Advances in Computer Communications and Networks
From Green, Mobile, Pervasive Networking to Big Data Computing

Editors

Kewei Sha

University of Houston
Clear Lake, USA

Aaron Striegel

University of Notre Dame, USA

Min Song

Michigan Tech, USA

River Publishers

Published, sold and distributed by:
River Publishers
Alsbjergvej 10
9260 Gistrup
Denmark

River Publishers
Lange Geer 44
2611 PW Delft
The Netherlands

Tel.: +45369953197
www.riverpublishers.com

ISBN: 978-87-93379-87-9 (Hardback)
978-87-93379-88-6 (Ebook)

5

Inferring Network Topologies in MANETs: Application to Service Redeployment

S. Silvestri[1], B. Holbert[2], P. Novotny[3], T. La Porta[2], A. Wolf[3] and A. Swami[4]

[1]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, United States
[2]Department of Computer Science and Engineering, Pennsylvania State University, State College, PA 16801, United States
[3]Department of Computing, Imperial College, London SW7 2AZ, United Kingdom
[4]Army Research Laboratory, Adelphi, MD 20783, United States

Acknowledgement

This work was supported by the Defense Threat Reduction Agency under the grant HDTRA1-10-1-0085 State Estimation and Optimal Recovery in Networks with Massive Cascading Failures, and by the U.S. Army Research Laboratory and the U.K. Ministry of Defense agreement number W911NF-06-3-0001.

Abstract

The heterogeneous and dynamic nature of tactical coalition networks poses several challenges to common network management tasks, due to the lack of complete and accurate network information. We consider the problem of redeploying services in mobile tactical networks. We propose M-iTop, an algorithm for inferring the network topology when only partial information is available. M-iTop initially constructs a virtual topology that overestimates the number of network components and then repeatedly merges links in this topology

to resolve it towards the structure of the true network. We also propose an iterative service redeployment (iSR) algorithm for service redeployment in tactical networks. Extensive simulations show that M-iTop and iSR allow an efficient redeployment of services over the network despite the limitation of partial information.

Keywords: Topology inference, Partial information, Service redeployment.

5.1 Introduction

Tactical coalition networks are typically Mobile Ad hoc Networks (MANETs) composed of devices carried by soldiers or on vehicles from different nations and coalitions [1]. Some of these devices provide services, such as positioning and map services, to other devices often over multiple hops [2, 3]. As a consequence of device mobility, a static service deployment may result in poor performance and even complete lack of service. For this reason, services need to be redeployed as the topology of the network changes over time.

Previous works on service redeployment generally assume full knowledge of the service interconnections [4] and of the network topology [5–12]. This knowledge, however, may not be available in tactical coalition networks for several reasons. First, some coalitions may not share information such as the structure of their services or the connectivity of the devices. Second, wireless communications can easily experience transmission errors, especially in critical battlefield scenarios, preventing information collection. Finally, nodes are mobile and the collected information rapidly becomes stale. As a result, only partial and possibly inaccurate information on the network is available, which makes the service redeployment task challenging.

In order to address the lack of complete topology information, we propose a topology inference algorithm called M-iTop. M-iTop infers the network topology and its deployment in the physical space given the partial collected information. To the best of our knowledge, this is the first time that topology inference techniques are applied to service redeployment in MANETs.

M-iTop periodically probes the network to determine its topology. Due to the different coalition partners in the network and possible transmission errors, some nodes may not participate in the probing or may fail to provide their information. As a result, the learned topology shows missing nodes and links. Based on the collected information, M-iTop first generates a *virtual topology* by augmenting the learned topology with virtual links and nodes to restore known connectivity. It then repeatedly merges links in this topology according to consistency rules which are derived from an analysis of the collected information. The merging process resolves the virtual topology towards the structure of the true network. The inferred topology is then finally mapped in the physical space by inferring the geographical positions of the nodes.

The ability of M-iTop to accurately infer the network topology is limited by two main factors: the observability of the network (not all paths are exercised, and all links may not be covered by the probes), and the dynamics in the network while the algorithm is running. Our results show that for the typical size of a tactical network, M-iTop runs fast enough to accurately infer the current network topology even though nodes may have moved since the inference process started.

We also consider the problem of redeploying multiple service replicas in tactical networks. We formalize the problem as a multi-objective mixed integer linear programming problem, formulated using scalarization techniques. The objective is to maximize the number of nodes receiving service as well as the overall Quality of Service (QoS). In order to efficiently solve this problem, we propose the Iterative Service Redeployment (iSR) algorithm. Our results show that the short execution time of M-iTop and the accuracy of the inferred topology enable efficient service redeployment by iSR over the network, despite the limitations of partial information.

In summary, our main contributions are the following:

- We propose M-iTop to infer the network topology of mobile coalition networks.
- We formulate a mixed integer linear programming problem for multiple service redeployment.
- We propose the iSR to efficiently solve the redeployment problem.
- We show that M-iTop has a short execution time for typical coalition networks.
- We show that M-iTop and iSR provide an accurate estimate of the actual topology which enables efficient service redeployment.

5.2 Related Work

The problem of service redeployment has been recently considered in both wired networks with static topology [5–7, 13] and wireless networks with dynamic topology [8–11, 14, 15]. These methods assume full knowledge of the network topology and would not be able to operate in a tactical coalition network characterized by partial topology information.

The problem of inferring network topology in the presence of non-cooperating nodes which leads to partial information has been considered in several recent works [16–19]. The problem was first introduced in [16] and extended in [17] to scenarios with less information. Reference [19] identifies patterns in the traces that indicate certain structures in real network. While the approach in [19] has lower runtime complexity than [17], it is not as accurate. The approach proposed in [18] supplements trace information with additional data from the *record route*. This results in a more accurate inferred topology, but requires information which may not be available. Finally, in [20] we propose an algorithm called iTop to infer static network topologies. This algorithm provides higher inference accuracy than previous approaches.

The above-cited works only focus on static Internet-like topologies. As a consequence, they are not directly applicable to the context of MANETs. In this chapter, for the first time, we apply topology inference techniques to service redeployment in MANETs. We introduce the algorithm M-iTop, which extends our previous algorithm iTop [20] by specifically taking into account features of dynamic networks such as MANETs (hence the name Mobile-iTop). In particular, M-iTop addresses the node mobility and infers the network deployment in the physical space.

5.3 Network Model

We consider a network of nodes which form a MANET. Nodes are mobile and able to communicate with each other. We consider a binary disk communication model, i.e., two nodes are able to communicate if their physical distance is less than the fixed *transmission range* R_{tx}. Our approach can be easily extended to anisotropic and heterogeneous communication environments. We assume the MANET is composed of two coalition partners. Hence, nodes belong to two teams, to which we refer as *owned network* C_{own} and *allied network* C_{ally}. We assume the availability of an underlying routing algorithm in which all the nodes participate for communications[1]. We further assume that pairs of nodes in the owned network have a mechanism to measure the hop distance of the path between them, for example by analyzing the routing tables.

[1]In the following we assume that the routing algorithm is based on shortest path. Our approach can be easily extended to different routing strategies.

We refer to the real topology as the *Ground Truth* topology. We represent the network at time t by an undirected graph $G^t_{GT} = (V_{GT}, E^t_{GT})$. V_{GT} is the set of nodes. There is an edge in E^t_{GT} between two nodes if they can directly communicate at time t, i.e., they are within physical distance R_{tx}.

Periodically, nodes in the owned network probe the network in order to acquire information about the current topology. In particular, each owned node sends a probe to every other node in C_{own}. We refer to the topology generated by the union of the probed paths as the *observable topology* $G^t_{obs} = (V_{obs}, E^t_{obs})$. This topology is a subgraph of the G^t_{GT} topology as, in general, routing paths between owned nodes may only partially cover the network. Owned nodes reply to the probing with their information such as ID and geographic position. On the contrary, allied nodes do not provide any information even though they forward probes. We refer to the partial (or total) path information acquired by an owned node as a *trace*.

Traces are centrally collected and combined at the Network Operating Center (NOC). Since allied nodes do not provide their information when probed, the acquired information only partially represents the observable topology G^t_{obs}. The NOC executes our topology inference algorithm in order to estimate the observable topology. A service redeployment algorithm is then executed on the inferred topology.

5.4 M-iTop Approach

In this section we describe the operation of M-iTop, our topology inference approach with partial information. In the following we consider the operations performed by M-iTop at a given time t, but for ease of presentation the superscript t is omitted.

M-iTop operates in four phases. In the first phase it analyzes the traces and constructs the *virtual topology* $G_{VT} = (V_{VT}, E_{VT})$, which is a vastly overestimated topology compared to the observable topology. During the construction of the virtual topology, M-iTop classifies nodes in the virtual topology on the basis of their observed behavior with respect to the probes. In the second phase, M-iTop determines the *merge options* for each link in the virtual topology. These options indicate pairs of links in the virtual topology that can be merged while preserving consistency with respect to the observed characteristics of the ground truth topology and the node classifications assigned in the previous phase. In the third phase, M-iTop infers the merged topology $G_{MT} = (V_{MT}, E_{MT})$ by iteratively merging pairs of links based on their merge options and removing any merge options that are made invalid as a result.

The first three phases of M-iTop estimate the topological structure, i.e., the network graph, of the observable topology. As a result, these phases make use of the *hop distance* between nodes. The fourth phase infers the geographical locations of nodes of the allied network by exploiting the known locations of owned nodes and their connectivity in G_{MT}. For this reason, this phase makes use the *physical distances* between nodes.

The goal of the algorithm is to infer a G_{MT} that is as close as possible to G_{obs} given the partial information collected by the owned nodes.

5.4.1 Virtual Topology Construction

The NOC collects the information gathered by all nodes in the owned network and constructs the virtual topology as follows. Consider two owned nodes m_1 and m_2 connected by the path $m_1, v_1, v_2, \ldots, v_{n-1}, m_2$. Without loss of generality, assume that node m_1 initiates a probe to node m_2, to estimate their mutual hop distance $d(m_1, m_2)$ and collect the path information. The gathered information and hop distances are sent to the NOC by m_1, which analyzes them to infer the virtual topology and partitions the nodes into two classes. These classes are introduced in order to guide the merging process of M-iTop and reflect the node behavior as observed in the traces.

We define two classes of nodes. R is the class of *responding* nodes and A is the class of *anonymous* nodes. Responding nodes belong to the owned network while anonymous nodes belong to the allied network. When processing the collected traces, the NOC marks each node as belonging to one of these classes as follows.

Consider the case in which a node m_1 probes the path to m_2 and successfully receives a response from m_2. Since a reply was received, the NOC can conclude that these two nodes are connected. The path information is of the form $(m_1, x_1, \ldots, x_{n-1}, m_2)$ where $d(m_1, m_2) = n$ and each x_i either identifies a node v_i that is responding, or is a $*$ to denote no response. All nodes corresponding to a $*$ are anonymous. The NOC adds a node in the virtual topology for each responding node observed in the trace and connects them accordingly. It marks these nodes as responding and combines multiple instances of the same responding nodes reported by other owned nodes to avoid duplication of observed components. The anonymous nodes and the links connecting them are also added to the virtual topology. A virtual node is added for each $*$ in the trace and it is marked as anonymous.

The constructed G_{VT} topology overestimates the network because it may contain multiple anonymous nodes which are the same node in the G_{GT} topology. Since allied nodes do not provide their information, there is no simple way to determine which ones are the same. Therefore they are assumed to be distinct nodes until merged in the third phase of M-iTop.

Figure 5.1 shows an example scenario. The G_{GT} topology is shown in Figure 5.1(a). Dark nodes are the owned nodes, which take part and respond to the probing, while white nodes belong to the allied network and do not take part in the probing nor provide their information. In this example we assume shortest path routing for probing. As a result, nodes F, G, and H do

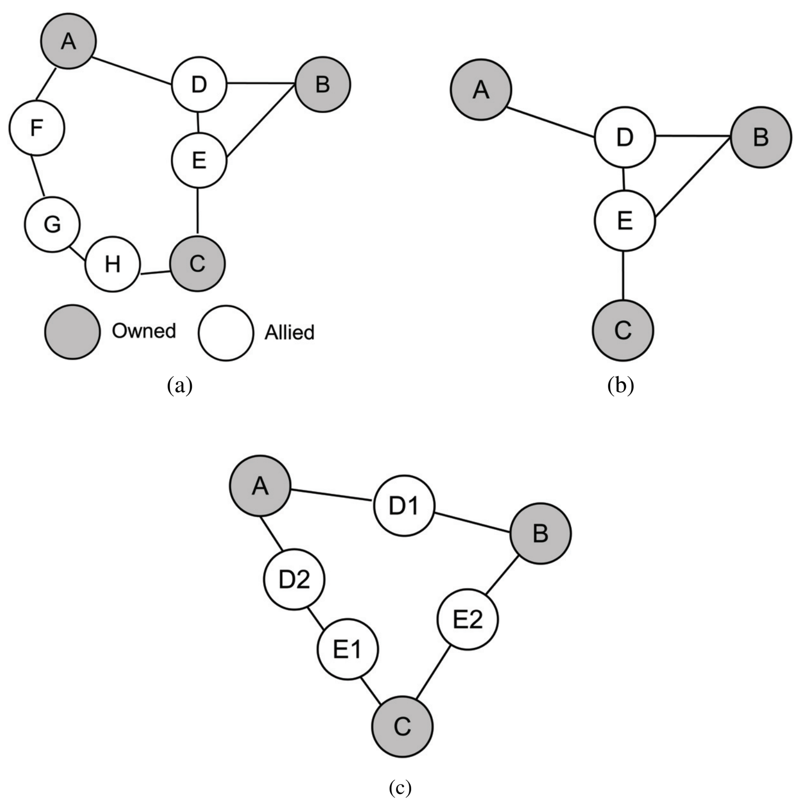


Figure 5.1 Example: Ground truth topology G_{GT} (a), observable topology G_{obs} (b) and virtual topology G_{VT} (c).

not appear in the observable topology G_{obs} as they are not covered by any shortest path between owned nodes as shown in Figure 5.1(b). Figure 5.1(c) shows the corresponding virtual topology. Multiple occurrences of the same anonymous node in the traces are represented as different nodes in the virtual topology, resulting in an overestimation of the observable topology G_{obs}. These occurrences are then merged in the subsequent phases of M-iTop.

5.4.2 Merge Options

In order to infer G_{MT} from G_{VT}, M-iTop identifies the valid merge options for each link e_i in E_{MT}, i.e., the set of links with which e_i can be merged. Initially, $E_{MT} = E_{VT}$. We introduce three conditions which have to be satisfied for a merge option to be valid. These conditions check the consistency of a merge option with the information gathered from the traces and with the node classification provided in the previous phase.

The set M_i denotes the set of links which are valid merge options for link e_i. A link e_i has a valid merge option with e_j if all the following conditions are verified. The sets of merge options are then used during the merging phase to determine which merges occur and in what order.

Trace Preservation: Since paths do not contain loops, a link will never appear twice in the same path. A merge option between two links satisfies trace preservation if these links do not appear together in any path.

Hop Distance Preservation: The hop distance between any two owned nodes in G_{VT} is consistent with the G_{obs} topology. A merge option is valid if the hop distance between owned nodes in G_{MT} is the same as that in G_{VT}. For example, in Figure 5.1(c), A-D1 can be merged with A-D2, C-E1 with C-E2, but not B-D1 with B-E2.

Link Endpoint Compatibility: The node classification gives us additional information on valid merging options. In particular, a node can be either responding or anonymous. A merge option between two links is valid if there is a way to combine their endpoints without violating their classification. In particular, a merge option between two links e_i and e_j is valid in two cases: (1) the endpoints of e_i and e_j are all classified as anonymous, or (2) for both e_i and e_j one endpoint is classified as anonymous and the other as responding. This second option is valid only if the responding node is the same for both links.

5.4.3 Merging Links

The next phase merges the links in the virtual topology to derive the M-iTop topology $G_{M\mathrm{T}}$. Initially, $G_{MT} = G_{VT}$. The merging phase reduces G_{MT} by iteratively merging pairs of links based upon the existing merge options. Each merge combines two links in E_{MT}, combining their endpoints and reducing the number of components in the network accordingly. When no merge options remain, the merging phase is complete and the M-iTop topology is finalized.

Algorithm 5.1 shows the pseudo-code of the merging phase. In each step, M-iTop chooses two links and attempts to merge them. Several alternatives are possible to determine the order in which links are merged, which influences the resulting final topology. Since links with few merging options have fewer merging possibilities, they are more likely to be the same link in the ground truth topology. On the basis of this observation, we first select the link e_i with the fewest merging options and then link e_j which has the fewest merging options out of the links with which e_i can be merged (Algorithm 5.1, lines 2–3). We experimented with several alternative heuristics and the one described above provides the best results.

Link merging

Two links e_i and e_j are merged by the function $Merge(e_i, e_j)$ (Algorithm 5.1, line 4). For ease of exposition we describe the merging of e_j into e_i, as the same result would be obtained by the opposite merging. All paths containing e_j are modified to contain e_i in its place and the set M_i is changed so that $\mathrm{M}_i = \mathrm{M}_i \cap \mathrm{M}_j$. Any links which could have been merged with both e_i and e_j retain their merge option with e_i and the merge option for e_j is removed. Any links which had a merge option with either e_i or e_j but not both have that option removed. This ensures that e_i can only be further

Algorithm 5.1 M-iTop merging phase

Input: Initial Virtual Topology $G_{VT} = (V_{VT}, E_{VT})$, Merge Options M
Output: Merged M-iTop Topology $G_{MT} = (V_{MT}, E_{MT})$
1 $G_{MT} = G_{VT}$;
2 **while** $\exists e_i \in E_{MT} \wedge \mathsf{M}_i \neq \emptyset$ **do**
3 $\quad e_i = argmin_{e_i \in E_{MT}} |\mathsf{M}_i|$;
4 $\quad e_j = argmin_{e_j \in \mathsf{M}_i} |\mathsf{M}_j|$;
$\quad$ // Link merging
5 $\quad$ Merge(e_i, e_j);
6 **return** $G_{MT} = (V_{MT}, E_{MT})$

merged with links that before were valid merges for both e_i and e_j. The link e_j is then removed from E_{MT}.

5.4.4 Inference of Nodes Physical Locations

The previous phases of M-iTop enable us to infer the structure of the network topology. In this phase we infer the geographical locations of nodes in the inferred topology. Owned nodes provide their location information when the network is probed. On the contrary, allied nodes do not, and thus their positions need to be inferred. In the following we describe the procedure M-iTop adopts for this purpose.

Let s be a node of the allied network and let $N_{loc}(s)$ be the set of its one-hop neighbors whose positions are known; this set may be empty. For each allied node s, we calculate the area $\mathcal{R}(s)$ defined as follows:

$$\mathcal{R}(s) = \Big(\bigcap_{p \in N_{loc}(s)} R_{tx}(p) \Big) \setminus \Big(\bigcup_{p \in (C_{own} \setminus N_{loc}(s))} R_{tx}(p) \Big) \tag{5.1}$$

where $R_{tx}(p)$ is the transmission area of node p, i.e., the circle centered at p and with radius R_{tx}. $\mathcal{R}(s)$ defines the area in which node s can lie given the physical distance constraints imposed by the inferred connectivity. In order to determine the location of allied nodes, we rely on graph drawing algorithms modified to our needs. In particular, we fix the position of owned nodes and we constrain the position of each allied node s to $\mathcal{R}(s)$. We further modify the drawing algorithm by imposing the physical distance between two neighboring nodes in G_{MT} to be at most R_{tx}.

Given the location constraints, we execute a graph drawing algorithm until it converges to a stable configuration. We tried several graph drawing algorithms and adopted Force Atlas [21] as it provides good performance in the considered scenarios.

Note that, if owned nodes can estimate the physical distance to neighboring nodes, for example using the received signal strength [22], the above process can be further improved by including triangulation.

5.5 Iterative Service Redeployment (iSP) Algorithm

In this section we present the formalization of the multiple service replicas redeployment problem as well as of the Iterative Service Redeployment (iSR) algorithm to efficiently solve such problem.

5.5.1 Formalization of the Multiple Service Replicas Deployment Problem

As a consequence of node movements, the network is partitioned into m connected components $G_1, \ldots, G_m$. We represent each component G_i as a graph, i.e., $G_i = (V_i, E_i)$. The primary goal of the problem is to maximize the number of users that can access a service. To this purpose, up to R service replicas can be deployed on the nodes of the owned network. The secondary goal is to optimize the overall QoS provided by such services. We assume that, for each node v in the owned network, we know a QoS metric (e.g., delay, packet loss) $q(v, u)$ for any other node u in the entire network. Note that the QoS metric may be only partially available when service redeployment is applied to an inferred topology. Nevertheless, it can be estimated using the nodes' geographical locations inferred in the fourth phase of M-iTop [23].

We formulate the problem as a multi-objective optimization problem, since we aim to optimize both the number of served users and the overall QoS. The problem is expressed as a mixed integer linear programming problem using scalarization [24]. Let $V_i^{own} = V_i \cap C_{own}$ be the set of owned nodes in the G_i component. We formally define the problem as follows.

$$\max_{x_u} \sum_{i=1}^{m} z_i |G_i| + \beta \sum_{i=1}^{m} \frac{1}{|V_i|} \sum_{u \in V_i^{own}} \sum_{v \in V_i} y_{u,v} q(u, v) \tag{5.2}$$

$$\text{subject to } z_i \leq \sum_{u \in V_i^{own}} x_u \; i = 1, \ldots, m \tag{5.3}$$

$$\sum_{u \in V_i^{own}} y_{u,v} \leq 1 \; \forall v \in V_i \tag{5.4}$$

$$y_{u,v} \leq x_u \; \forall u, v \tag{5.5}$$

$$\sum_{i=1}^{m} \sum_{u \in V_i^{own}} x_u \leq R \tag{5.6}$$

$$x_u, z_i, y_{u,v} \in \{0, 1\} \tag{5.7}$$

where x_u is equal to 1 if a service is deployed on node u, and to 0 otherwise; $y_{u,v}$ is equal to 1 if node v receives service from node u, to 0 otherwise; finally, z_i is equal to 1 if the connected component G_i contains at least one service, and to 0 otherwise. Table 5.1 summarizes the meaning of variables and constants used in the optimization problem.

Table 5.1 Summary of variables and constants used in the optimization problem (5.2)

Symbol	Meaning
$G_i = (V_i, E_i)$	i-th connected component
x_u	Equals 1 if a service is deployed on node u, 0 otherwise
$y_{u,v}$	Equals 1 if node v receives service from node u, 0 otherwise
z_i	Equals 1 if the component G_i contains at least one service, 0 otherwise
R	Number of services replicas to be deployed
$q(u, v)$	QoS metric between node u and node v

The goal is to jointly maximize the number of nodes served by at least one service as well as the average QoS in each component. We introduce the parameter $\beta \in \mathbb{R}^+$ to prioritize these objectives. In the remainder of the chapter, we assume that β is sufficiently small, such that having more nodes receiving service is always prioritized with respect to improving the QoS.

The constraint in Equation (5.3) ensures that a component is counted in the objective function only if at least one service is deployed in that component. Equation (5.4) constrains a node to receive service from at most one node. Equation (5.5) enables a node v to receive service from u, only if a service is deployed on u. Finally, Equation (5.6) ensures that at most R replicas are deployed.

The optimization problem may be computationally too intense to be solved exactly in a timely manner. For this reason, we propose the algorithm iSR that efficiently finds an approximate solution.

5.5.2 The iSR Algorithm

The Iterative Service Redeployment (iSR) algorithm is executed by the NOC and it is designed to efficiently find an approximate solution to the problem described in the previous section. The algorithm outputs, for each component G_i, the set of nodes $S_i \subseteq V_i^{own}$ on which service replicas are deployed. The pseudo code of the algorithm is shown in the algorithm iSR. In the pseudo-code, we explicitly omit the components with no owned nodes.

iSR iteratively deploys one service at a time. Initially, it considers the largest component that does not contain a service, and selects the node that optimizes the average QoS to all other nodes in that component (lines 4–7). Subsequently, when all components have at least one service (line 8), it identifies the node that would provide the best improvement in each component. To this purpose it calculates for each component G_i, the node u_i^* that provides the best average QoS improvement Δ_i^* in G_i (lines 9–11). It then selects the node with the highest Δ_i^* (lines 12–13).

Algorithm 5.2 Iterative service redeployment algorithm (iSR)

Input: Topology $G = (V, E)$, distance metric function $q\colon V \times V \rightarrow \mathbb{R}^{+}$, number of services to be deployed R
Output: Set of selected nodes $S_1, \ldots, S_m$ in each component for service redeployment

$r = 0$;
$S_i = \emptyset \;\; \forall i = 1, \ldots, m$;
while $r \leq R$ **do**
  if $\exists\, G_i$ *s.t.* $S_i = \emptyset$ **then**
    // Select largest connected component
    $G^* = (V^*, E^*) = \mathrm{argmax}_{G_i\ s.t.\ S_i=\emptyset} |G_i|$;
    // Select largest best node
    $u^* = \mathrm{argmax}_{u \in V^*} \frac{1}{|V^*|} \sum_{v \in V^*} q(u, v)$;
    $S_i = S_i \cup \{u^*\}$;
  else
    for *i = 1 to m* **do**
      $u_i^* = \mathrm{argmax}_{u \in V_i \setminus S_i} \frac{1}{|V_i|} \sum_{v \in V_i} \max_{\hat{u} \in S_i \cup \{u\}} q(\hat{u}, v)$;
      $\Delta_i = \frac{1}{|V_i|} \Big(\sum_{v \in V_i} \max_{\hat{u} \in S_i \cup \{u\}} q(\hat{u}, v) - \sum_{v \in V_i} \max_{\hat{u} \in S_i} q(\hat{u}, v) \Big)$;
    Let G^* be the component with maximum Δ_i and u^* the selected node;
    $S^* = S^* \cup \{u^*\}$;
  r++;
return $S_1, \ldots, S_m$;

iSR has complexity $O(m \log m + R \times m \times \max_i |V_i|^2)$. $m \log m$ is necessary to sort the m components by size, and for each of the R service replicas deployed, it is necessary to find the node with best improvement, which is proportional to the square of the number of nodes in the component.

5.6 Results

In this section we evaluate the performance of iSR when applied to the topology inferred by M-iTop. To this aim we developed a simulator based on the Wireless module of the Opnet environment [25].

When iSR is applied to an inferred topology, several sources of errors may occur in selecting nodes for service deployment. First, the topology inference algorithm may have a non-negligible execution time. As a result, nodes in the current topology might have moved, possibly making the inferred topology inaccurate by the time it is produced. Second, the nodes selected in the inferred topology may not be the actual best nodes in the G_{GT} topology, due to inaccuracies of the considered QoS metric when calculated on the inferred topology. Finally, the actual connected component sizes in the G_{GT} topology may not correspond to the sizes in the inferred topology. This is due to the inherent partial observability of the G_{GT} topology and to the possible inaccuracies in the inferencing process.

We use the following simulation setting. We randomly deploy nodes on a bounded square area of size $500\ \mathrm{m} \times 500\ \mathrm{m}$. Nodes move at a maximum speed of $0.5\ \mathrm{m/s}$ according to a local random way point model. In particular, each node selects a random position in a circle of radius $100\ \mathrm{m}$ around the initial location in which it was deployed. Nodes have a waiting time between consecutive movements chosen randomly in the interval $[0, 300]$ s. This movement strategy reflects mission-oriented local movements typical of military MANETs. We also assume shortest path routing and hop distance as the QoS metric for the service redeployment algorithm. Communication delays are integrated using the realistic wireless communications module of the Opnet simulator [25]. We use different numbers of nodes, communication ranges and percentage of nodes in the owned coalitions for different scenarios.

We first study the execution time of M-iTop by increasing the network size. For this experiments we use a transmission range of 200 m, which results in a highly connected network, and we assumed that 50% of the nodes are in the owned coalition. We averaged the results over 100 initial random deployments. Results are shown in Figure 5.2. Although the execution time of M-iTop rapidly increases with the network size, it is reasonably low for networks of 60 nodes, and in particular on the order of few seconds for networks with 40 nodes or less. The short execution time of M-iTop for small networks makes it suitable for inferring the topologies of tactical networks, whose size is generally on the order of a few tens of nodes.

In order to study the performance of iSR on the topology inferred by M-iTop, we performed four sets of experiments. In the first set we focus on the effects of the execution times of M-iTop, and of the inaccuracies of the inferred topologies, on the performance of iSR. In the second set, we study how the performance is affected by the percentage of nodes in the owned coalition. In the third set, we perform a sensitivity analysis to the setting of the transmission range and the nodes' speed. In the first three sets we consider the deployment of a single service replica, i.e., $R = 1$, and in the fourth set we consider the deployment of multiple replicas.

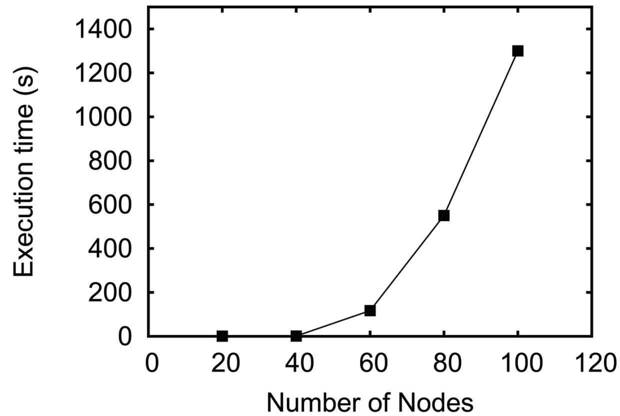
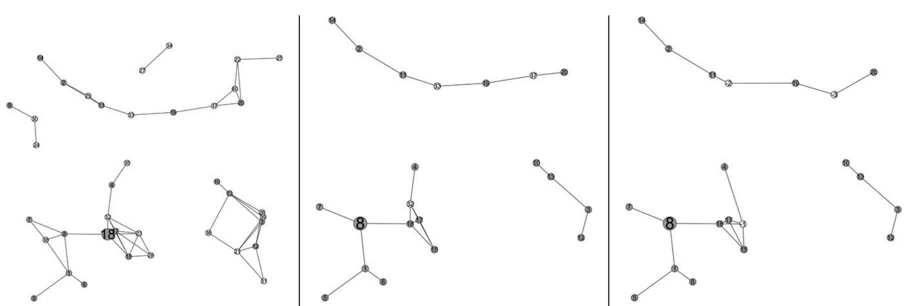


Figure 5.2 Execution time of M-iTop with increasing network size.

Before showing the results, we show an example of iSR applied to the G_{GT} topology, the observable topology G_{obs} and the topology inferred by M-iTop, to deploy one service replica. We consider a network with 40 nodes and transmission range $100\ \mathrm{m}$. The G_{GT} topology is shown in Figure 5.3(a). Dark nodes belong to the owned network while white nodes to the allied network. The observable topology G_{obs} obtained by probing the network between owned nodes is shown in Figure 5.3(b). As expected this topology is a subset of the G_{GT} topology. Figure 5.3(c) shows the topology inferred by M-iTop. It closely matches G_{obs}, showing the effectiveness of the M-iTop algorithm.

In Figure 5.3(a–c), the nodes highlighted with a larger size are selected by the service redeployment algorithm. The node that belongs to the largest



(a) (b) (c)

Figure 5.3 Example of ground truth topology G_{GT} (a), observable topology G_{obs} (b) and topology inferred by M-iTop G_{MT} (c).

component and minimizes the average hop distance to the other nodes is 18, as selected in the G_{GT} topology. In the G_{obs} topology, the largest component can still be correctly identified, but the partial information given by probing results in the selection of node 8. The topology inferred by M-iTop closely matches G_{obs}, which enables it to correctly determine the largest component and to select node 8. In the following experiments we show that the nodes selected from the inferred topologies provide good service when compared to the best choice that would be obtained with complete information.

5.6.1 First Set of Experiments

In these experiments we assume an equal number of allied and owned nodes. We periodically execute iSR every 500 s. This period provides good performance considering the mobility model adopted in this chapter. We apply iSR to deploy a single service replica, i.e., $R = 1$, to G_{GT}, G_{obs} and to the topology inferred by M-iTop G_{MT}. The case of multiple service replicas is studies in Section 5.6.4. For G_{GT} and G_{obs} the service redeployment takes effect instantly. On the contrary, for M-iTop the service redeployment is delayed by the time required for probing the network, collecting the information at the NOC and executing M-iTop. In particular, as soon as a new slot for service redeployment starts, the network is probed and M-iTop is executed. The node selected in the previous slot still provides service until M-iTop terminates and a new node can be selected. For comparison, we also consider the ideal unrealistic case in which the service is placed on the optimal node at each instant in time. This case is referred to OPT in the figures.

In the experiments, given a node u selected by iSR, we consider as a performance metric the actual average hop distance of u to the nodes belonging to its connected component in the G_{GT} topology.

5.6.1.1 Single connected component

We initially use a large transmission range of $200\ m$, which always results in a single connected component. This experiment enables us to focus on the effects of the execution times of M-iTop and of the inaccuracies of the inferred topology.

Figure 5.4(a) shows the average hop distance over time of the node selected by iSR, for a network of 40 nodes. Vertical lines indicate the times at which iSR is executed. In the case of M-iTop, these are the moments at which the network is probed and inferred. Once the inference is completed the service redeployment algorithm is then executed.

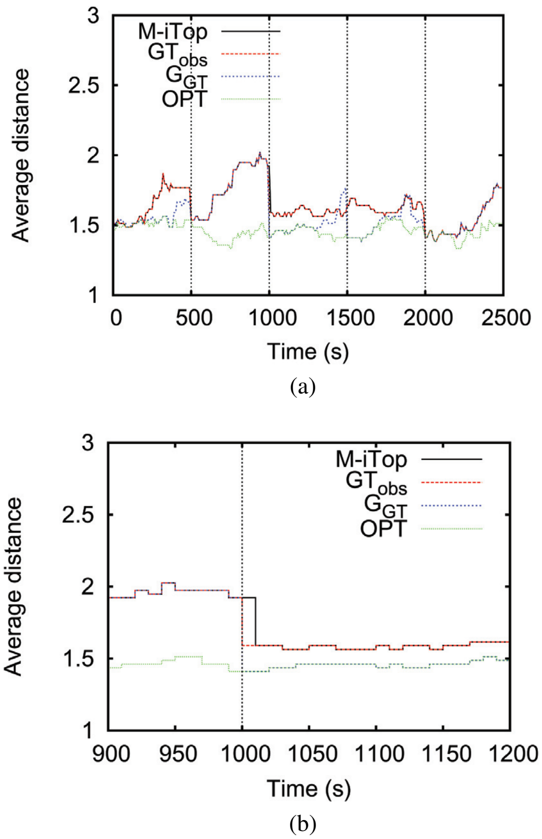


Figure 5.4 Network with 40 nodes, single connected component. Average hop distance over time (a), zoom on the service redeployment at 1000 s (b).

The distances under G_{GT}, G_{obs} and M-iTop generally get worse with respect to OPT as the amount of time since the last execution of the service redeployment algorithm increases. This is due to node mobility, which changes the structure of the network and causes the service redeployment decision to become stale. The vertical performance gap of M-iTop with respect to G_{GT} in Figure 5.4(a) shows the performance worsening due to the partial information available and to the possible inaccuracies of the inference process. The horizontal gap shows instead the delay which is incurred by the execution time of M-iTop. The results highlight that M-iTop has an average distance close and often equal to that of G_{GT} and G_{obs}. The inferred topology enables proper selection of a node for service redeployment even in the presence of partial information.

In order to highlight the effects of the execution times of M-iTop, we zoom on the results of the previous experiment around $1000\ \mathrm{s}$ in Figure 5.4(b). As the service redeployment algorithm is executed at time $t = 1000$, G_{GT} and G_{obs} instantly switch to the new node since no execution delay is considered for these approaches. M-iTop terminates in less than $10\ s$ for a network of 40 nodes. As a result, after a short time a new node is selected on the basis of the inferred topology. This is the same node that is selected by G_{obs}, highlighting that the inferred topology closely matches the observable topology. The G_{GT} topology enables the selection of a slightly better node, but requires full knowledge of the network topology that is not available in a coalition network.

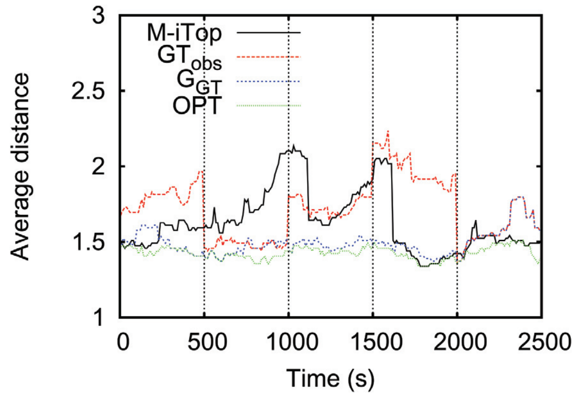
Figures 5.5(a) and (b) show the average hop distance over time for a 60-node network. The execution time of M-iTop is around $120\ \mathrm{s}$ for sixty nodes. Also in this case, M-iTop achieves the performance close to and often coinciding with G_{GT} and G_{obs}, with a slightly longer delay due to the algorithm execution.

5.6.1.2 Multiple connected components

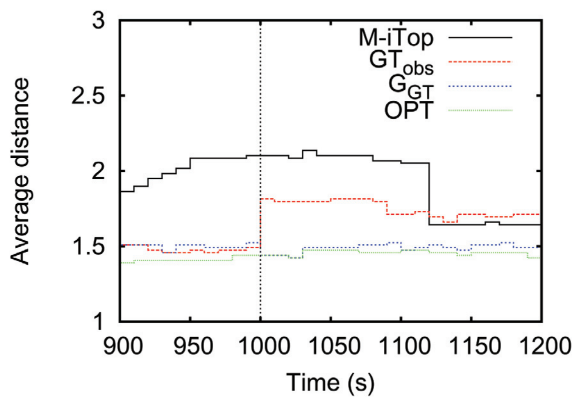
We now consider a reduced setting of the transmission range to $100\ \mathrm{m}$, which generates a network partitioned into several connected components over time. These experiments enable us to also show the possible inaccuracies incurred due to an incorrect selection of the largest connected component. We consider a network of 40 nodes.

Figures 5.6(a) and (b) show the average hop distance between the selected node and the number of nodes receiving service. In this case, M-iTop is able to provide the performance close to that of G_{GT} and G_{obs} most of the time. In the time interval between $t = 1000\ \mathrm{s}$ and $t = 1500\ \mathrm{s}$, M-iTop has a lower average distance than do G_{GT} and G_{obs}. In this case, iSR applied to G_{MT} fails to select the actual largest connected component in G_{GT}. As a result, the average distance is lower but at the expense of fewer nodes receiving service. It should be noted that the network partition allows only a subset of the nodes to be connected to the service node. Figure 5.6(b) highlights that under M-iTop the selected node belongs to the largest component in most cases.

Overall, the topologies inferred by M-iTop enable the service redeployment algorithm to select a node which provides performance close to that in the case of complete knowledge, even in the presence of partial information, network partitions and node mobility.



(a)

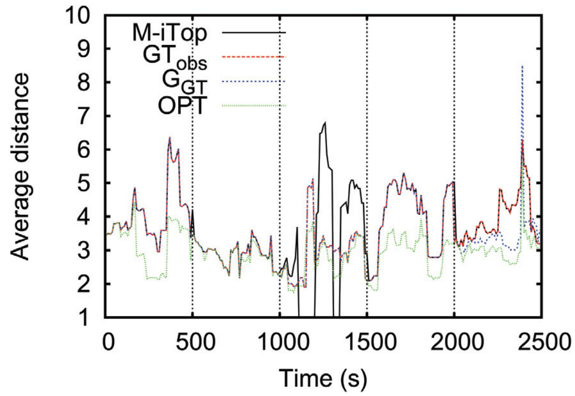


(b)

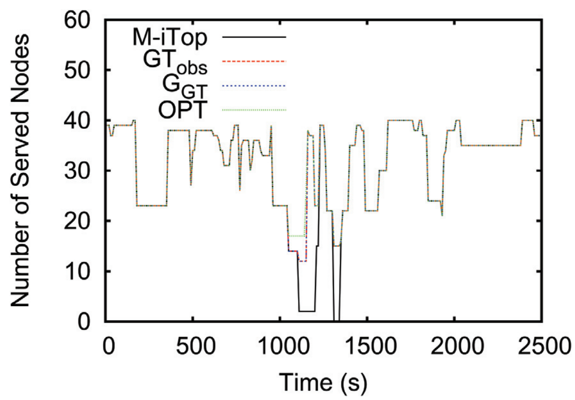
Figure 5.5 Network with 60 nodes, single connected component. Average hop distance over time (a), zoom on the service redeployment at 1000 s (b).

5.6.2 Second Set of Experiments

In this set of experiments we vary the percentage of owned nodes and study the performance of iSR applied to the inferred topologies. Also in this case, iSR deploys a single service replica ($R = 1$). Intuitively, a smaller number of owned nodes reduces the extent of the observable topology G_{obs}, and thus it also provides less information for the topology inference algorithm. On the contrary, when a large fraction of nodes belongs to the owned network, more information is available and thus the inferred topology more accurately represents the ground truth topology. In these experiments we set $R_{tx} = 200\ m$ which ensures the presence of a single connected component.



(a)

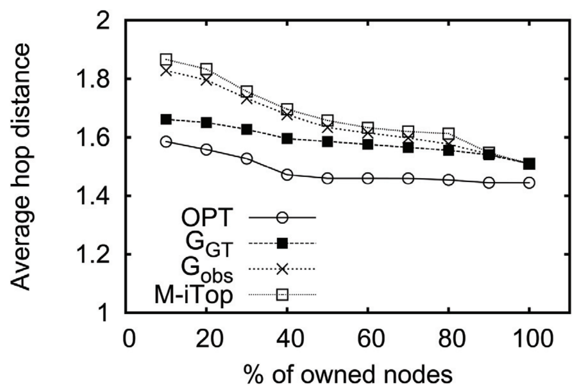


(b)

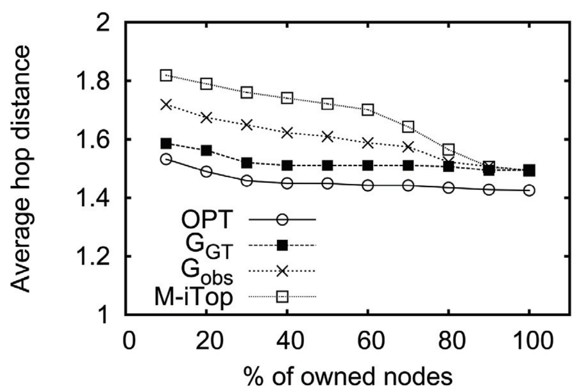
Figure 5.6 Network with 40 nodes, multiple connected components. Average hop distance over time (a), number of served nodes over time (b).

We study the average hop distance over time of the node selected by the service redeployment algorithm by increasing the percentage of nodes belonging to the owned network C_{own}. The distance is averaged over a period of 3000 s. and iSR is executed every 500 s. We compared the performance of iSR when applied on the full topology G_{GT}, on the observable topology G_{obs} and on the topology inferred by M-iTop. Also in these experiments we consider the ideal unrealistic case (OPT) in which the service is placed on the optimal node at each instant in time.

Figure 5.7(a) shows the results for a network of 40 nodes. When the full knowledge of G_{GT} is available, the service deployment algorithm achieves the best performance. The gap with respect to OPT is due to the time gap between subsequent executions of the deployment algorithm, during which nodes move and thus may result in suboptimal performance. When the percentage of owned nodes is small, G_{obs} only partially represents G_{GT}. As a result, the performance of the service deployment algorithm is inevitably poor. On the contrary, as the percentage of owned node increases, the performance improves and eventually converges to the case in which full knowledge is available. For a network of 40 nodes, M-iTop is able to accurately and quickly infer a



(a)



(b)

Figure 5.7 Average hop distance vs. the percentage of owned nodes. Network with 40 nodes (a), and network with 60 nodes (b).

topology close to the observable topology. As a result, the service deployment algorithm performs similar to the case in which the observable topology is perfectly known.

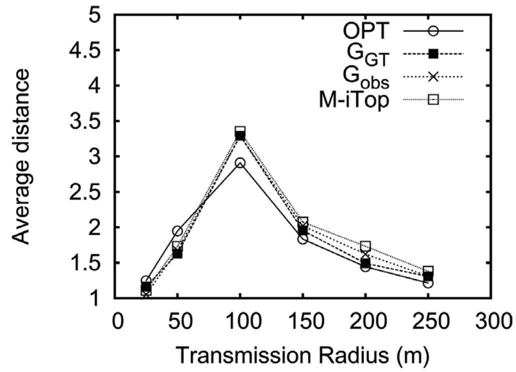
Figure 5.7(b) shows the results for a network of 60 nodes. The performance of G_{GT} and G_{obs} are similar to that in the case of 40 nodes. M-iTop has slightly worse performance than in the case of a 40-node network, due to the longer execution times (approximation 120 s. for a 60-node network). Such execution times delay the service redeployment, thus increasing the average distance (see also Figure 5.5). Nevertheless, M-iTop is always within 10% from the case in which the observable topology is perfectly known.

5.6.3 Third Set of Experiments

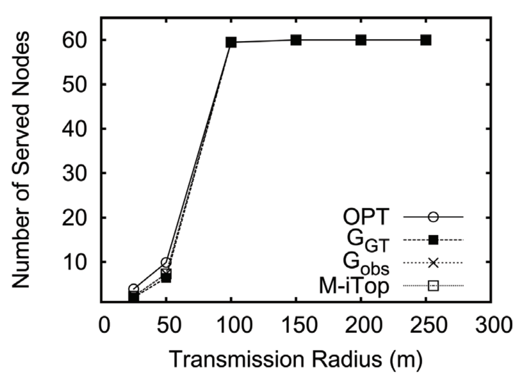
In this set of experiments we perform a sensitivity analysis of the performance of M-iTop to the setting of the communication radius R_{tx}. Intuitively, a smaller radius results in a higher number of small components, while a longer radius reduces the number of components and increases their size. In the experiments we consider a network of 60 nodes and we apply iSR every 500 s to deploy a single service replica. We increase the transmission radius from 25 to 250 m.

Figure 5.8(a) shows the average hop distance from the selected service nodes, while Figure 5.8(b) shows the number of nodes that receive service. When the transmission range is small ($\leq$50 m) there are several small components, and additionally such components are constantly changing over time due to node mobility. OPT continuously redeploys the service, thus being able to always correctly select the largest component. Conversely, with G_{GT}, G_{obs} and M-iTop, iSR is executed every 500 s, penalizing the accuracy in selecting the largest component. As a result, when the radius is small, G_{GT}, G_{obs} and M-iTop show a lower average distance than OPT, which is the result of the fewer number of nodes receiving service.

As Figure 5.8(b) shows, there is a *percolation transition* as we increase the transmission range [26]. In particular, when R_{tx} increase from 50 to 100 m there is a sudden creation of a single *giant component*. Therefore, when $R_{tx} \geq 100$ m, all nodes in the network receive service under all approaches. This explains the peak around $R_{tx} = 100$ m in Figure 5.8(a). This value is close to the *percolation threshold;* therefore, there is a single, sparse, giant component. The sparsity of the component results in a higher hop distance to reach the service. As we increase the transmission radius, the density of



(a)



(b)

Figure 5.8 Average hop distance vs. communication radius (a). Number of served nodes vs. communication radius (b).

the component increases, and consequently the average distance decreases. Overall, M-iTop performs closely to G_{GT}, which however assumes perfect knowledge of the topology.

We further study the performance sensitivity to the nodes' speed. We set the transmission radius to 200 m, which always results in a single connected component, and we vary the speed from 0.1 to 3 m/s. Results are shown in Figure 5.9. When the speed is very low, the graph is more static; therefore, all approaches allow iSR to correctly estimate the largest component. M-iTop achieves a slightly higher average distance due to the non-negligible execution time. All approaches, including OPT, show an increase in the average hop

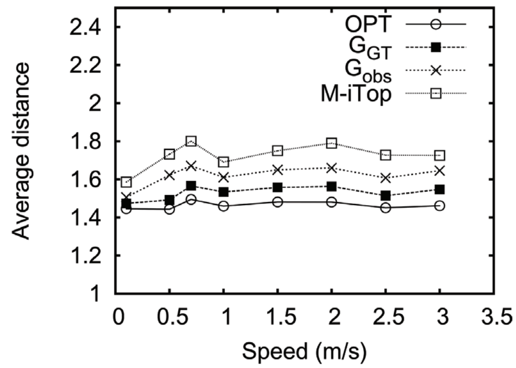


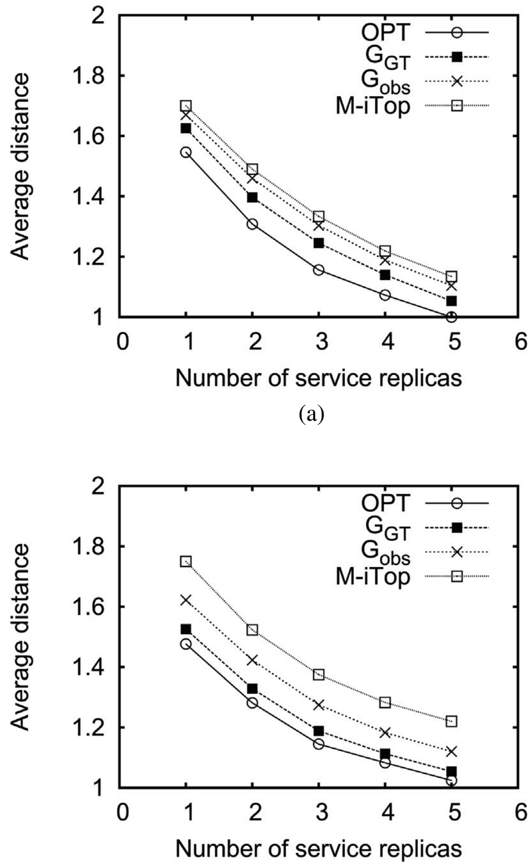
Figure 5.9 Average hop distance vs. nodes' movement speed.

distance as we increase the speed (0.5–0.75 m/s), after which they stabilize around that value, showing very little sensitivity to the speed parameter. This is due to the fact that, when the speed is sufficiently high, the nodes moving farther from the service are balanced by the nodes getting closer to it. Also in this case, M-iTop performs always within 10% from the case in which the observable topology is perfectly known.

5.6.4 Fourth Set of Experiments

The last set of experiments studies the performance of iSR when multiple services are deployed. We set the transmission radius to 200 m to have a single connected component in the network and a speed of 0.5 m/s. We assume that, when multiple services are present, a node receives service from the closest service replica. We compare the average hop distance achieved by iSR when applied to the topologies G_{GT}, G_{obs} and M-iTop. iSR is executed every 500 s.

Figures 5.10(a) and (b) show the average hop distance in a network of 40 and 60 nodes, respectively. As expected, in all approaches, the average distance decreases as we increase the number of replicas deployed. When the network is composed of 40 nodes, M-iTop can be executed in a short time and it is able to accurately infer the observable topology. Therefore, it achieves results similar to those in the case in which the observable topology is perfectly known. When 60 nodes are present, M-iTop is slightly penalized by the running time. Nevertheless, even in this case it performs within 10% of the case in which G_{obs} is known.



(a)

(b)

Figure 5.10 Average hop distance vs. number of service replicas deployed. Network with 40 nodes (a), and network with 60 nodes (b).

5.7 Discussion and Future Research Directions

The M-iTop algorithm provides a good approximation of the observable topology, which enables iSR to select nodes that provide good performance. We next discuss additional aspects which may lead to further performance improvements and will be investigated in our future works.

The M-iTop algorithm mainly focuses on network connectivity and hop distance metrics to perform the merging. The probing phase may provide additional information such as quality of service information on nodes and links. This information can be included in the merging process to perform more accurate merging. Furthermore, M-iTop can be extended in order to return quality of service information on the unobservable part of the network which can be used to improve the service redeployment algorithm.

The complexity of M-iTop can be reduced by considering constraints, such as the nodes' geographical position, that can significantly reduce the size of the merging options sets. The resulting shorter execution time will enable the applicability of the algorithm to large-scale networks.

Finally, movement prediction schemes can be included in the service redeployment algorithm to base the selection on the future expected position of nodes, improving the performance over time.

5.8 Conclusions

In this chapter, for the first time, we apply topology inference techniques to the service redeployment problem in tactical coalition networks. Service redeployment requires accurate topology information which is challenging in coalition networks, where the presence of allied nodes prevents the collection of full topology information. We present M-iTop, an algorithm for topology inference in the presence of partial information, and iSR, an iterative algorithm for multiple service redeployment. We show that M-iTop has a short execution time when applied to networks with the typical size of coalition networks. We perform extensive simulations showing that M-iTop, coupled with iSR, achieves the performance close to that of the optimal redeployment strategy based on complete information.

References

[1] Cirincione, G., and Gowens, J. (2007). The international technology alliance in network and information science. *IEEE Commun. Mag.* 6 (3), 14–18.

[2] Lund, K., Eggen, A., Hadzic, D., Hafsoe, T., and Johnsen, F. T. (2007). Using web services to realize service oriented architecture in military communication networks. *IEEE Commun. Mag.* 45 (10), 47–53.

[3] Lund, K., Skjervold, E., Johnsen, F. T., Hafsøe, T., and Eggen, A. (2010). Robust web services in heterogeneous military networks. *IEEE Commun. Mag.* 48 (10), 78–83.

[4] Novotny, P., Wolf, A. L., and Ko, B. J. (2013). Discovering service dependencies in mobile ad hoc networks. In *IFIP/IEEE IM*, pp. 527–533.

[5] Heydarnoori, A., and Binder, W. (2011). A graph-based approach for deploying component-based applications into channel-based distributed environments. *J. Softw.* 6, (8).
[6] Kalyvianaki, E., Wiesemann, W., Vu, Q. H., Kuhn, D., and Pietzuch, P. (2011). Sqpr: Stream query planning with reuse. In *IEEE ICDE*.
[7] Ooi, B.-Y., Chan, H.-Y., and Cheah, Y.-N. (2012). Dynamic service placement and replication framework to enhance service availability using team formation algorithm. *J. Syst. Softw.* 85 (9), 2048–2062.
[8] Ali, S., Mitschele-Thiel, A., Diab, A., and Rasheed, A. (2010). A survey of services placement mechanisms for future mobile communication networks. In *ACM FIT*.
[9] Herrmann, K. (2010). Self-organized service placement in ambient intelligence environments. *ACM Trans. Autonomous Adapt. Syst.* 5 (2), 6:1–6:39.
[10] Maia, M. E. F., Fonteles, A., Neto, B., Gadelha, R., Viana, W., and Andrade, R. M. C. (2013). Loccam – loosely coupled context acquisition middleware. In *ACM SAC*.
[11] Vega, D., Medina, E., Messeguer, R., Royo, D., and Freitag, F. (2011). A node placement heuristic to encourage resource sharing in mobile computing. In *Springer-Verlag ICCSA*.
[12] Tati, S., Silvestri, S., He, T., and Porta, T. L. (2014). Robust network tomography in the presence of failures. *IEEE ICDCS*.
[13] Famaey, J., Wauters, T., Turck, F. D., Dhoedt, B., and Demeester, P. (2011). Network-aware service placement and selection algorithms on large-scale overlay networks. *Elsevier Comput. Commun.* 34 (15), 1777–1787.
[14] Chadha, R., Poylisher, A., Deb, B., Littman, M., and Sabata, B. (2005). Adaptive dynamic server placement in manets. *IEEE MILCOM*.
[15] Graupner, S., Andrzejak, A., Kotov, V., and Trinks, H. (2005). Adaptive service placement algorithms for autonomous service networks. *Springer Eng. Self-Org. Syst.* 3464, 280–297.
[16] Yao, B., Viswanathan, R., Chang, F., and Waddington, D. (2003). Topology inference in the presence of anonymous routers. *IEEE INFOCOM*.
[17] Jin, X., Yiu, W.-P., Chan, S.-H., and Wang, Y. (2006). Network topology inference based on end-to-end measurements. *IEEE J. Selected Areas in Commun.* 24 (12), 2182–2195.
[18] Sherwood, R., Bender, A., and Spring, N. (2008). Discarte: A disjunctive internet cartographer. *ACM SIGCOMM*.

[19] Gunes, M. H., and Sarac, K. (2008). Resolving anonymous routers in internet topology measurement studies. In *IEEE INFOCOM*.

[20] Holbert, B., Tati, S., Silvestri, S., La Porta, T., and Swami, A. (2015). Network topology inference with partial path information. *IEEE ICNC*.

[21] Kobourov, S. G. (2012). Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv:1201.3011*.

[22] Li, X. (2007). Collaborative localization with received-signal strength in wireless sensor networks. *IEEE Trans. Vehic. Technol.* 56 (6), 3807–3817.

[23] Belghith, A., and Belhassen, M. (2011). Routing enhancements in dynamic manets with obstacles. *IFIP WMNC*, 1–5, Oct.

[24] Jahn, J. (1985). *Scalarization in multi objective optimization*. Springer.

[25] Riverbed Technologies. http://www.riverbed.com/.

[26] Newman, M. (2010). *Networks: An introduction*. Oxford University Press.