Diversity-Inspired Clustering for Self-Healing MANETs: Motivation, Protocol, and Performability Evaluation

Ann T. Tai[†] Kam S. Tso[‡]
IA Tech, Inc.
Los Angeles, CA 90094

William H. Sanders
University of Illinois
Urbana, IL 61801

Abstract

Swarm systems, which typically comprise a large number of lightweight mobile components, must be capable of self-healing. In this paper, we propose a self-organizing, self-healing framework called "superimposed" clustering for such systems. The framework makes a significant departure from traditional clustering algorithms that apply a single policy to form clusters through iterations. Specifically, our superimposed clustering protocol (SCP) selects a pair of diversified clustering polices to simultaneously build two sets of clusters, which we view as two cluster layers with one on top of the other. Via redundancy shadowing, SCP is able to extract and combine the complementary portions of the two layers to form a clustered network such that the vast majority of nodes can be organized through a single round. Moreover, SCP exploits shadow redundancy to enable gracefully degradable clustering coverage to mitigate cluster damage caused by node failure, death, or migration. We present the notion of superimposed clustering by devising a protocol and conducting a performability evaluation.

1 Introduction

Fast-advancing micro- and nano-electro-mechanical technologies have enabled the creation of systems consisting of hundreds of small mobile components (e.g., micro-scale robots, rovers, and UAVs). Those systems are often called *swarm systems* (see [1, 2], for example).

Typical applications of swarm systems include micro-robot swarms for natural disaster rescue, relief, and recovery; micro-robot teams for detection of potential biological, chemical, or radiological threats to homeland security; and micro-rover groups for sample-return deep-space exploration missions. Not surprisingly, those applications are built over mobile ad hoc wireless networks (MANETs).

Since most of swarm system applications are mission-critical, robustness and self-healing properties are essential to such systems. In addition, their deployment fields are often harsh and hazardous, making swarm components vulnerable to failure. On the other hand, such systems are built upon a large number of nondedicated resources and have the ability to aggregate the limited capacities of individual resources to achieve big, but adaptive goals. This enables loss-tolerance and gracefully degradable performance [3, 4, 5]. In addition, adaptive goals and coordination among system components may necessitate proactive reclustering of the surviving components to maintain or restore system serviceability (see [6], for example). Since self-healing is typically achieved by a system's self-organizing capability, a clustering protocol (a self-organizing mechanism for scalable network communication) that supports gracefully degradable performance and that is adequately efficient to permit proactive reclustering is highly desirable.

While many important research results have been published through the past decade to address optimization problems in cluster formation, issues concerning how to make a clustering protocol support gracefully degradable performance remain largely untouched. In addition, most clustering algorithms are aimed at perfect clustering coverage. While their initial cluster formation is able to achieve perfect coverage, traditional clustering approaches typically require reactive cluster maintenance, as nodes may migrate, fail, or die due to power exhaustion. Such reactive maintenance or clusterhead reelection could result in service disruption and a ripple effect in reclustering [7].

With the above motivation, we propose a notion called *superimposed clustering*. Rather than search for an optimal clusterhead selection or guarantee perfect clustering coverage, our intent is to let the vast majority of nodes be organized in a predictable time frame. In turn, this enables reasonably frequent reclustering to keep surviving hosts organized — a proactive means for self-healing. In particular, our superimposed clustering protocol (SCP) applies two diversified clustering policies simultaneously to form two different sets of clusters, which we view as two cluster layers, with one on top of the other. Subsequently, via redundancy

[†] Ann T. Tai is now affiliated with WW Technology Group.
[‡] Kam S. Tso is now affiliated with iRise.

978-1-4244-7499-8 /10/$26.00 ©2010 IEEE

shadowing, an SCP is able to extract and combine the complementary portions in the two layers to enable significantly better clustering coverage in a single round. In addition to the performance advantage, the shadowed redundancies let cluster-damage mitigation be accomplished without service disruption, enabling gracefully degraded performance between two epochs of reclustering.

It is noteworthy that resource diversity is traditionally applied to validate computation correctness based on the convergence of computation results [8]. In contrast, SCP aims to apply diversified clustering policies to obtain results that are mutually compensating. While superimposed clustering is a general self-organizing framework for MANETs, we investigate the framework in this paper based on an instance of SCP. Among other choices for a pair of clustering criteria (of which the combination will enable the realization of superimposed clustering), we select the well-known maximum ID and minimum ID clustering policies [7].

The remainder of the paper is organized as follows. Section 2 provides the fundamentals of clustering and the system model. Section 3 explains the SCP algorithm. Section 4 presents a performability evaluation. Section 5 describes related work. The concluding section discusses the significance of the framework of superimposed clustering.

2 Fundamentals

2.1 Terminology and Basics of Clustering

A cluster can be viewed as a unit disk with a radius equal to the *center node's transmission range*. As a result, any non-center nodes in a cluster are one-hop neighbors of the center node. The center node and its neighbors are called *clusterhead* (CH) and *clustermembers* (CMs), respectively, while a node that has one-hop neighbors in two or more clusters can become a *gateway* (GW) node. After an autonomous cluster formation, only CHs and GWs participate in the intercluster communication; CMs may talk to each other either directly or via their CHs. As the backbone of a cluster-based network consists only of the CHs and GWs, system-wide information dissemination can be done far more efficiently than with flat flooding.

For simplicity of illustration, we use the term "neighbor" to refer to a one-hop neighbor. In addition, the words "node," "host," and "component" are used interchangeably in the remainder of this paper. Finally, when we say "a CH loses its serviceability," we mean that a CH fails, dies, or abruptly departs from its cluster.

2.2 System Model

As mentioned earlier, our intent is to devise a clustering framework for systems consisting of a large number of micro-scale hosts with low to moderate mobility. Examples are collaborating micro-robot or micro-rover swarms, micro-satellite constellations, and micro-UAV formation flying in which UAVs have low relative mobility (in addition to group mobility). While all the hosts are allowed to move to any locations in the deployment field, some may exhibit model-based statistical preferences [9] in mobile behavior, to improve the chances of reaching an application-specific goal (a typical mechanism used in self-stabilizing systems).

Since mobile hosts can circumvent asymmetric link problems via open- or closed-loop power control, all the hosts presumably have the same transmission range. And since memory density has doubled every 18 months per Moore's law, swarm hosts are able to have a large amount of memory with no increased size, weight, or energy consumption. In swarm systems, local coordinators (i.e., CHs in a cluster-based network) are typically ordinary hosts instead of superpower entities. Thus, minimizing clusterhead count is not essential, but energy conservation is important. As various lightweight localization algorithms are now available to allow an autonomous host to compute its distances to other hosts or landmarks, a swarm host is capable of being location-aware even if it is not equipped with a GPS (see [10], for example).

Additionally, mobile-host collaboration usually takes place among nearby peers, so that periodic neighborhood probing is often required for collaboration planning (e.g., to get enough hosts to lift a piece of rock). Consequently, reasonably frequent proactive reclustering is desirable for both performance and self-healing purposes [3]. As swarm systems plan to adopt self-adaptive spread spectrum techniques to circumvent the crowded spectrum problem in ad hoc wireless networks [11], radio resource allocation will not be a problem for reclustering.

3 Superimposed Clustering

3.1 Concept

We first let $N_1(v)$ denote the node set that includes all the one-hop neighbors of node v (excluding v itself). Using the notation, we can then define the two-hop neighborhood of v as follows: $N_2(v) = N_1(v) \cup (\cup_{u \in N_1(v)} N_1(u))$.

Further, we define two subsets of $N_1(v)$:

$\hat{N}_1(v) \subseteq N_1(v)$, such that $\forall u \in \hat{N}_1(v), ID(u) > ID(v)$

$\check{N}_1(v) \subseteq N_1(v)$, such that $\forall u \in \check{N}_1(v), ID(u) < ID(v)$

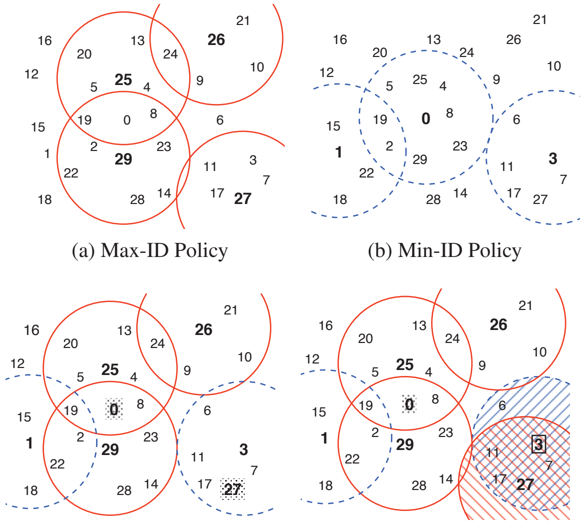
Then if we let $G_h(v)$ and $G_m(v)$ be the indicator variables to show whether v is a CH or CM of a cluster, respectively, the *combined MaxMin-ID clustering policy* of the proposed SCP can be defined as follows:

$$G_h(v) = 1 \Leftrightarrow N_1(v) \neq \emptyset \wedge (\hat{N}_1(v) = \emptyset \vee \check{N}_1(v) = \emptyset)$$
$$G_m(v) = 1 \Leftrightarrow N_1(v) \neq \emptyset \wedge (\exists u, u \in N_1(v), G_h(u) = 1)$$

Due to policy diversity, v will never be qualified to be a CH by both Max-ID and Min-ID criteria. On the other hand, the above definitions collectively reveal that superimposed clustering permits a node v to be a CH per one policy and a CM per the other. In turn, this implies the possibility that one cluster could completely overlap with another cluster or several other clusters, resulting in *cluster redundancy*.

Accordingly, the essence of superimposed clustering is to extract and combine the complementary portions of cluster layers through *redundancy shadowing*. While the concept of redundancy shadowing is explained in Section 3.2, below we informally describe the idea of SCP.

Figure 1 illustrates the idea of superimposed clustering based on an example scenario. For clarity, we let 30 nodes be represented by their unique IDs (0 through 29) and be randomly distributed such that the nodes with boldface IDs are the CHs. For simplicity, in the remainder of the text, we use the term "node n" to refer to a node whose ID is n. Figures 1(a) and 1(b) show that when Max-ID and Min-ID policies are applied alone, 6 and 11 nodes are left unclustered, respectively, upon the completions of the first rounds of the corresponding protocols.



(a) Max-ID Policy

(b) Min-ID Policy

(c) MaxMin-ID Policy

(d) Degraded Coverage

Figure 1: Superimposed Clustering: An Example

However, when the two policies are applied in parallel, only two nodes are left unclustered after a single round, as shown in Figure 1(c), a significant improvement of clustering efficiency over single-policy-based or sequential hybrid (i.e., conditional use of an alternative policy) approaches.

Note also that in Figure 1(c), the clusters in which the CHs are node 0 and node 27 disappear from the superimposed clusters. That is a result of redundancy shadowing, which is described in the next subsection.

3.2 Redundancy Shadowing

Informally speaking, redundancy shadowing is a process through which only non-redundant clusters will remain active in a cluster-based network. When we say a "non-redundant cluster," we mean that some node will become unclustered if that cluster is removed; whereas a "redundant cluster" can be removed without causing any node to become unclustered. The definitions can be stated in mathematical terms as follows:

Definition 1 *A cluster $N_1(v) \cup \{v\}$ in which $N_1(v) \neq \emptyset$ and v is the CH is said to be a "non-redundant cluster" if $\exists u, u \in N_1(v) \cup \{v\}, \forall w, w \in N_1(u) - \{v\}, G_h(w) = 0$.*

Definition 2 *A cluster $N_1(v) \cup \{v\}$ in which $N_1(v) \neq \emptyset$ and v is the CH is said to be a "redundant cluster" if $\forall u, u \in N_1(v) \cup \{v\}, \exists w, w \in N_1(u) - \{v\}, G_h(w) = 1$ and $N_1(w) \cup \{w\}$ is a non-redundant cluster.*

In addition, we use the term *preliminary N_2 knowledge* to refer to a node v's knowledge about 1) $\{ID(u) \mid u \in N_1(v)\}$, and 2) $\forall u, u \in N_1(v)$, $\{ID(w) \mid w \in N_1(u)\}$. Coupled with the clustering policy stated in Section 3.1, the preliminary N_2 knowledge will enable a node v that is not a CH candidate to determine whether it is affiliated with one and only one cluster (i.e., a sole cluster affiliation), which is stated formally in the following theorem:

Theorem 1 *A node v that is not a CH candidate will be able to determine whether it is affiliated with one and only one cluster if $N_1(v) \neq \emptyset$ and v has preliminary N_2 knowledge.*

Proof. If $N_1(v) \neq \emptyset$ and v has preliminary N_2 knowledge, then $\forall u, u \in N_1(v)$, v will be aware of whether the following condition holds:

$$\hat{N}_1(u) = \emptyset \vee \check{N}_1(u) = \emptyset$$

Hence, by checking each neighbor's CH qualification, v is able to identify its sole cluster affiliation. Q.E.D.

Furthermore, we define a special type of non-redundant cluster as follows:

Definition 3 *A cluster $N_1(v) \cup \{v\}$ in which $N_1(v) \neq \emptyset$ and v is the CH is said to be an "inclusion-type non-redundant cluster" if $\exists u, u \in N_1(v), N_1(u) \cup u \subset N_1(v) \cup v, \forall w, w \in N_1(u) - \{v\}, G_h(w) = 0$.*

As stated by Theorem 2 below, preliminary N_2 knowledge suffices the purpose of identifying a non-redundant cluster of that type (which we call "N_2-detectable" for simplicity).

Theorem 2 *An inclusion-type non-redundant cluster is an N_2-detectable non-redundant cluster.*

Proof. Based on its preliminary N_2 knowledge, v will be aware of the following scenario:

$$\exists u, N_1(u) \cup \{u\} \subset N_1(v) \cup \{v\},$$

which can be translated into:

$$\forall w, w \in N_1(u) - \{v\} \Rightarrow w \in N_1(v).$$

Then $\forall w, w \in N_1(u) - \{v\}$, v will be able to use its preliminary N_2 knowledge to determine if the following condition holds:

$$\hat{N}_1(w) \neq \emptyset \wedge \check{N}_1(w) \neq \emptyset$$

Hence, by learning that aside from v itself, none of u's neighbors are qualified to be CHs, v will be able to confirm that v itself is the CH of the sole cluster with which u is affiliated. Q.E.D.

On the other hand, when a CH candidate v confirms that its cluster is redundant, v will change its CH status to be passive. For simplicity, we use the word *withdraw* to refer to that type of CH status change. In addition, if v's confirmation is made according to Definition 2, then v's withdrawal will be *safe* (because the withdrawal will not cause any nodes to become unclustered). Conversely, when v is unable to confirm whether its cluster is redundant based on that definition, then its withdrawal will be *unsafe*.

To illustrate the idea of redundancy shadowing, we revisit the example shown in Figure 1. We observe that node 0 withdraws its CH candidacy (which is suggested by the Min-ID policy). The reason is that the cluster completely overlaps with two inclusion-type non-redundant clusters (centered at nodes 25 and 29), both of which contain nodes having sole cluster affiliations with them. Note that node 27 also withdraws its CH candidacy (which is suggested by the Max-ID policy), as all the nodes in the cluster, including the CH itself (node 27), are overlapped with an inclusion-type non-redundant cluster in which node 3 is the CH.

3.3 Algorithm

Just like a typical ID-based protocol, SCP forms clusters through two neighborhood profile exchanges (NPEs). In order to efficiently utilize the neighborhood knowledge, redundancy shadowing correspondingly has a two-stage realization. The algorithm is shown below and is explained according to the ordering of stages.

First Neighborhood-Profile Exchange

As shown in Algorithm 1, during the first NPE (line 4), each node sends a message that is a 3-tuple $\langle ID(v), N_1(v), CHstatus(v)\rangle$. The third field $CHstatus(v)$ indicates whether v is a CH candidate, and by which clustering policy v is qualified if v is a CH candidate. Since the nodes do not yet have N_2 knowledge yet at that point, CH candidates mark themselves as tentative CHs, i.e., $CH^{\text{tv}}_{\text{max}}$ or $CH^{\text{tv}}_{\text{min}}$.

Algorithm 1 Superimposed Clustering

```
 1: send(ID(v));
 2: $N_1(v) \leftarrow$ receive(integerSet);
 3: CHstatus(v) $\leftarrow$ MAXorMIN(ID(v),$N_1(v)$);
 4: send(ID(v),$N_1(v)$,CHstatus(v));                 // 1st NPE
 5: cRegistry(v) $\leftarrow$ digest(receive(3-tupleSet));
 6: neighborCHs(v) $\leftarrow$ identifyCHs(cRegistry(v));
 7: Stage-1-RedundancyShadowing(cRegistry(v));
 8: send(ID(v), CHstatus(v), neighborCHs(v));    // 2nd NPE
 9: cRegistry(v) $\leftarrow$ digest(receive(3-tupleSet));
10: Stage-2-RedundancyShadowing(cRegistry(v));
11: identifyGWs(cRegistry(v));
```

By receiving the messages from all its neighbors, v will obtain the preliminary N_2 knowledge, and will be able to identify 1) whether any of its neighbors are qualified to become CH candidates, and 2) how many such candidates are in v's neighborhood. The digested N_2 knowledge will then be stored into v's *cluster registry* ($cRegistry$).

In addition, the findings about the CH candidates in $N_1(v)$ will be recorded in an array $neighborCHs$ (line 6 of Algorithm 1). Each entry of the array is a pair $\langle HID, CHtype\rangle$ in which HID is the ID of a CH candidate in v's neighborhood and $CHtype$ indicates by which policy the CH candidate is qualified. Then v becomes ready to enter the stage-1 redundancy shadowing.

Stage-1 Redundancy Shadowing

The first stage of redundancy shadowing is a preparation stage. Equipped with the N_2 knowledge, a node v that is a CH candidate will be able to check (see Theorem 2) its potential cluster to see if it is an inclusion-type non-redundant cluster based on Definition 3 (see line 3 of Algorithm 2 below). If confirmed, v will mark itself as $CH^{\text{nr}}_{\text{max}}$ or $CH^{\text{nr}}_{\text{min}}$. Otherwise, its candidate status will remain tentative.

Algorithm 2 Stage-1 Redundancy Shadowing

```
1: Stage-1-RedundancyShadowing(3-tupleSet)
2: if $CHstatus(v) \neq null$ then
3:     if inclusionTypeNonRedundant(cRegistry(v)) then
4:         confirmNonRedundant(CHstatus(v));
5:     end if
6: else if | neighborCHs($v$) |= 1 then
7:     confirmNonRedundant(neighborCHs(v)[1].CHtype);
8: end if
```

If v is not a CH candidate, it will be able to learn (see Theorem 1), per the information in $neighborCHs$, whether v has a sole cluster affiliation. If the result is positive, v

will mark the CH of the sole cluster with which v is affiliated as the CH of a non-redundant cluster. The updates thus collectively elaborate the preliminary N_2 knowledge.

Second Neighborhood-Profile Exchange

The elaborated N_2 knowledge is then exchanged among the nodes in the second NPE (line 8 of Algorithm 1). Each message is again a 3-tuple. But the 3-tuple now is $\langle ID(v), CHstatus(v), neighborCHs(v) \rangle$. The second field may show its updated CH status if v is a CH candidate (e.g., from $CH^{\text{tv}}_{\text{min}}$ to $CH^{\text{nr}}_{\text{min}}$), while the third field will tell whether it has a sole cluster affiliation if v is not a CH candidate.

Stage-2 Redundancy Shadowing

Upon the completion of the second NPE, v first updates its cluster registry based on the elaborated N_2 knowledge obtained from the neighbors. Then, based on the organized knowledge, v carries out stage-2 redundancy shadowing, as shown in Algorithm 3.

Algorithm 3 Stage-2 Redundancy Shadowing

```
1: Stage-2-RedundancyShadowing(3-tupleSet)
2: if $CHstatus(v) \neq null$ then
3:     if redundantCluster(clusterRegistry(v)) then
4:         goShadow(CHstatus(v));
5:     else if nonRedundant(clusterRegistry(v)) then
6:         confirmNonRedundant(CHstatus(v));
7:     end if
8: end if
```

In particular, if a CH candidate v sees from its $cRegistry$ that all the potential CMs and v itself are included in the CM lists of non-redundant clusters, v will determine (based on Definition 2) that its potential cluster is redundant and update its status as a passive CH (line 4 of Algorithm 3). Conversely, if v sees in its $cRegistry$ that a node u that is not a CH candidate has only one CH candidate in its neighborhood and that the candidate is v itself, v will change its status to $CH^{\text{nr}}_{\text{max}}$ or $CH^{\text{nr}}_{\text{min}}$ (based on Definition 1).

Table 1 shows a sample cluster registry maintained by node 0 in the running example illustrated by Figure 1. Note that the entries without and with a mark $*$ are updated by the ends of stage-1 and stage-2 redundancy shadowing, respectively. For example, nodes 25 and 29 learn, based on preliminary N_2 knowledge, that their clusters are non-redundant (per Theorem 2). Thus, they mark themselves accordingly during the stage-1 redundancy shadowing. However, node 0 is unable to determine its status until after nodes 25 and 29 have made their "non-redundant" announcements in the second NPE. Therefore, node 0 changes its status from $CH^{\text{tv}}_{\text{min}}$ to $CH^{\text{ps}}_{\text{min}}$ (per Definition 2) during the stage-2 redundancy shadowing. Meanwhile, node 0 updates the $CHstatus$ fields of nodes 25 and 29 per what node 0 has learned from their announcements in the second NPE.

Table 1: A Cluster Registry

$ID(u)$	$N_1(u)$	CHstatus(u)	neighborCHs(u)	GW(u)
0	4,5,8 ...	$CH^{\text{ps}}_{\text{min}}$ *	$25^*, 29^*$	yes*
25	0,4,5 ...	$CH^{\text{nr}}_{\text{max}}$ *	0^*	no*
29	0,2,8 ...	$CH^{\text{nr}}_{\text{max}}$ *	0^*	no*
...	...	...	...	...

In addition, cluster registries make gateway selection straightforward. As shown in Table 1, GW qualification of each node can be determined per the multiplicity of $neighborCHs$. So after the second-stage redundancy shadowing, cluster registries will offer adequate information for gateway selection and route discovery on the backbone.

3.4 Discussion

As mentioned earlier, our goal is to provide swarm systems with better self-healing ability. For cluster-based swarm systems, self-healing is supposed to be accomplished through circumventing or mitigating the effects of excessive clustering coverage degradation. It is noteworthy that there are two types of coverage degradation. The first type is gradual coverage degradation caused by normal host mobility. With a protocol that is efficient enough and has a predictable clustering time, such as the single-round SCP, proactive reclustering would be more cost-effective than frequent cluster maintenance, in terms of preventing excessive system-wide coverage deterioration.

The second type is unexpected cluster damage caused by CH failure, death, or abrupt departure. In those circumstances, a passive CH will reactively take over from the CH that loses serviceability to let the affected cluster maintain gracefully degradable performance (and will adjust its position to progressively improve the coverage), until the next epoch of reclustering. Such a mechanism is indeed both reactive and proactive in nature in the sense that the passive CH is predesignated by SCP.

Moreover, in systems that apply superimposed clustering, passive-CH-enabled recovery is advantageous over event-triggered local CH reelection. The reason is that the latter is likely to 1) result in unexpected service disruption, and 2) be more costly, since all the hosts must be location-aware all the time (as otherwise the computation overhead of reelection would worsen the service disruption).

More importantly, passive CHs are *inherent redundancies* resulting from superimposed clustering (meaning that there is no extra cost for attaining the redundancy). Thus, the readiness of a passive CH enables the affected cluster to be recovered with little service disruption.

To further explain the concept, we could revisit the running example depicted in Figure 1. But now we assume that node 3 has failed. Hence, as illustrated in Figure 1(d), the passive CH, node 27, takes over from node 3.

4 Performability Evaluation

4.1 Measures and Formulation

In order to evaluate the effectiveness of SCP, we define three types of performability measures concerning 1) the efficiency of SCP with respect to the coverage achieved in a single round, 2) the likelihoods of clustering redundancy from the perspectives of a CH and a CM, and 3) the gracefully degraded coverage when a passive CH takes over from a CH that loses serviceability.

Clustering Efficiency

For the measure of the first type, we choose to evaluate $P^{scp}_{hm}(v)$, the probability that a node v will become affiliated with a cluster, as either a CH or a CM, through a single round of superimposed clustering. Clearly, the value of $P^{scp}_{hm}(v)$ will depend upon the value rank of its ID and the size of its neighborhood. Therefore, we must first evaluate the conditional probabilities. In addition, as node IDs are unique but are not necessarily consecutive, we give each node an ordinal number to rank the value of its ID. (Note the ordinal numbers are unique and consecutive.) Then, if we let $P^{scp}_{h}(v)$ denote the probability that node v will be clustered as a CH and assume that hosts are approximately uniformly distributed, we have a closed-form solution for $P^{scp}_{h}(v)$ as follows:

$$\begin{aligned} P^{scp}_{h}(v) &= \sum_{i=1}^{I} P(ID(v)=i) \sum_{n=1}^{I-1} P(|N_1(v)|=n \mid ID(v)=i) \\ &\quad P(G_h(v)=1 \mid ID(v)=i, |N_1(v)|=n) \\ &= \sum_{i=1}^{I} \frac{1}{I} \sum_{n=1}^{I-1} \binom{I-1}{n} \left(\frac{A_v}{A_t}\right)^n \left(1-\frac{A_v}{A_t}\right)^{(I-1)-n} \\ &\quad \left(\frac{\binom{i-1}{n}}{\binom{I-1}{n}} + \frac{\binom{I-i}{n}}{\binom{I-1}{n}} \right) \end{aligned} \tag{1}$$

where A_v and A_t are the sizes of the unit disk determined by the transmission range of v and the total field in which I nodes are uniformly distributed, respectively; the last two terms (in parentheses) are the conditional probabilities that v will be a CH per the Max-ID and Min-ID policies (given that the ordinal number of v's ID is i, and v has n neighbors), respectively.

To formulate the probability that v will become a cluster member, namely $P^{scp}_{m}(v)$, we translate this measure into the likelihood that there exists at least one node in v's neighborhood that will be qualified to be a CH (whose transmission range will cover v). As it is simpler to compute the complement of this likelihood, i.e., the probability that none of v's neighbors will be qualified to be a CH, we have,

$$\begin{aligned} P^{scp}_{m}(v) &= \sum_{n=1}^{I-1} \binom{I-1}{n} \left(\frac{A_v}{A_t}\right)^n \left(1-\frac{A_v}{A_t}\right)^{(I-1)-n} \\ &\quad (1-P^{scp}_{h}(v))(1-(1-P^{scp}_{h}(u))^n) \\ &= (1-P^{scp}_{h}(v)) \left(1-\left(1-\frac{A_v}{A_t}P^{scp}_{h}(u)\right)^{I-1}\right) \end{aligned} \tag{2}$$

Finally, $P^{scp}_{hm}(v) = P^{scp}_{h}(v) + P^{scp}_{m}(v)$.

The solution derivation for Eq. (2) (and Eqs. (3) and (4)) is based on the following mechanism: 1) to carefully expand and rearrange the terms; 2) to add and then subtract a term for $n = 0$ (so that the summation would start from $n = 0$ instead of $n = 1$); and 3) to simplify the solution using the Binomial theorem.

Note also that while the ordinal numbers of node IDs are unique, for a closed-form solution of Eq. (2), we assume that ID replacement is allowed (so that we can compute $P^{scp}_{h}(u)$ independently). Since the node populations of swarm systems are typically large (in the hundreds) and the ratio of the area of a cluster to that of the total deployment field is small, "ID collisions" (i.e., two nodes contend for the same ID) in a cluster are unlikely. Hence, the error resulting from the approximation would be insignificant.

For comparison, we also compute $P^{bl}_{hm}(v)$, the probability that v will be a CH or CM in the baseline system (which applies Max-ID or Min-ID as the sole clustering policy). The measure is formulated in a way that is analogous to that for $P^{scp}_{hm}(v)$. We skip the solution derivation due to space limitations.

Clustering Redundancy

For the measure of the second type, we first define $P^{scp}_{r|ch}(v)$ as the probability that a cluster formed by SCP in which v is the active CH will accommodate at least one passive CH. Due to policy symmetry, we can formulate such a measure as the conditional probability that cluster C contains at least one CH that is qualified by the Min-ID (or Max-ID) policy, given that the active CH v of C is elected per the Max-ID (or Min-ID) policy. Letting such a conditional probability be denoted by $P^{min}_{r|chmax}(v)$ (or $P^{max}_{r|chmin}(v)$), we have

$$\begin{aligned} P^{scp}_{r|ch}(v) &= P^{min}_{r|chmax}(v) = P^{max}_{r|chmin}(v) \\ &= \sum_{n=1}^{I-1} \binom{I-1}{n} \left(\frac{A_v}{A_t}\right)^n \left(1-\frac{A_v}{A_t}\right)^{(I-1)-n} \\ &\quad \left(1-\left(1-P^{min}_{h}(u)\right)^n\right) \\ &= 1-\left(1-\frac{A_v}{A_t}P^{min}_{h}(u)\right)^{I-1} \end{aligned} \tag{3}$$

Next, we define a measure for redundant coverage (from an arbitrary CM's perspective). Specifically, we evaluate the probability that v will be covered by a passive CH elected by the Min-ID (or Max-ID) policy, given that v is a CM of a cluster formed by SCP per the Max-ID (or Min-ID) policy. Letting such a conditional probability be denoted by $P^{scp}_{r|cm}(v)$, we have

$$\begin{aligned} P^{scp}_{r|cm}(v) &= \sum_{n=1}^{I-1} \binom{I-1}{n} \left(\frac{A_v}{A_t}\right)^n \left(1-\frac{A_v}{A_t}\right)^{(I-1)-n} \\ &\quad \left(1-\left(1-P_h^{min}(u)\right)^{n-1}\right) \\ &= 1-\left(1-\frac{1}{1-P_h^{min}(u)}\right)\left(1-\frac{A_v}{A_t}\right)^{I-1} - \\ &\quad \frac{1}{1-P_h^{min}(u)}\left(1-\frac{A_v}{A_t}P_h^{min}(u)\right)^{I-1} \qquad (4) \end{aligned}$$

Gracefully Degraded Coverage

For the last measure type, we define G (a random variable) as the fraction of the CMs (of the cluster where a CH u loses serviceability) that will be covered by the passive CH v at the point of takeover. Clearly, the value of G will depend upon the distance D (a random variable) between u and v.

As mentioned earlier, in self-stabilizing systems, statistical preferences are a mechanism used for improving the chances of achieving a goal. Moreover, the mechanism is often realized by letting host behavior be driven by a mathematical model called *biased random walk* (also called "guided" or "informed" random walk). For example, [9] proposed to apply biased random walk techniques in MANETs for service discovery.

Accordingly, we assume that the moves a passive CH makes are driven by a biased random walk model. In particular, when the distance between the positions of a passive CH v and the active CH u exceeds h meters (i.e., a threshold), with probability p ($p > 0.5$) and $1-p$, v will move toward and away from u, respectively. Typically, random walks are regarded as Markov processes. Therefore, we choose to use a SAN (stochastic activity network [12]) model to represent such random walks.

Due to node mobility, it is very difficult to solve or quantitatively evaluate the distribution of G. Nonetheless, the assumption of a uniformly distributed node population permits us to translate G into the fraction of the area that is covered by the active CH that is reachable by the transmission range of the passive CH, for an approximated solution (of the distribution of G). As shown by the shaded region in Figure 2, G is a decreasing function of D. Accordingly, we construct a SAN model to evaluate the distribution of D and then translate the result into the distribution of G. (Note that to estimate G at the point of takeover is pessimistic, since the passive CH will be able to adjust its position afterward to progressively improve clustering coverage.)

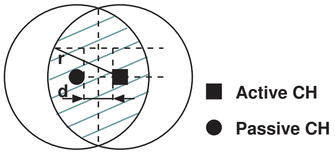


Figure 2: Gracefully Degraded Coverage upon Takeover

Figure 3 depicts the SAN model that captures, based on the X-Y coordinates, the two-dimensional biased random walk of the passive CH (before it takes over from the active CH). We exploit the marking-dependent specification capability of SANs to make the resulting model compact. In particular, we use the marking-dependent specifications to model the passive CH's threshold-dependent random walks. More succinctly, after D reaches threshold h, the case probability distributions for the instantaneous activities X_{mv} and Y_{mv} change correspondingly to represent the biased random walk explained above. In order to limit the state space, we push the computation concerning the distribution of D into the reward variable specification as follows:

$$\mathit{if}\ \left(\sqrt{(\mathit{MARK}(\mathit{Xchp}))^2 + (\mathit{MARK}(\mathit{Ychp}))^2} \leq d\right)$$
$$\mathit{reward\_rate} = 1;$$

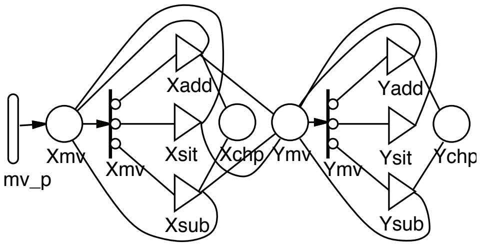


Figure 3: SAN Model

As described above, the movement of a passive CH is driven by a biased random walk model. In this model, we could choose to let the passive CH move anywhere (without constraint), or move to any location within the cluster. We assume the latter (which would yield a finite state space) in order to use the analytic solvers of *UltraSAN*. In addition, we notice that the transient solver shows that the probability distribution of D becomes stationary after 25 minutes. Coupled with the assumption that the time to CH failure, death, or abrupt departure is generally much longer, that suggests to us to use the steady-state solver for the quantitative evaluation.

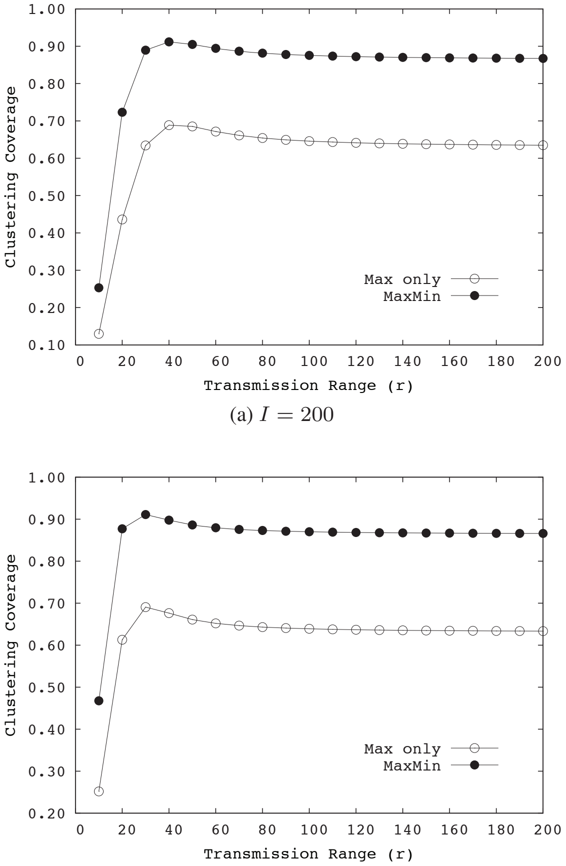
4.2 Quantitative Results

Using Mathematica and *UltraSAN*, we obtain quantitative results that demonstrate the effectiveness of SCP. We first evaluate $P^{scp}_{hm}(v)$. Figure 4 displays the results of this measure as a function of transmission range r. In the first scenario, as shown in Figure 4(a), we let 200 nodes be uniformly distributed in a 500×500 field. We see from Figure 4(a) that when the transmission range r is small (i.e., 10 meters), the values of $P^{scp}_{hm}(v)$ indicate a coverage improvement of more than 90% over the baseline protocol (i.e., $P^{bl}_{hm}(v)$). Nonetheless, neither of the protocols is able to achieve practically meaningful coverage. The reason is that with such a small r, many scattered nodes would be out of the transmission range of any other nodes, meaning that any clustering mechanisms would not be able to organize them.

However, when r increases, which means that node v is likely to have more neighbors, SCP begins to perform in a satisfactory manner. Its coverage exceeds 90% at $r = 40$, while the baseline protocol has coverage lower than 70%. Nonetheless, both protocols have a slightly decreased coverage beyond that point. The reason is that while a larger transmission range r makes a cluster accommodate more nodes, the number of competitors of v in its neighborhood also increases. That means it would be more difficult for the ID of v to be ranked as the maximum or minimum. As a result, the likelihood that v will become a CH slightly decreases. Nonetheless, the value of coverage stays fairly stable for larger values of r. By examining the constituent measures, namely $P^{scp}_{h}(v)$ and $P^{scp}_{m}(v)$, we see that the values of the former and latter slowly decrease and increase, respectively, when r continues to increase, implying that the value of $P^{scp}_{hm}(v)$ enters into an approximately equilibrium state.

In Figure 4(b), the results displayed are based on a scenario in which we assume that the 500×500 field accommodates 400 nodes. We note that for such a dense node population, both SCP and the baseline protocol reach their peak performance earlier, i.e., at $r = 30$. Beyond that, SCP's coverage remains reasonably high and stable throughout the transmission range we consider.

Figures 5(a) and 5(b) demonstrate the dual-policy-induced clusterhead redundancy and node coverage redundancy, respectively. While the redundancies are a by-product of superimposed clustering, they result in robustness advantages over the single-policy-based approaches. Figures 5(a) and 5(b) display the values of $P^{scp}_{r|ch}(v)$ and $P^{scp}_{r|cm}(v)$, respectively.



(a) $I = 200$

(b) $I = 400$

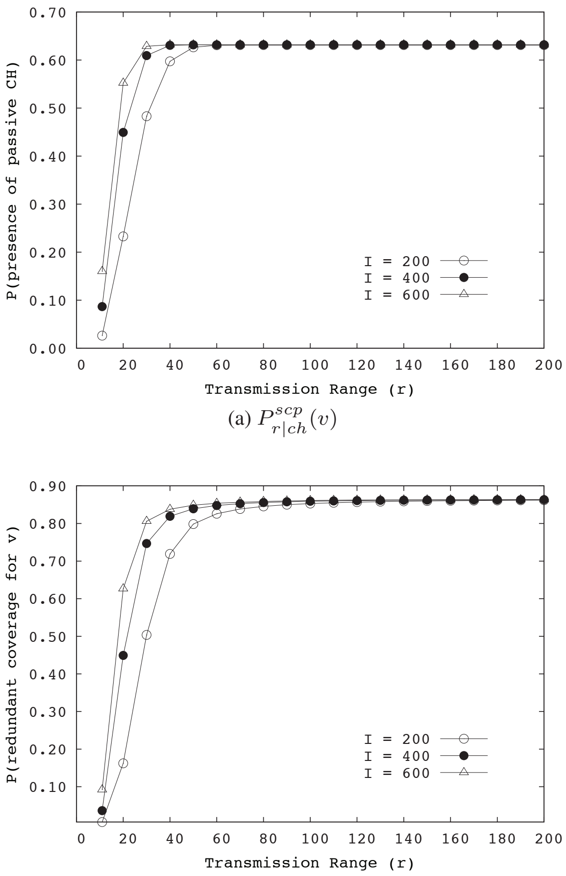
Figure 4: $P^{scp}_{hm}(v)$ vs. $P^{bl}_{hm}(v)$

We see in both figures that the likelihoods of redundancy are low when r is below 20, especially in the cases where the node population density is low. Meanwhile, we observe that both the probability of the presence of a redundant CH and the probability of redundant coverage are increasing functions of transmission range. The reason is that a larger cluster will accommodate more nodes, which in turn, will increase the likelihood of redundancies. Those are indeed desirable results, since the loss of coverage for a larger node population would have a more severe consequence, compared with the case in which a small cluster containing a few nodes is damaged. Nonetheless, the probabilities of CH redundancy and coverage redundancy fairly quickly reach an approximately equilibrium state, suggesting that a further increase of r will result in very limited benefits.

Table 2 shows the results we obtain from the SAN model illustrated in Figure 3. For the evaluation, we assume that the time between the moves of the passive CH is exponentially distributed with a rate of 0.1 (per second). When the timed activity "mv_p" is activated, the passive CH will make a move in either the positive or negative direction along the X and/or Y axis within a cluster that has a radius of 30 meters (i.e., the transmission range r). The threshold for activating/deactivating statistical preferences is kept as a variable that varies from 4 to 20 meters. Specifically, when D exceeds threshold h, we let the passive CH v have 70%

Table 2: Gracefully Degraded Coverage upon Passive CH's Takeover

$P(D \leq d)$	$P(G) \geq g$	$h = 4$	$h = 8$	$h = 12$	$h = 16$	$h = 20$
$P(D \leq 8)$	$P(G \geq 0.83)$	9.567691e-01	7.872394e-01	5.254487e-01	3.606914e-01	2.619039e-01
$P(D \leq 12)$	$P(G \geq 0.75)$	9.989058e-01	9.915983e-01	9.343174e-01	7.828988e-01	5.876513e-01
$P(D \leq 16)$	$P(G \geq 0.66)$	9.999764e-01	9.998396e-01	9.986137e-01	9.862885e-01	9.194160e-01
$P(D \leq 20)$	$P(G \geq 0.58)$	9.999994e-01	9.999967e-01	9.999738e-01	9.997524e-01	9.973851e-01



(a) $P^{scp}_{r|ch}(v)$

(b) $P^{scp}_{r|cm}(v)$

Figure 5: Inherent Redundancy

and 30% chances to move toward and away from the cluster center, respectively.

Each entry in Table 2 is thus the probability that the distance between the passive CH and cluster center will at most be d and the (translated) probability that the gracefully degraded clustering coverage G provided by the passive CH will at least be g.

The results reveal an interesting tradeoff between h and G. Furthermore, as shown in the table, even with a very relaxed threshold, i.e., $h = 20$, the probability that the passive CH will cover nearly 60% (or above) of the cluster area will still be over 99%.

5 Related Work

In recent years, many research efforts have been devoted to clustering protocols. Clustering protocols can be classified into two major categories, according to their clustering policies. In the first category, protocols apply clustering policies that explicitly address specific requirements. Examples are mobility-aware clustering, energy-efficient clustering, and combined-metrics-based clustering. The effectiveness of those protocols are dependent on the accuracy of the parameters, the validity of the assumptions, and the available amount of system profile data (for calculating the weighting factors for combined-metrics protocols). For example, with the mobility-aware clustering protocol MOBIC [13], mobile hosts with low speed relative to their neighbors become clusterhead candidates, such that hosts with similar speeds can be grouped into the same cluster to reduce the needs for cluster maintenance. Hence, the performance of MOBIC is highly dependent on the assumption of group mobility.

Protocols of the second category apply clustering policies that address requirements in an implicit fashion. For example, with passive clustering (PC) [14], cluster formation and maintenance are invoked only when mobile hosts have data to disseminate. Thus PC does not need dedicated control packets; rather, it lets control messages piggyback on the upper-layer traffic. As a result, PC allows the clustering overhead and maintenance costs to be low for MANETs with high mobility. A drawback of PC is that the readiness of a cluster-based routing structure will be a problem when a MANET exhibits bursty traffic.

The notion of dual-homing, widely used in the Internet, has been adapted to improve survivability of wireless sensor networks (WSNs). In [15], dual-homing is realized based on the composition of two paths consisting of both disjointed and jointed links, fostering partially protected multicast to ensure scalability. While SCP has a similar philosophy, it complements the notion of partial protection by enabling self-healing of cluster-based MANETs through permitting graceful performance degradation.

Several dominant pruning algorithms have been developed to reduce broadcast redundancy in ad hoc wireless networks. For example, Lou and Wu [16] developed approximation algorithms to efficiently reduce the size of the dominating set. While the purpose of the dominant pruning algorithms was to decrease the number of nodes on data forwarding paths, the objective of redundancy shadowing in SCP is to extract and combine the complementary portions of two cluster layers and to preserve the redundancies for the future needs of self-healing. Finally, it is noteworthy that our superimposed clustering framework is flexible enough to permit clustering policies to address specific requirements explicitly or implicitly.

6 Concluding Remarks

We have developed a framework of superimposed clustering and investigated an instance of SCP. This effort is meaningful for several reasons. First, since self-healing is typically realized through autonomous reconfiguration, superimposed clustering provides a low-cost approach for self-healing of large-scale systems that allows performance to be gracefully degradable.

Second, we have exploited the notion of diversity in a novel fashion. Rather than aim to reach a convergence or majority agreement for the computation results from diversified computing resources, our intent is to let diversified clustering policies create two cluster layers that will compensate for each other after redundancy shadowing. Moreover, the shadowed redundancies let a cluster-based system be prepared for self-healing.

While design diversity is often expensive in terms of development, maintenance, and performance costs, the combined use of diversified policies that are judiciously chosen will be affordable. With the MaxMin-ID-based SCP instance described in this paper, the parallel application of two clustering policies will require exactly the same neighborhood information that those policies would require when used alone. This implies that the parallel use of clustering policies and the related computation (i.e., redundancy shadowing) require no additional message exchanges between hosts in general. Further, since (relative to computation) message transmission is the major drive of performance overhead and energy consumption in MANETs, SCP can be justified with respect to its affordability as well as its efficiency.

References

[1] A. Martinoli, K. Easton, and W. Agassounon, “Modeling swarm robotic systems: A case study in collaborative distributed manipulation,” *International Journal of Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004.

[2] D. Stormont, “Autonomous rescue robot swarms for first responders,” in *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, (Orlando FL), Mar. 2005.

[3] A. Solanas and M. Garcia, “Coordinated multi-robot exploration through unsupervised clustering of unknown space,” in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Sendai, Japan), pp. 717–721, 2004.

[4] P. Xuan, “Techniques for robust planning in degradable multiagent systems,” in *Coordination of Large Scale Multiagent Systems* (P. Scerri, R. Vincent, and R. Mailler, eds.), pp. 311–340, Springer, Oct. 2005.

[5] A. Winfield and J. Nembrini, “Safety in numbers: fault tolerance in robot swarms,” *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[6] H. Zhuge, “The future interconnection environment,” *IEEE Computer*, vol. 38, pp. 27–33, Apr. 2005.

[7] J. Y. Yu and P. H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys*, vol. 7, no. 1, pp. 32–48, 2005.

[8] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 11–33, Jan. 2004.

[9] R. Beraldi, “Service discovery in MANET via biased random walks,” in *Autonomics: Proceedings of the 1st international conference on Autonomic computing and communication systems*, (Rome, Italy), pp. 1–6, 2007.

[10] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Communications*, vol. 7, pp. 28–34, Oct. 2000.

[11] G. Zhou *et al.*, “SAS: Self-adaptive spectrum management for wireless sensor networks,” in *Proceedings of 18th International Conference on Computer Communications and Networks*, (San Francisco, CA), pp. 1–6, Aug. 2009.

[12] W. H. Sanders and W. D. Obal II, “Dependability evaluation using *UltraSAN*,” in *Digest of the 23rd Annual International Symposium on Fault-Tolerant Computing*, (Toulouse, France), pp. 674–679, June 1993.

[13] P. Basu, N. Khan, and T. Little, “A mobility based metric for clustering in mobile ad hoc networks,” in *Proceedings of International Conference on Distributed Computing Systems Workshop*, (Mesa, AZ), pp. 413–418, 2002.

[14] T. J. Kwon *et al.*, “Efficient flooding with passive clustering – an overhead-free selective forward mechanism for ad hoc/sensor networks,” *Proceedings of the IEEE*, vol. 91, pp. 1210–1220, Aug. 2003.

[15] J. Wang, M. Yang, B. Yang, and S. Q. Zheng, “Dual-homing based scalable partial multicast protection,” *IEEE Trans. Computers*, vol. 55, pp. 1130–1141, Sept. 2006.

[16] W. Lou and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 1, pp. 111–122, Apr. 2002.

978-1-4244-7499-8 /10/$26.00 ©2010 IEEE