Delta f_{v_i}(G_M) \times Weight_{factor}(v_i)$ We, then, apply the stopping rule in each step:

Stopping Rules: *We will stop checking the node's profit p after checking the ith node if* $p(v_i) \geqslant E(p(\overline{v_{i+1}}))$, where $E(p(\overline{v_{i+1}}))$ is the maximal expected profit p from checking the unchecked nodes in the same step.

Once we stop checking node in a step, the node v_i, which has been checked and has the maximal $\Delta f_{v_i}(G_M) \times Weight_{factor}(v_i)$, will be selected and added to G_M.

Now, we have the algorithm framework presented below. The detail designs of the factors including $Time_{factor}(v_i)$, $\Delta f_{v_i}(G_M)$, and $Weight_{factor}(v_i)$ will be presented in Sec. III-C.

Real-Time CDS Construction Algorithm Framework

Inputs:

$\overline{T}$: Time Constrain
$G(V, E)$: A Connected Graph
$W(v_i)$: $\forall v_i \in V$, The Node Importance (Weight)

Algorithm Framework:

```
V_M = ∅
While (N_C(\overline{G_M}) ≠ 1 or N_C(G_M) ≠ 1) {
  Sort the nodes in V \ V_M according to Weight_factor;
  While (\overline{T} has not been used up) {
     Check v_i ∈ sorted V \ V_M;
     If (p(v_i) ⩾ E(p(\overline{v_{i+1}}))) {
      break;}
  }
  D = {v_i : argmax(Δf_{v_i}(G_M) × Weight_factor(v_i))
            and v_i is available and has been checked};
  Add a v_i ∈ D to V_M;
  If (\overline{T} has been used up) {
    Add node from sorted V \ V_M one by one to V_M
    until N_C(\overline{G_M}) = 1 and N_C(G_M) = 1;
    break;}
}
```

Output:

G_M: The CDS of G

C. Factor Models

1) Time Factor: Given time $\overline{T}$, which is the maximal time that can be used to find a CDS, the time factor $Time_{factor}(v_i)$ is jointly determined by the remaining time T_{remain}, the time $T_{\Delta f}$ for calculating $\Delta f_{v_i}(G_M)$, the time T_{weight} for calculating $Weight_{factor}(v_i)$, $|V_M|$, and the current value of $f(G_M)$. We can predict the number of remaining steps $N^{step}_{remain} = f_{max} - f(G_M) + r(G)(|V \setminus V_M| - f_{max} + f(G_M))$ for finding a CDS s.t. $f(G_M) = f_{max}$. where $r(G)$ is the parameter for predicting N^{step}_{remain} given an input graph G. $r(G)$'s selection will be discussed in Sec. III-C2.

Once N^{step}_{remain} is known, we can calculate the time for getting a result from each remaining step as $T^{step}_{remain} = T_{remain}/N^{step}_{remain}$ Therefore, in each step, we can check at most $N^{step}_{check} = T^{step}_{remain}/(T_{\Delta f} + T_{weight})$ number of nodes. We then can formulate the time factor as the following:

$$Time_{factor}(v_i) = \begin{cases} 1 - (i-1)/N^{step}_{check} & : i \leq N^{step}_{check} \\ 0 & : i > N^{step}_{check} \end{cases}$$

where i is the ith checked node in a step.

2) Profit Factor: Will introduce the best known algorithm for calculating $\Delta f_{v_i}(G_M)$.

To calculate the expectation of the profit function, we assume that $\Delta f_{v_i}(G_M)$ is a random variable following Poisson distribution in the range $[0, f_{max} - f(G_M)]$, with a mean λ. $\Delta f_{v_i}(G_M)$ cannot be less than 0 or greater than $f_{max} - f(G_M)$. Note that the $r(G)$ is related to λ. As less number of steps may be needed when λ is bigger, we set $r(G) = 1 - \lambda/(f_{max} - f(G_M))$

3) Weight Factor: The node importance $W(v_i)$, the dominating contribution degree $Degree^{dominating}_{G_M}(v_i)$ and the potential connecting contribution degree $Degree^{connecting}_{G_M}(v_i)$ jointly determine $Weight_{factor}(v_i)$. To calculate the expectation of the profit function, we assume that the $Weight_{factor}(v_i)$ is a random variable following uniform distribution in the range $(0, 1)$.

IV. Performance Evaluation

In this section we will perform performance evaluation by simulations and quantitative analysis on our system model. We first evaluate the impact of the Connectivity degree of the input graph (The Network) on CDS size as well as computation time. Then, we evaluate the impact of number of FRs of the network on both factors.

A. Connectivity degree Impact

The connectivity degree of the graph is determined by the communication range and the physical distance of nodes. Connectivity is the average of the number of nodes that can be reached by each node and it varies with the communication range.

We ran total of six experiments on networks with sizes of 30 and 50 nodes, where for each experiment the number of FRs and network size is kept fixed and the connectivity degree is the changing variable. We ran each experiment 100 times and used the average of the trial for performance analysis.

1) Connectivity degree impact on the CDS size: As mentioned, in this subsection we analysis the impact of connectivity degree on CDS size. Figure 1 shows the changes of CDS size in respect to the connectivity degree changes. The figure is comparing the CDS size changes in three scenarios. In all scenarios network size is fixed at 30 nodes and the number of FRs is fixed to 3, 9, and 18 for scenario one, two, and three respectively, and the Connectivity degree is changing from 2 to 20 with the step of 0.07 increment.

According to Figure 1, the CDS size is highly affected by changes in Connectivity degree. By increasing the connectivity degree of the network, the CDS size is decreasing significantly and after passing the threshold of the connectivity degree all graphs will merge into the same minimal point regardless of the number of FRs.

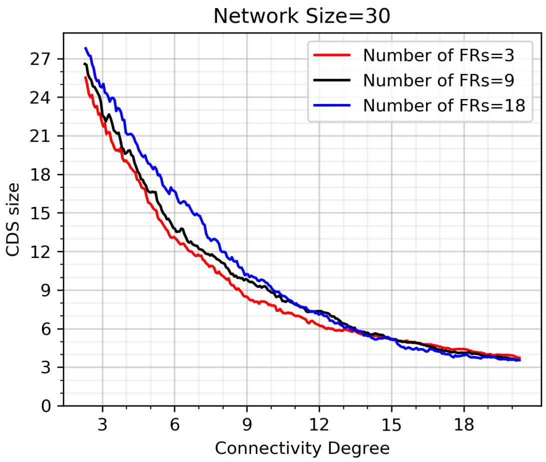

Figure 1: CDS size changes in respect to the Connectivity degree changes. Network size=30, Compared 3 scenarios when the Number of FRs is equal to 3, 9, and 18.

2) Connectivity degree impact on time: In this comparison we analysis the impact of Connectivity degree on computation time. In the same set of experiment in subsection IV-A1 we recorded the computation time. We first analysis to see, on how many instances we reached the maximum time threshold to solve for CDS. Then, we also check to see, on average, how long does it take to solve for CDS without considering the maximum time thresholds. Figure 2 shows the number of instances (%) that the computation time reached the maximum time threshold to solve for CDS in respect to the connectivity degree in networks with 30 nodes. Figure 3 is comparing the computation time (regardless of maximum threshold occurrences) in the same experiments. The figures are comparing the Time factors changes in same scenarios explained in IV-A1.

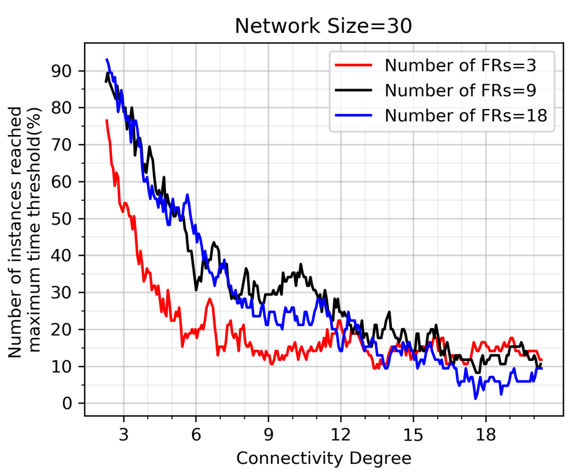

Figure 2: Number of instances that the computation time reached maximum threshold(%) in respect to the Connectivity degree changes. Network size=30, Compared 3 scenarios when Number of FRs is equal to 3, 9, and 18.

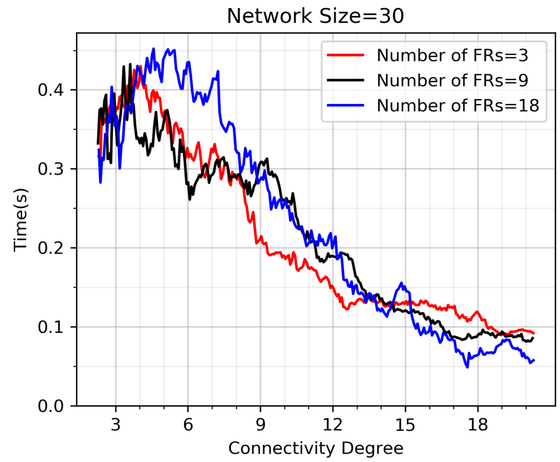

Figure 3: Computation Time(s) in respect to the Connectivity degree changes. Network size=30, Compared 3 scenarios when Number of FRs is equal to 3, 9, and 18.

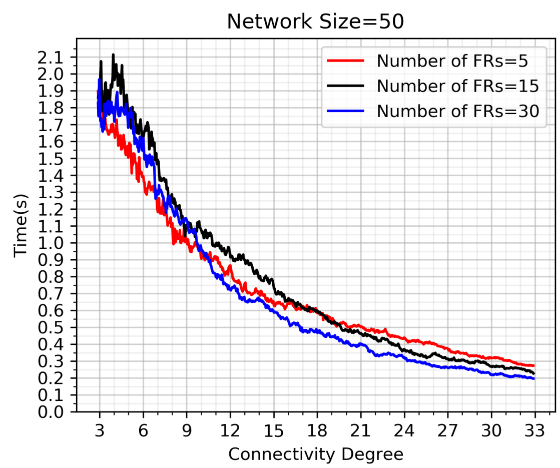

Figure 4: Computation Time(s) in respect to the Connectivity degree changes. Network size=50, Compared 3 scenarios when Number of FRs is equal to 5, 15, and 30.

According to figures 3 time is affected by both factors of Connectivity degree and the number of FRs. In both networks we can observe that if the connectivity degree is more than 5, by increasing the connectivity degree, the computation time is descending. Also, we can see that in each network after passing a certain threshold the computation time, regardless of the number of FRs will merge to the minimum computation time for a given network size. The same scenario holds for number of instances that reached maximum time threshold. Figure 2 shows, by increasing the connectivity degree after the same threshold, the percentage of time will merge to 10% regardless of the size of network or the number of FRs. By comparing the simulation result in IV-A we can see that for every network size we can find an optimal threshold for the connectivity degree where we will have the minimal computation time. We can see the same threshold when we are analysing the Percentage of maximum time occurrences as well as the CDS size.

B. Number of FRs Impact

In this section we want to analysis the impact of changes in number of FRs on our model. We ran total of six experiments on networks with sizes of 30 and 50 nodes, where for each experiment the connectivity degree and network size is kept fixed and the number of FRs is the changing variable. We ran each experiment 100 times and used the average of the trial for performance analysis.

1) Number of FRs impact on Size of CDS: In this subsection we analysis the impact of the number of FRs in the input graph on the CDS size. similar to previous tests, in this simulation we analyzed two sets of experiments with networks size at 30 and 50 nodes.

We observed that in networks with smaller connectivity degrees, number of FRs has minor impact on the CDS size. In both network sizes, we can find a threshold level for the number of FRs, where the the CDS size reaches the minimum possible size at each connectivity degree regardless of the number of FRs. On the other hand, when the connectivity degree is higher the number of FR does not affect the CDS size.

2) Number of FRs impact on Time: In the same set of experiments the computation time was recorded, In this comparison we analyze the impact of number of FRs on computation time. We first analyze to see, on how many instances we reach the maximum time threshold to solve for CDS. Then, we check to see on average how long does it take to solve for CDS without considering the maximum time thresholds. Figure 5 and 6 are showing the number of instances (%) that the computation time reached the maximum time threshold to solve for CDS in respect to the number of FRs in networks with 30 and 50 nodes respectively. Figure 7 and 8 are comparing the computation time (regardless of maximum threshold occurrences) in the same experiments. The figures are comparing the Time factors changes in same scenarios explained in IV-B1.

According to these observations, time is affected by both factors of Connectivity degree and the number of FRs. In networks with smaller Connectivity degree, as the number of FRs is increasing computation time and Time percentage are increasing and then decreasing until reaching to a steady descending behavior or a steady constant value. We consider this point as a threshold for number of FRs where from that point by increasing the number of FRs the computation time will either decrease or at it's minimum. The observed Threshold value for number of FRs in networks sizes of 30 and 50 is observed to be at the 25% of the network size.

Our observation from figures 7 and 8 shows that time is affected by both factors of Connectivity degree and the number of FRs. In both networks Connectivity degree limits the minimum computation time and the change in number of FRs determines the maximum computation time per connectivity degree. however, in both networks, after a Threshold, the computation time reaches a constant value regardless of the changes in the number of FRs. A similar scenario holds for number of instances that reached maximum time threshold. Figure 5 and 6 shows that the minimum percentage is determined by the connectivity degree. However, by increasing the Number of FRS we have a really large increment in the time percentage, after the threshold value for the Number of FRs is passed, the percentage of time will have a descending behavior as the Number of FRs increases, until it reaches a constant point. By comparing the simulation result in IV-A we can see that for every network size we can find an optimal Threshold for the connectivity degree where we will have the minimal computation time. Our observation shows that there is a threshold value for the number of FRs in each network

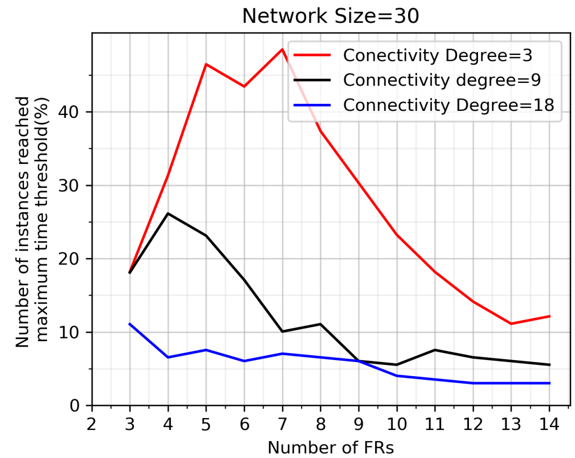

Figure 5: Number of instances that the computation time reached maximum threshold (%) in respect to Number of FRs. Network size=50, Compared 3 scenarios when Connectivity degree is equal to 10, 20, and 40.

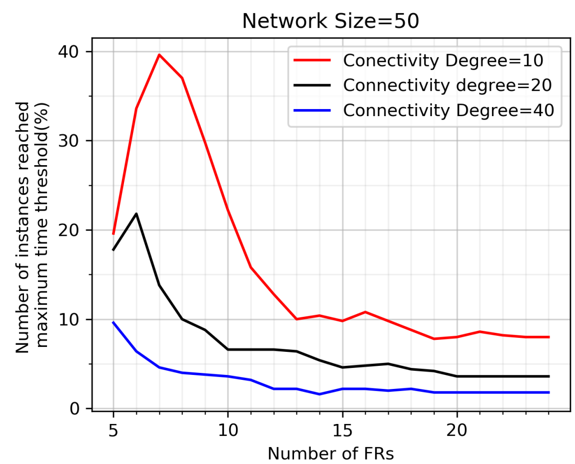

Figure 6: Number of instances that the computation time reached maximum threshold (%) in respect to Number of FRs. Network size=50, Compared 3 scenarios when Connectivity degree is equal to 10, 20, and 40.

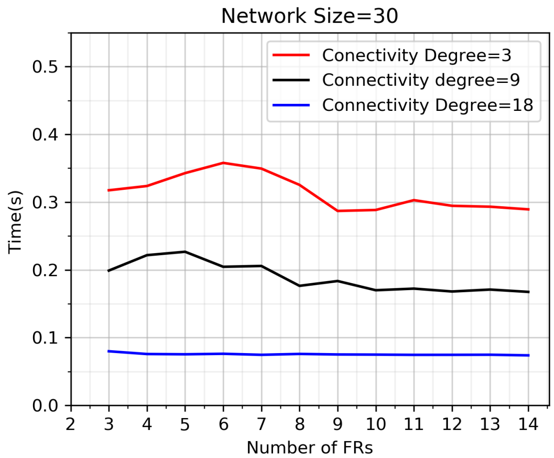

Figure 7: Computation Time(s) in respect to the Number of FRs. Network size=50, Compared 3 scenarios when Connectivity degree is equal to 10, 20, and 40.

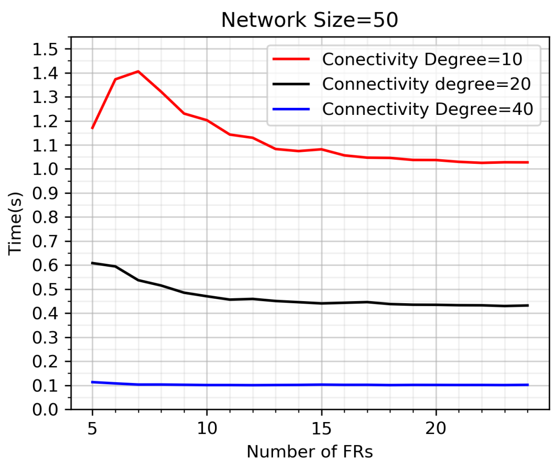

Figure 8: Computation Time(s) in respect to the Number of FRs. Network size=50, Compared 3 scenarios when Connectivity degree is equal to 10, 20, and 40.

size where in that threshold we have the optimal or close to minimal CDS size as well as the computation time.

V. Conclusion

In this paper, we propose a framework for relay/BS deployment in PSN. We initially analyzed the algorithm's performance in terms of approximation ratio and time bound. Further performance evaluation was conducted by simulations and quantitative analysis. We performed analysis on our model with analyzing the impact of connectivity degree and the number of FRs of the Network on the computation time of the algorithm as well as the size of the CDS graph. We've shown that for any given network size connectivity degree is a determining factor on both the CDS size and the computation time. For each network with a certain connectivity degree, the CDS size and the computation time have a minimum values regardless of the number of FRs. We can see that by increasing the connectivity degree the CDS size and the computation time reduced, and at a threshold point for each network size after a threshold value for the connectivity degree, we reach the minimum CDS size and minimum computation time. On the other, analyzing the number of FRs shows that, when the connectivity degree is less than it's threshold value, the number of FRs has some distorting impacts on both CDS size and the computation time. as a result, for every network size the number of FRs also has a threshold value, where at that point the CDS size and computation time starts to have descending behavior or reached their minimal value for a given connectivity degree.

Acknowledgement

This research was supported by NIST PSIAP 2017 Mission Critical Voice Award: Coverage, Capacity, and Resilience Enhancement in Limited PSN

References

[1] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[2] A. N. Shiryaev, *Optimal stopping rules*. Springer Science & Business Media, 2007, vol. 8.

[3] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[4] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[5] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2002, pp. 1597–1604.

[6] J. A. Torkestani, "An energy-efficient topology construction algorithm for wireless sensor networks," *Computer Networks*, vol. 57, no. 7, pp. 1714–1725, 2013.

[7] Z. Lin, H. Liu, X. Chu, Y.-W. Leung, and I. Stojmenovic, "Constructing connected-dominating-set with maximum lifetime in cognitive radio networks," *IEEE Transactions on Computers*, 2013.

[8] J. Yu, N. Wang, and G. Wang, "Constructing minimum extended weakly-connected dominating sets for clustering in ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 1, pp. 35–47, 2012.

[9] W. Guo, B. Zhang, G. Chen, X. Wang, and N. Xiong, "A pso-optimized minimum spanning tree-based topology control scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, p. 985410, 2013.

[10] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless networks*, vol. 1, no. 3, pp. 255–265, 1995.

[11] S. Basagni *et al.*, "Distributed clustering for ad hoc networks." in *ispan*, vol. 99, 1999, pp. 310–315.

[12] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[13] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless networks*, vol. 8, no. 5, pp. 481–494, 2002.