

# Minimum Disruption Service Composition and Recovery in Mobile Ad hoc Networks

Shanshan Jiang, Yuan Xue, Douglas Schmidt  
EECS Department, Vanderbilt University  
{shanshan.jiang,yuan.xue,d.schmidt}@vanderbilt.edu

**Abstract**—The dynamic nature of mobile ad hoc networks poses fundamental challenges to the design of service composition schemes that can minimize the effect of service disruptions. Although improving reliability has been a topic of extensive research in mobile ad hoc networks, little existing work has considered service deliveries spanning multiple components. Moreover, service composition strategies proposed for wireline networks are poorly suited for wireless ad hoc networks due to their highly dynamic nature. This paper proposes a new service composition and recovery framework designed to achieve minimum service disruptions for mobile ad hoc networks. The framework consists of two-tiers: *service routing*, which selects the service components, and *network routing*, which finds the network path that connects these service components. Our framework is based on a novel concept: *disruption index*, which characterizes different aspects of service disruptions, including frequency and duration. For ad hoc networks with a known mobility plan, we formulate the problem of minimum-disruption service composition and recovery (MDSCR) as a dynamic programming problem and analyze the properties of its optimal solution. Based on the derived analytical insights, we present our MDSCR heuristic algorithm for ad hoc networks with uncertain node mobility. This heuristic approximates the optimal solution with one-step lookahead prediction, where service link lifetime is predicted using a multivariate linear regression. We evaluate the performance of our algorithm via simulations conducted under various network environments.

## I. INTRODUCTION

Mobile ad hoc networks are self-organized wireless networks formed by mobile nodes. They can be rapidly deployed without the support of fixed infrastructure and are therefore useful in a range of application scenarios, such as disaster relief and military operations. The diverse application domains have fueled an increasing demand for a range of functionalities and services in mobile ad hoc networks. *Service composition* [1], [2], [3] is a crucial technology to meet such demands by integrating loosely coupled distributed service components into a composite service that provides a comprehensive function for end users.

Existing literature on service composition techniques has focused on finding a service path over wireline networks that satisfies various quality of service (QoS) requirements [2], [4], [5] or provides highly available services [6], [7]. While this research has made critical steps towards constructing high quality service paths in a variety of networking environments, it cannot be extended directly to service composition in mobile ad hoc networks since researchers have not considered the intermittent link connectivity and dynamic network topology caused by node mobility. To date, therefore, enabling high-quality service composition in mobile ad hoc networks remains an open issue.

This paper investigates the impact of node mobility and dynamic network topology on service composition. Our goal is to *provide dynamic service composition and recovery strategies that enable highly reliable service delivery that incurs the fewest disruptions to end users in mobile ad hoc networks*. We employ an optimization-based approach to study the best service composition and recovery strategies for ad hoc networks. To achieve this goal, we address the following three challenges:

- *How to quantitatively characterize and measure the impact of service disruptions.* Reliability and availability are two commonly used metrics that quantify the ability of a system to deliver a specified service. For example, the reliability metric helps guide and evaluate the design of many ad hoc routing algorithms [8], [9] and component deployment mechanisms [10]. The basic idea is to use the path with maximum reliability for data/service delivery.

We faced two problems when using reliability as a metric for service composition and recovery design. First, it does not count for any service repair and recovery. Second, reliability is a dynamic metric that is usually estimated based on the signal strength of a wireless link or packet loss ratio along a path. Its constantly changing value may cause repeated service adjustments, especially if an application wants to use the path with maximum reliability. Availability is also insufficient to evaluate the effect of disruptions since it can not characterize the impact of disruption frequency.

- *How to deal with the relation between service routing and network routing.* In an ad hoc network, a service link that connects two service components is supported by the underlying network routing. Its ability to deliver a service therefore depends on the network path in use, *i.e.*, the transient and enduring wireless network link and path failures can constantly change the service delivery capability of a service link. Conversely, service routing determines the selection of service components, which in turn defines the source and destination nodes for network routing. Such interdependencies between service routing and network routing complicate the design of service composition and recovery schemes. To maintain a service with minimum disruption, therefore, routing operations must be coordinated at both the service and network levels.

- *How to realistically integrate the knowledge of node mobility in the service composition and recovery strategies.* Node mobility is a major cause of service failures in ad hoc networks. To ensure highly reliable service delivery and reduce service disruptions, therefore, we need to predict the sustainability of service links based on node mobility patterns. Accurate prediction is hard, however, for the following reasons: (1) the mobility-caused link failures are highly dependent and (2) the sustainability of a service link is also affected by the network path repair operations and the new nodes emerging in its vicinity.

To address these challenges, we created a new service composition and recovery framework for mobile ad hoc networks to achieve minimum service disruptions. The framework consists of two-tiers: (1) *service routing*, which selects the service components that support the service delivery, and (2) *network routing*, which finds the network path that connects these service components. We built our framework on a novel concept: the *disruption index*, which characterizes different service disruption aspects, such as frequency and duration, better than conventional metrics, such as reliability and availability.

For ad hoc networks with known mobility plan, we formulate the problem of *minimum-disruption service composition and recovery* (MDSCR) as a dynamic programming problem and analyze the properties of its optimal solution. Based on the derived analytical insights, we present our MDSCR heuristic algorithm for ad hoc networks with uncertain node mobility. This heuristic approximates the optimal solution with one-step lookahead prediction, where the sustainability of a service link is modeled through its lifetime and predicted using a multivariate linear regression.

This paper makes the following contributions to work

on service composition and recovery in mobile ad hoc networks: (1) it creates a theoretical framework for service composition and recovery strategies for ad hoc networks that characterizes the effect of service disruption; (2) it uses dynamic programming analysis to present a set of properties (*e.g.*, reactive recovery and the relation between service-level recovery and network-level recovery) for the minimum disruption service composition and recovery strategies, which provides important analytical insights for MDSCR heuristic algorithm design; (3) it presents a simple yet effective statistical model based on multivariate linear regression that predicts the lifetime of a service link in the presence of highly correlated wireless link failures and the network path repairs.

The remaining of this paper is organized as follows: Section II provides the network and service model; Section III describes the service composition and recovery framework for ad hoc networks. Section IV formulates the MDSCR problem and analyzes the properties of the optimal solution; Section V describes the heuristic MDSCR algorithm; Section VI evaluates our simulation results; and Section VII presents concluding remarks.

## II. NETWORK AND SERVICE MODEL

We consider a wireless ad hoc network consisting of a set of mobile nodes  $\mathcal{N}$ . In this network, link connectivity and network topology change with node movement. We model this mobile ad hoc network at time  $t$  as  $\mathcal{G}(t) = (\mathcal{N}, \mathcal{L}(t))$ , where  $\mathcal{L}(t)$  represents the set of wireless links at time  $t$ , *i.e.*, for link  $l = (n, n') \in \mathcal{L}(t)$ , nodes  $n$  and  $n'$  are within the transmission range of each other.<sup>1</sup> We further denote a network path that connects node  $n_1$  and  $n_m$  in this graph as  $\mathcal{P}_{(n_1, n_m)}(t) = (n_1, n_2, \dots, n_m)$ , where  $(n_j, n_{j+1}) \in \mathcal{L}(t)$ . We also use  $|\mathcal{P}(t)|$  to denote the path length of  $\mathcal{P}(t)$ .

To characterize the structure of distributed applications that are expected to run in the mobile environments, we apply a component-based software model [11]. All application components are constructed as *autonomous services*, which perform independent operations (such as transformation and filtering) on the data stream passing through them. This paper focuses on the *uni-cast service connectivity*, *i.e.*, service components are linked in a sequence order with only one receiver. We call such a composed service a *service path* and denote it as  $\mathcal{S} = (s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_r)$ , where  $s_k (k = 1, \dots, r - 1)$  is a service component, and  $s_r$  is the service receiver. Moreover, we call one hop in a service path ( $s_k \rightarrow s_{k+1}$ ) a *service link*.

<sup>1</sup>For simplicity, we only consider bi-directional wireless links in this work.

In an ad hoc network, each service component  $s_k$  can be replicated at multiple nodes to improve the service availability, [12]. We denote the set of nodes that can provide services  $s_k$  as  $\mathcal{N}_k \subseteq \mathcal{N}$  and the service  $s_k$  that resides on node  $n$  as  $s_k[n], n \in \mathcal{N}_k$ . Figure 1 shows an example service deployment and service paths. Note that a service link is an overlay link that may consist of several wireless links in the network, *i.e.*, a network path. In the figure, the service link ( $s_1[a] \rightarrow s_2[f]$ ) is supported by the network path ( $a, e, f$ ).

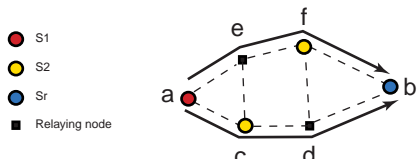


Fig. 1. Example Service Deployment and Service Paths

The composed service usually needs to satisfy certain QoS requirements. To have a focused discussion on the impact of service failures caused by node mobility, this paper considers a simple QoS metric – *service link length*, which is the number of wireless links traversed by a service link. In particular, we require the service link length be bounded by  $H$  hops.

### III. A SERVICE COMPOSITION AND RECOVERY FRAMEWORK FOR AD HOC NETWORK

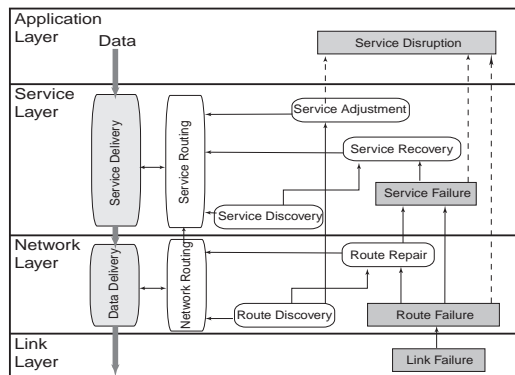


Fig. 2. A Service Composition and Recovery Framework in a Mobile Ad hoc Network

*Service composition* refers to the process of finding a service path in the network. As shown in Figure 2, service composition in a wireless ad hoc network involves the following two tightly-coupled processes:

- *Service routing*, which selects the service components (out of many replicas) for the service path. It relies on service component discovery [13], [14] to find the candidate service components, then selects the appropriate ones to compose a service path. Formally, a service routing scheme is represented as  $\pi_S = (s_1[n_1], s_2[n_2], \dots, s_r[n_r])$ , where  $n_k \in \mathcal{N}_k$  is the hosting node for the selected service component  $s_k$ .

- *Network routing*, which finds the network path that connects the selected service components. Formally, the network routing scheme could be represented as a set of paths  $\pi_N = \{\mathcal{P}_{(n_k, n_{k+1})}, k = 1, \dots, r - 1\}$  where  $\mathcal{P}_{(n_k, n_{k+1})}$  represents the network path that supports the service link ( $s_k[n_k] \rightarrow s_{k+1}[n_{k+1}]$ ).

These two processes closely interact with each other. The component selection in service routing determines the source and destination nodes in network routing. Likewise, the path quality in network routing also affects the selection of service components in service routing. Collectively, a service composition scheme is represented as  $\pi = (\pi_S, \pi_N)$ .

A service failure may occur due to a violation of its QoS or failures of service components and/or service links along its service path. This paper focuses on *service failures caused by node mobility*. In an ad hoc network, wireless links may fail due to node mobility, which may cause failures of service links and in turn service path failures.

To sustain service delivery, the service path must be repaired. This repair process essentially *recomposes the service path* and is called *service recovery*. Service recovery is triggered by service failure detection at either link-level (*e.g.*, via IEEE 802.11 ACK frame), network-level (*e.g.*, through HELLO messages), or service-level. Similar to service composition, service recovery process also involves two processes, namely, *network-level recovery*, which repairs the data path between two components, and *service-level recovery*, which replaces one or more service components. Network-level recovery usually depends on the specific ad hoc routing protocol in use and the route repair mechanism built within this routing protocol. Service-level recovery involves discovery of new components and establishment of a new service path.

Service recovery differs from service composition since it must consider not only the quality of the recomposed (repaired) path, but also the service path previously in use (the one that just failed). Intuitively, to reduce the repair overhead and recovery duration, we prefer a service path that could maximally reuse the current nodes/components. Using such a service recovery strategy, however, the new service path may have a poor QoS and/or may fail soon in the future.

Though node mobility can cause service failures, it can sometime enable a better service path. *Service adjustment* is the process of modifying the current service path for better QoS or higher reliability by using new network path(s) or new component(s) that appear in the vicinity. Similar to the dilemma faced by service recovery, however, such changes can disrupt the service,

even though they improve the sustainability and quality of the new path.

#### IV. MDSR THEORETICAL FRAMEWORK

A fundamental research challenge for service recovery is *how to best tradeoff the time and overhead involved in service recovery and adjustment and the sustainability of composed service path so that end users will perceive minimum disruptions to the service during its lifetime*. To address this challenge, we need a theoretical framework that allows us to analytically study the problem of service composition, adjustment, and recovery strategies to achieve minimum service disruptions. This section establishes such an optimization-based theoretical framework based on dynamic programming that quantitatively characterizes the impact of service disruption.

##### A. Service Disruption Model

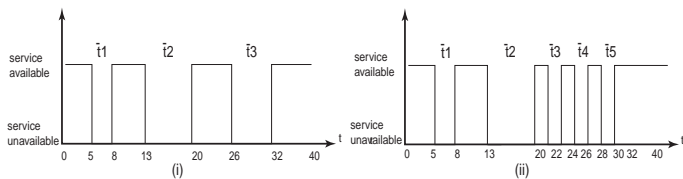


Fig. 3. Example Service Disruption Processes

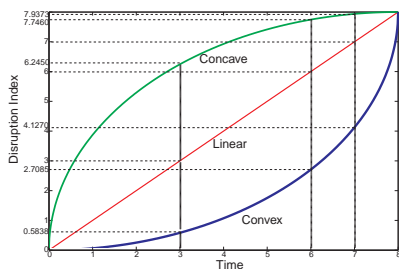


Fig. 4. Example Disruption Penalty Function

Consider a service  $\mathcal{S}$  that starts at time instance 0 and ends at  $T$ . Figure 3 shows an example of two service disruption processes. Let  $\bar{t}_1, \bar{t}_2, \dots, \bar{t}_q$  be the sequence of disruption durations. A classical way to model service disruption is *service availability*, which is defined as the fraction of service available time during the service lifetime:  $A = \frac{T - \sum_{i=1}^q (\bar{t}_i)}{T}$ . Using availability as the metric to characterize the impact of service disruption, however, we face the following two problems:

- *Service availability cannot characterize the impact of service failure frequency*. For example, in Figure 3, scenario (i) and (ii) have the same service availability ( $\frac{24}{40}$ ). The user-perceived disruption could be different, however, since scenario (ii) has a higher service failure frequency but smaller disruption durations. To precisely model the effect of service disruption, therefore, we need

a new metric that characterizes both failure durations and failure frequency.

- *Service availability is hard to compute*. The calculation of service availability is based on the calculation of disruption durations, which include the service failure time and recovery time. Such durations are determined by many factors, such as network topology, routing protocol, and system conditions, which are dynamic and hard to be incorporated into service composition and recovery decisions. To establish a theoretical framework that provides realistic insight to implementation of service composition and recovery strategy, we need a metric that is stable, easily computed, and can provide a good estimation of disruption durations.

To address the first problem regarding the impact of service failure frequency, we associate a *disruption penalty function*  $F(\bar{t})$  defined over the disruption duration  $\bar{t}$  with an end user. The shape of  $F(\bar{t})$  reflects its relative sensitivity to disruption duration and frequency. Figure 4 shows three basic types of failure penalty functions. We further define *disruption index*  $D$  as a metric to characterize the impact of service disruption, as follows:

$$D = \frac{\sum_{i=1}^q F(\bar{t}_i)}{T} \quad (1)$$

To show how the disruption index  $D$  characterizes the user-perceived disruption effect and integrates both disruption duration and failure frequency, we calculate the disruption indices for the two service disruption processes in Figure 3 using the different failure penalty functions shown in Figure 4. The results are summarized in Table I.

Penalty Function	linear	concave	convex
Process (i)	$\frac{16}{40}$	21.9282	7.4193
Process (ii)	$\frac{16}{40}$	30.0568	5.4729

TABLE I

DISRUPTION INDICES UNDER DIFFERENT PENALTY FUNCTIONS

Table I shows that if  $F(\bar{t})$  is a concave function then disruption process (ii) has a higher disruption index than process (i), *i.e.* its end user is more sensitive to failure frequency. When  $F(\bar{t})$  is a convex function, disruption process (i) has a higher disruption index than process (ii), *i.e.*, its end user is more impatient to disruptions with long duration. For a linear disruption penalty function the user is neutral, and the disruption index depends on the service availability.

To address the second problem regarding computing service availability, we present simple and stable estimations of disruption durations for network-level recovery and service-level recovery separately.

1) *Estimation for network-level recovery*: For network-level recovery, the service components remain the same, *i.e.*, we only need to repair the network path that connects them. Typical network-level recovery processes in repairing a network path in ad hoc networks [15] involve discovering an alternative route to replace the broken link/path and restarting the data delivery. Here we use the number of wireless link substitutions in the repair as a simple estimate for the disruption duration introduced by network-level recovery. Formally, let  $\mathcal{P}$  and  $\mathcal{P}'$  be the paths before and after recovery. We use  $N_{\mathcal{P} \rightarrow \mathcal{P}'}$  to denote the number of link substitutions from  $\mathcal{P}$  to path  $\mathcal{P}'$ . Let  $\mathcal{P} \cap \mathcal{P}'$  be the set of common links in these two paths, then  $N_{\mathcal{P} \rightarrow \mathcal{P}'} = |\mathcal{P}'| - |\mathcal{P} \cap \mathcal{P}'|$ .

Using the number of wireless link substitutions as an estimate for disruption duration introduced by network-level recovery is consistent with typical ad hoc network repair operations. For example, there are usually two repair mechanisms in ad hoc routing: *local repair* and *global repair*. For local repair, when a link fails, one of its end nodes will try to find an alternative path in the vicinity to replace this link. Local repair therefore involves fewer link substitutions and less recovery time. For global repair, the source node initiates a new route discovery, which takes more time than local repair and involves more link substitutions.<sup>2</sup>

2) *Estimation for service-level recovery*: A service-level recovery involves three operations: (1) finding the appropriate substitution components, (2) starting the new components and restoring the service states, and (3) finding a network path that supports the connectivity between the new components. Service-level recovery thus takes much more time than a network-level recovery. Similar to network-level recovery, the duration of service-level recovery depends largely on the searching/replacing scope of the service components. We can therefore use the number of substituted components to estimate its recovery duration. Formally, let  $\pi_{\mathcal{S}}$  and  $\pi'_{\mathcal{S}}$  be the service routing schemes before and after recovery. We use  $N_{\pi_{\mathcal{S}} \rightarrow \pi'_{\mathcal{S}}}$  to represent the number of component substitutions from  $\pi_{\mathcal{S}}$  to  $\pi'_{\mathcal{S}}$ . Then  $N_{\pi_{\mathcal{S}} \rightarrow \pi'_{\mathcal{S}}} = r - |\pi_{\mathcal{S}} \cap \pi'_{\mathcal{S}}|$ , where  $r$  is the service path length and  $|\pi_{\mathcal{S}} \cap \pi'_{\mathcal{S}}|$  is number of common components in these two sets.

Based on the recovery duration estimation, we now proceed to redefine the disruption index. Consider a service  $\mathcal{S}$  that starts at time instance 0 and ends at  $T$ . Let  $\pi(t_1), \pi(t_2), \dots, \pi(t_l)$  be the sequence of service composition schemes used during the service lifetime.

The disruption duration  $\bar{t}_k$  from service composition  $\pi(t_k)$  to  $\pi(t_{k+1})$  is estimated as

$$\begin{aligned} \bar{t}_k &= \beta \times N_{\pi(t_k) \rightarrow \pi(t_{k+1})} \\ &= \beta \times (N_{\pi(t_k) \rightarrow \pi(t_{k+1})}^{\mathcal{N}} + \alpha N_{\pi(t_k) \rightarrow \pi(t_{k+1})}^{\mathcal{S}}) \end{aligned} \quad (2)$$

where  $N_{\pi(t_k) \rightarrow \pi(t_{k+1})}^{\mathcal{N}}$  and  $N_{\pi(t_k) \rightarrow \pi(t_{k+1})}^{\mathcal{S}}$  denote the number of substituted wireless links in network-level recovery (if any) and the number of substituted components in service-level recovery (if any) incurred by the service composition transition from  $\pi(t_k)$  to  $\pi(t_{k+1})$ .  $\beta$  is the parameter that converts the number of substitutions to disruption time.  $\alpha \gg 1$ , denotes the relative weight between service component substitution and link substitution on disruption duration. Based on the discussions above, the disruption index  $D$  could be estimated as

$$\tilde{D} = \frac{\sum_{k=1}^{l-1} F(\beta \times N_{\pi(t_k) \rightarrow \pi(t_{k+1})})}{T} \quad (4)$$

### B. MDSCR Problem Formulation

We now formulate the *minimum disruptive service composition and recovery* (MDSCR) problem. First, we define a service composition and recovery policy as a sequence of service composition schemes:  $\Pi = (\pi(t_1), \pi(t_2), \dots, \pi(t_l))$ , where  $0 = t_1 < \dots < t_l \leq T$ .  $\pi(t_k)$  gives the service composition during time  $[t_k, t_{k+1})$ . We say service composition  $\pi(t_k)$  is *feasible* on network  $\mathcal{G}(t_k)$ , if and only if all the network paths in  $\pi_{\mathcal{N}}(t_k)$  exist on  $\mathcal{G}(t_k)$ ; Policy  $\Pi$  is *feasible* if and only if each of its service composition  $\pi(t_k)$  is feasible. Note that  $\Pi$  gives initial service composition  $\pi(t_1)$  and all the service recovery schemes  $\pi(t_k) \rightarrow \pi(t_{k+1})$ ,  $k = 1, \dots, l-1$ .

We denote the set of all feasible service composition policies over mobile ad hoc network  $\mathcal{G}(t)$  as  $\Phi(\mathcal{G})$ . For a feasible service policy  $\Pi \in \Phi(\mathcal{G})$ , there is a corresponding disruption index  $\tilde{D}(\Pi)$  defined as:

$$\tilde{D}(\Pi) = \frac{\sum_{k=1}^{l-1} F(\beta \times N_{\pi(t_k) \rightarrow \pi(t_{k+1})})}{T} \quad (5)$$

The goal of the MDSCR algorithm is to find the best policy  $\Pi \in \Phi(\mathcal{G})$  that is feasible for  $\mathcal{G}(t)$ , so that  $\tilde{D}(\Pi)$  is minimized. Formally,

$$\mathbf{MDSCR} : \text{ minimize } \tilde{D}(\Pi) \quad (6)$$

$$\Pi \in \Phi(\mathcal{G}) \quad (7)$$

If the graph series  $\mathcal{G}(t)$  is given, which means that the mobility plan is determined a priori, the optimization problem **MDSCR** could be solved using dynamic programming. The mobility plan, however, is usually unavailable, *i.e.*,  $\mathcal{G}(t)$  is unknown in practice. To derive a practical solution for MDSCR problem, we first study the optimal MDSCR solution and derive its analytical properties. Based on these analytical insights, we then present the MDSCR heuristics in Section V.

<sup>2</sup>For simple estimation, we do not consider the impact of route caches here.

### C. Optimal Solution

If  $\mathcal{G}(t)$  is given, the minimum disruptive service composition problem **MDSCR** is essentially a dynamic programming problem. Let  $\mathcal{J}(\pi(t_w))$  be the minimum disruption index for the service disruption experienced by the service from time instance  $t_w$  when composition scheme  $\pi(t_w)$  is used, *i.e.*:

$$\mathcal{J}(\pi(t_w)) = \min_{\Pi \in \Phi(\mathcal{G})} \frac{\sum_{k=w}^{l-1} F(\beta \times N_{\pi(t_k) \rightarrow \pi(t_{k+1})})}{T} \quad (8)$$

From Eq. (5) and Eq. (8)  $\mathcal{J}(\pi(t_1)) = \min_{\Pi \in \Phi(\mathcal{G})} \tilde{D}(\Pi)$ . Based on dynamic programming, we have

$$\mathcal{J}(\pi(t_w)) = \min_{\pi(t_{w+1})} \left\{ \frac{F(\beta \times N_{\pi(t_w) \rightarrow \pi(t_{w+1})})}{T} + \mathcal{J}(\pi(t_{w+1})) \right\} \quad (9)$$

When the mobility plan of the ad hoc network is known, the equation shown above could be used to give the optimal MDSCR solution via standard dynamic programming techniques [16]. In particular, solving  $\mathcal{J}(\pi(t_1))$  gives the optimal initial service composition  $\pi(t_1)$ . At time  $t_w$  with service composition scheme  $\pi(t_w)$ , solving Eq. (9) gives the optimal service recovery scheme (minimum disruption service recovery) that changes the service composition from  $\pi(t_w)$  to  $\pi(t_{w+1})$ .

### D. Analysis

The optimal solution described above reveals the following interesting properties for MDSCR strategies:

- **Reactive recovery.** The first property of an optimal solution is the reactive adjustment and recovery strategy. Specifically, if the failure penalty function  $F$  is a linear or concave function (neutral or disruption frequency sensitive user), a service path is changed if and only if one of the underlying wireless link used by the service path is broken in an optimal MDSCR strategy. This property means that service composition remains the same on the discovery of new nodes and new service components in the neighbor (no service adjustment) and the node failures that are not on the service path. Formally, this property is presented in the following theorem.

**Theorem 1:** Let  $\Pi^* = (\pi^*(t_1), \dots, \pi^*(t_l))$  be the optimal MDSCR policy, and  $F$  be a concave or linear function. Then for any two consecutive service compositions  $\pi^*(t_w)$  and  $\pi^*(t_{w+1})$ ,  $\pi^*(t_w)$  is **not** feasible on the network topology  $\mathcal{G}(t_{w+1})$ .

- **Reactive service-level recovery.** When  $\alpha \gg 1$ , for an optimal solution service-level recovery is invoked if and only if network-level recovery can not repair one of the service link in use (*i.e.*, there is no feasible network path connecting these two service components). This property is formally summarized in the following theorem.

**Theorem 2:** Let  $\Pi^* = (\pi^*(t_1), \dots, \pi^*(t_l))$  be the optimal MDSCR policy. Assume  $\alpha \gg 1$ . Consider a sub-sequence of  $\Pi^*$  where service routing schemes are changed. We denote this sub-sequence only with its service routing scheme as  $\Pi_S^* = (\pi_S^*(t_1^s), \dots, \pi_S^*(t_g^s))$ . Then for any two consecutive service compositions in  $\Pi_S^*$ ,  $\pi_S^*(t_w^s)$  and  $\pi_S^*(t_{w+1}^s)$ ,  $\pi_S^*(t_w^s)$  is **not** feasible on the network topology  $\mathcal{G}(t_{w+1}^s)$  at  $t_{w+1}^s$ , *i.e.*, there exists a service link in  $\pi_S^*(t_w^s)$  which has no feasible network path in  $\mathcal{G}(t_{w+1}^s)$ .

Due to space constraints, the proofs for these two theorems are given in [17].

## V. MDSCR HEURISTIC ALGORITHM

### A. Two-tier MDSCR Algorithm

Based on the analysis in Sec IV-D, we can reduce the complexity of MDSCR problem by decomposing it into two sub-problems: (1) the service-level MDSCR problem and (2) the network-level MDSCR problem. The service-level MDSCR is the primary problem. Its objective is to minimize the service-level disruption index  $\tilde{D}_S$  via service routing, where  $\tilde{D}_S$  is defined as

$$\tilde{D}_S = \frac{1}{T} \sum_{k=1}^{g-1} F(\beta \alpha N_{\pi_S(t_k^s) \rightarrow \pi_S(t_{k+1}^s)}^S) \quad (10)$$

Here  $\Pi_S = (\pi_S(t_1^s), \dots, \pi_S(t_g^s))$  is the sub-sequence of service routing schemes whose service components are changed from  $\Pi$ . In particular, the initial service composition solution at the service level is given by solving the following equation:

$$\mathcal{J}(\pi_S(t_1^s)) = \min \frac{1}{T} \sum_{k=1}^{g-1} F(\beta \alpha N_{\pi_S(t_k^s) \rightarrow \pi_S(t_{k+1}^s)}^S) \quad (11)$$

At time  $t_{w+1}^s$  with service routing scheme  $\pi_S(t_w^s)$ , the service recovery scheme that changes the service path from  $\pi_S(t_w^s)$  to  $\pi_S(t_{w+1}^s)$  is given by solving the following equation:

$$\begin{aligned} & \mathcal{J}(\pi_S(t_w^s)) \\ &= \min_{\pi_S(t_{w+1}^s)} \left\{ \frac{1}{T} F(\beta \alpha N_{\pi_S(t_w^s) \rightarrow \pi_S(t_{w+1}^s)}^S) + \mathcal{J}(\pi_S(t_{w+1}^s)) \right\} \end{aligned} \quad (12)$$

The network-level MDSCR is the secondary problem. It tries to minimize the disruption index caused by network-level recovery during the lifetime of a service link. Formally, its objective is to minimize the network-level disruption index  $\tilde{D}_N$  (defined as follows) during the lifetime of each service link via network routing.

$$\tilde{D}_N(t_w^s \rightarrow t_{w+1}^s) = \frac{1}{T} \sum_{t=t_w^s}^{t_{w+1}^s} F(\beta N_{\pi(t) \rightarrow \pi(t+1)}^N) \quad (13)$$

### B. One-step Look-ahead Approximation

The decomposition mechanism presented above separates concerns in MDSCR into two-levels, so that service-level MDSCR and network-level MDSCR can be treated separately. Here we focus our discussion on service-level MDSCR and rely partially on the existing ad hoc network routing protocols for network-level MDSCR.

Finding the solution to service-level MDSCR problem is still impossible for ad hoc networks with uncertain mobility plan since it needs the complete knowledge of future network topologies. Specifically, the service recovery decision at  $t_{w+1}^s$  requires the knowledge of network topology after this time to calculate the future disruption index  $\mathcal{J}(\pi_S(t_{w+1}^s))$ . To address this problem, we present a one-step look-ahead approximation method where future disruption index is estimated in the time period until its first service-level path failure. When this failure occurs, its number of component substitutions is approximated by an average value  $E(N^S)$ .

Formally, let  $L_{n_k \rightarrow n_{k+1}}$  be the expected lifetime<sup>3</sup> for the service link  $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ . The service routing scheme at time  $t_{w+1}^s$  is  $\pi_S(t_{w+1}^s) = (s_1[n_1], s_2[n_2], \dots, s_r[n_r])$ . Its failure rate is estimated as  $\gamma_{\pi_S(t_{w+1}^s)} = \sum_{k=1}^{r-1} \frac{1}{L_{n_k \rightarrow n_{k+1}}}$ . Likewise,  $\mathcal{J}(\pi_S(t_{w+1}^s))$  is estimated as

$$\hat{\mathcal{J}}(\pi_S(t_{w+1}^s)) = F(\beta\alpha \times E[N^S]) \times \gamma_{\pi_S(t_{w+1}^s)} \quad (14)$$

The initial service composition strategy is to find  $\pi_S(t_1^s)$  to minimize

$$F(\beta\alpha \times E[N^S]) \times \gamma_{\pi_S(t_1^s)} \quad (15)$$

The service-level recovery strategy involves finding a service routing scheme  $\pi_S(t_{w+1}^s)$  to minimize

$$\frac{1}{T} F(\beta\alpha N^S_{\pi_S(t_w^s) \rightarrow \pi_S(t_{w+1}^s)}) + F(\beta\alpha E[N^S]) \gamma_{\pi_S(t_{w+1}^s)} \quad (16)$$

Eq. (16) formally characterizes the trade-off between the recovery duration (first term) and the sustainability of the newly composed path (second term) faced by service recovery.

### C. Lifetime Prediction

Now the problem left in deriving a practical MDSCR solution is to estimate the service link lifetime. This problem is a non-trivial due to the highly inter-dependent wireless link failures and the impact from network path repairs. It therefore cannot be solved by traditional network path reliability estimation methods.

<sup>3</sup>Here the lifetime of a service link is defined as the time interval between its formation and the first time instance when the length of the shortest network path that supports this service link is larger than  $H$  hops.

To address this challenge, we present a service link lifetime prediction method based on multivariate linear regression. First, we identify a list of factors (e.g., distance between two components) that affect the service link lifetime. We then conduct a set of experiments and record the service link lifetimes with the corresponding values of these factors. We identify the most significant factors via principal components analysis and derive the relations among these factors via multivariate linear regression.

Using the method described above, we estimate the lifetime of service link  $(n \rightarrow n')$  based on (1) predicted distance between two components  $\tilde{d}_{n \rightarrow n'}(t + \Delta t)$ , which is calculated based on the current locations of the hosting nodes, their velocities and the prediction time  $\Delta t$ , and (2) the node density  $\rho_{n \rightarrow n'}(t)$  in the vicinity of the service link. Figure 5 plots the relation among the service link lifetime, its predicted distance and node density in a set of experiments.<sup>4</sup>

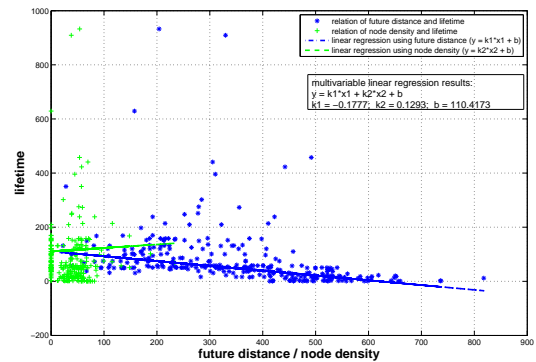


Fig. 5. Lifetime Prediction

The blue (dark) star in Figure 5 describes the relation of the predicted distance (x-value) and the service link lifetime (y-value); and the blue (dark) line is its linear regression results. The green (light) cross in the figure describes the relation of the node density and the link lifetime; and the green (light) line is its linear regression results. Based on the experiment results described above, the lifetime of a service link is computed via multivariate linear regression shown as follows.

$$L_{n \rightarrow n'} = K_1 \times \tilde{d}_{n \rightarrow n'}(t + \Delta t) + K_2 \times \rho_{n \rightarrow n'}(t) + B \quad (17)$$

where  $K_1 = -0.1777$ ;  $K_2 = 0.1293$ ; and  $B = 110.4173$ . In the simulation study, we derive the corresponding parameters for linear regression for different network setups, and pick the prediction time with the smallest standard error.

<sup>4</sup>The simulation parameters used this experiment are the same as the default ones in Section VI.

### D. Two-tier Predictive Heuristic Algorithm

We now summarize the discussions above and present the MDSCR heuristic algorithms. The deployment of our algorithm needs the support of location services [18] for node location and velocity information, and service discovery services [14]. Table II gives the minimum disruption service composition algorithm. This algorithm

<b>Alg. I: Minimum Disruption Service Composition</b>	
1	<i>First tier: service routing</i>
1.1	For all feasible service links $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ whose shortest underlying network path length $\leq H$ Estimate lifetime $L_{n_k \rightarrow n_{k+1}}$ .
2.2	Find the service routing scheme $\pi_S$ that minimizes Eq. (15). //This could be done based on any minimum cost routing algorithm
2	<i>Second tier: network routing</i>
2.1	For each service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ Find the network path with the maximum estimated lifetime and length $\leq H$ . $\mathcal{P}_{(n_k, n_{k+1})} \leftarrow MLNR(n_k, n_{k+1}, \mathcal{G})$ // MLNR is a minimum path failure rate routing algorithm that could be done based on any minimum cost routing algorithm

TABLE II

MINIMUM DISRUPTION SERVICE COMPOSITION ALGORITHM

has two tiers. The first tier is service routing that finds the service components for the service path. Once the service components are determined, the network routing algorithm in the second tier will find the data path to connect these components.

Table III gives the minimum disruption service recovery algorithm. This algorithm also has two tiers. The first tier is the network-level recovery, which is triggered by the failure of a wireless link on the current service path. If the network-level recovery succeeds the algorithm return successfully. If the network-level recovery fails, however, then service-level recovery will be triggered.

## VI. SIMULATION STUDY

This section evaluates the performance of our MDSCR algorithm via simulation.

### A. Simulation Setup

In the simulated ad hoc network, 50 nodes are randomly deployed over a  $2,000 \times 1,000m^2$  region. Each node has a transmission range of  $250m$ . Node mobility follows the random waypoint model. In this model, a node chooses a random destination and moves towards it with a constant speed uniformly distributed between zero and a *maximum speed* (default value is  $10ms$ ). When a node reaches its destination, it chooses a new destination

### Alg. II: Minimum Disruption Service Recovery

//Assume a wireless link that supports service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ fails	
1	<i>First tier: network-level recovery</i>
1.1	For all feasible network path $\mathcal{P}_{(n_k, n_{k+1})}$ with length $\leq H$ Estimate lifetime $L_{n_k \rightarrow n_{k+1}}$ . If no such feasible network path exists, goto 2
1.2	Find the network path with the maximum estimated lifetime return the path. //network-level recovery succeeds. // network-level recovery fails, try service-level recovery
2	<i>Second tier: service-level recovery</i>
//Assume the current service routing scheme is $\pi_S(t_w^s)$	
2.1	For all feasible service links $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ whose shortest underlying network path length $\leq H$ Estimate lifetime $L_{n_k \rightarrow n_{k+1}}$ .
2.2	Find the service routing scheme $\pi_S(t_{w+1}^s)$ that minimizes Eq. (16) //then perform network routing
2.3	For each service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ in $\pi_S(t_{w+1}^s)$ Find the network path with the maximum estimated lifetime and a length $\leq H$ . $\mathcal{P}_{(n_k, n_{k+1})} \leftarrow MLNR(n_k, n_{k+1}, \mathcal{G})$

TABLE III

MINIMUM DISRUPTION SERVICE RECOVERY

and begins moving towards it after waiting for certain *pause time* (default value is  $10s$ ). Each simulation runs for  $2000s$  (also the service lifetime).

The simulated service is composed from 3 components; each component has 8 replicas by default. Each service link requires its network path length  $H \leq 3$ . In the simulation, the prediction time is adjusted based on network setup to achieve the smallest prediction error (in default setting, prediction time is  $20s$ ). The value of  $\alpha$  is 10,  $\beta$  is 1.

We compare the performance of our MDSCR algorithm with the shortest path service composition (SPSC) algorithm, which is a commonly used ad hoc routing algorithm that chooses the path with the smallest hop number. In particular, the length of a service link is calculated as the length of the shortest network path that supports it. The service path with the shortest service link length will be chosen. For each experiment, we run both MDSCR algorithm and SPSC algorithm over the same network scenario (*i.e.*, each node in two runs of the simulation follows the same trajectory) and compare their disruption indices.



## B. Basic Comparison

We first experiment the MDSCR and SPSC algorithms over 50 different random network topologies. The network parameters used in the experiments were based on their default values. The simulation results are shown in Figure 6. In the figure, the y-axis shows the improve-

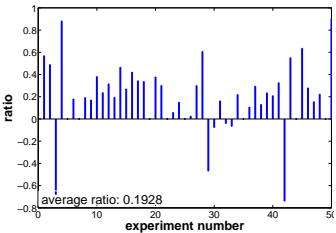


Fig. 6. Improvement ratio under default simulation parameters

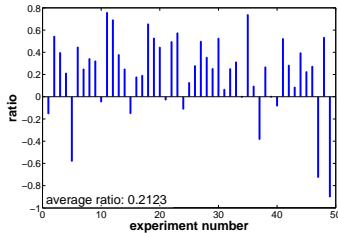


Fig. 7. Improvement ratio with a service path length of 4

ment ratio, which is defined as  $\text{improvement ratio} = \frac{\tilde{D}_{MDSCR} - \tilde{D}_{SPSC}}{\tilde{D}_{SPSC}}$ , where  $\tilde{D}_{MDSCR}$  and  $\tilde{D}_{SPSC}$  are the disruption indices of the MDSCR and SPSC algorithms, respectively. Figure 6 shows that the MDSCR algorithm outperforms the SPSC algorithm in most experiments by an average ratio of 19.28%. There are also several cases where the SPSC algorithm outperforms the MDSCR algorithm, due to the prediction errors in these scenarios, *e.g.*, the node moves towards the opposite direction right after our prediction.

## C. Impact of Service Path Length

We next measure the impact of service path length (*i.e.*, the number of service components involved in the service delivery) on the performance of our algorithm. This simulation adjusts the service path length from 3 to 4. The improvement ratios under 50 experiments are shown in Figure 7. From the results, we can see that MDSCR algorithm consistently outperforms SPSC algorithm under both service path lengths. Comparing Figure 6 with Figure 7, we also observe that the average improvement ratio 21% with longer service path length (4) is better than the one with service path length as 3. This result shows that the benefit of MDSCR algorithm increases relative to SPSC when the service path length gets longer, *i.e.*, more service components are involved in the service composition.

## D. Impact of $F$ Function

In the simulation described above, the failure penalty function  $F$  takes a linear function. We now study the performance of the MDSCR algorithm under different shapes of the  $F$  function. Figures 8 and 9 show the improvement ratios under concave and convex functions, respectively. These figures show that the convex function  $F$  gives a larger improvement ratio than the concave

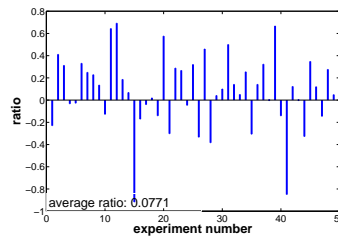


Fig. 8. Function  $F$  is concave

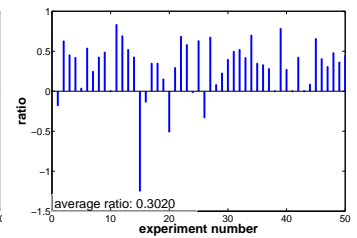


Fig. 9. Function  $F$  is convex

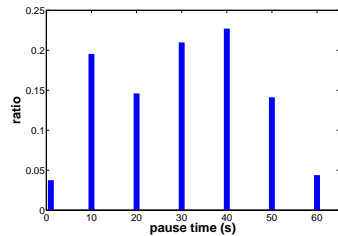


Fig. 10. Impact of pause time on improvement ratio

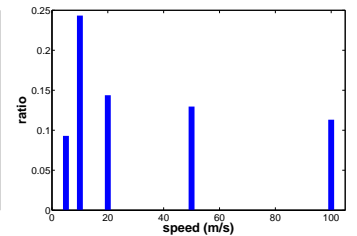


Fig. 11. Impact of node speed on improvement ratio

function. This result occurs because under convex function, local recovery (which tries to replace as few components/links as possible) incurs much less disruption penalty than global recovery due to the convex shape. Our MDSCR heuristic algorithm aggressively encourages local recovery and thus performs much better than SPSC. In the concave region, conversely, the benefits of local recovery are not significant, and the advantages of MDSCR are thus less prominent.

## E. Impact of System Dynamics

To analyze the impact of system dynamics, we simulate both the MDSCR and SPSC algorithms under different node speeds and pause times. In particular, we experiment with pause times of 1s, 10s, 20s, 30s, 40s, 50s, 60s and node speeds of 5ms, 10ms, 20ms, 50ms, 100ms. The prediction time is also adjusted in each experiment to reflect the best prediction results (*i.e.*, the smallest standard error in linear regression). We plot the average improvement ratios of 50 experiments under different pause times in Figure 10 and under different node speeds in Figure 11.

These two figures show that our MDSCR algorithm achieves better performance than the SPSC algorithm under all mobility scenarios. In particular, the MDSCR algorithm works best with pause time ranging from 10s to 50s and node speed as 10ms, which represents a medium-mobility environment. Under such a mobility environment, the service link lifetime prediction method gives the best prediction results.

## F. Impact of Number of Component Replicas

The performance of service composition and recovery algorithms intuitively depends on the service component

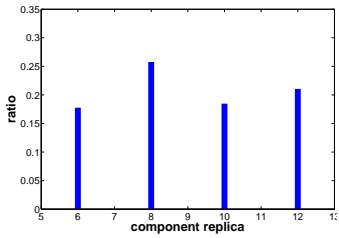


Fig. 12. Impact of Number of Component Replicas on Improvement Ratio

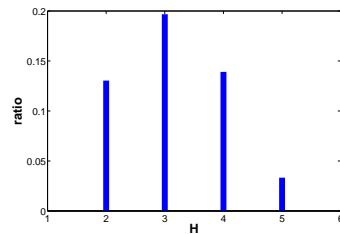


Fig. 13. Impact of  $H$  on improvement ratio

redundancy in the network (*i.e.*, number of component replica). We simulate both algorithms in networks with different numbers of component replica: 6, 8, 10, 12. Figure 12 plots the average improvement ratio of 20 runs over different randomly generated topologies. This figure shows that the improvement ratio from the SPSC algorithm to MDSCR algorithm does not change much, although the performance of each algorithm would change under different network redundancy situations.

#### G. Impact of Service Link Length Requirement $H$

The service link length requirement  $H$  can limit service link selection, and thus may affect the performance of the service composition and recovery algorithms. To study the impact of service link length requirement  $H$ , we ran simulations under different values of  $H$  (2,3,4,5). The simulation results in Figure 13 show that the MDSCR algorithm performs much better than SPSC algorithm for all  $H$  values.

The MDSCR algorithm also works best when the maximum service link length requirement is 3. The reason for this observation is that when the service link length requirement is small, the feasible service link set is also small, which in turn limits the possibility that MDSCR can choose a better service link. Conversely, if the service link length requirement is too large (*e.g.*, 5), then the service link lifetime depends largely on the network topology instead of the relative locations of its two components. The prediction method thus works less effectively due to randomness in the service link lifetime.

## VII. CONCLUDING REMARKS

This paper systematically investigates the service composition and recovery strategies that improve the ability of service delivery in mobile ad hoc networks under constant wireless link failures. It uses an optimization-based approach, develops a theoretical framework for minimum disruption service composition and recovery based on dynamic programming, and presents a MDSCR heuristic algorithm that provides an effective service composition and recovery solution for ad hoc networks.

Our simulation results show that the MDSCR algorithm can achieve much less disruption to end users than traditional methods, such as shortest path routing and service composition.

## REFERENCES

- [1] X. Gu and K. Nahrstedt, "Dynamic QoS-Aware Multimedia Service Configuration in Ubiquitous Computing Environments," in *Proc. of IEEE ICDCS*, 2002.
- [2] D. Xu and K. Nahrstedt, "Finding Service Paths in a Media Service Proxy Network," in *Proc. of SPIE/ACM MMCN*, 2002.
- [3] J. Balasubramanian, P. Lardieri, D. C. Schmidt, G. Thaker, A. Gokhale, and T. Damiano, "A Multi-layered Resource Management Framework for Dynamic Resource Management in Enterprise DRE Systems," *Journal of Systems and Software: special issue on Dynamic Resource Management in Distributed Real-Time Systems*, 2006.
- [4] Y. Liu, A. H.H. Ngu, and L. Zeng, "QoS Computation and Policing in Dynamic Web Service Selection," in *Proc. of the International Conference on WWW*, 2004.
- [5] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality Driven Web Services Composition," in *Proc. of the International Conference on WWW*, 2003.
- [6] B. Raman and R. H. Katz, "An Architecture for Highly Available Wide-Area Service Composition," *Computer Communications Journal*, vol. 26, no. 15, 2003.
- [7] L. Xiao and K. Nahrstedt, "Minimum User-perceived Interference Routing in Service Composition," in *Proc. of IEEE Infocom*, 2006.
- [8] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," in *Proc. of the IEEE Infocom*, 2003.
- [9] J. Tang, G. Xue, and W. Zhang, "Reliable Ad Hoc Routing Based on Mobility Prediction," *Journal of Combinatorial Optimization*, vol. 11, no. 1, 2006.
- [10] M. Mikic-Rakic, S. Malek, and N. Medvidovic, "Improving Availability in Large, Distributed Component-Based Systems via Redeployment," in *Proc. of International Working Conference on Component Deployment*, 2005.
- [11] V. Subramonian, G. Deng, C. Gill, J. Balasubramanian, L. Shen, W. Otte, D. C. Schmidt, A. Gokhale, and N. Wang, "The Design and Performance of Component Middleware for QoS-enabled Deployment and Conguration of DRE Systems," *Journal of Systems and Software: special Issue Component-Based Software Engineering of Trustworthy Embedded Systems*, 2006.
- [12] K. Wang and B. Li, "Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks," in *Proc. of the IEEE Infocom*, 2002.
- [13] U. C. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: An overall perspective on architectural choices and network layer support issues," *Ad Hoc Networks*, vol. 2, no. 1, 2004.
- [14] R. Koodli and C. Perkins, "Service Discovery in On-Demand Ad Hoc Networks," in *Internet Draft*, 2002.
- [15] C. E. Perkins, E. M. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," in *IETF Internet draft*, 2003.
- [16] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 2000.
- [17] S. Jiang, Y. Xue, and D. Schmidt, "Minimum Disruption Service Composition and Recovery in Mobile Ad hoc Networks," in *Vanderbilt University technical report #ISIS-06-711*, 2006.
- [18] Y. Xue, B. Li, and K. Nahrstedt, "A scalable location management scheme in mobile ad-hoc networks," in *Proc. of the IEEE Annual Conference on Local Computer Networks*, 2001.