Minimum Disruption Service Composition and Recovery over Mobile Ad Hoc Networks

Shanshan Jiang, Yuan Xue, Douglas C. Schmidt
ISIS and 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 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 the *disruption index*, which is a novel construct that characterizes different aspects of service disruptions, including frequency and duration.

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 give 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 algorithm approximates the optimal solution with one-step lookahead prediction, where service link lifetime is predicted using linear regression. We evaluate the performance of our algorithm via simulation study conducted under various network environments.

I. Introduction

Mobile wireless networks provide critical infrastructure support for ubiquitous computing that enables computing service accessible at anytime and anywhere. Mobile ad hoc networks (MANETs) are self-organized wireless networks that are dynamically formed through collaboration among mobile nodes. Since MANETs can be formed rapidly without deploying any fixed networking infrastructure, they can support ubiquitous computing effectively.

The diverse application needs in ubiquitous computing environments have fueled an increasing demand for new functionality and services. To meet these demands, component-based software development has been used to ensure the flexibility and maintainability of software systems. For example, project Gaia [1], [2], [3] provides component-based middleware that manages the ubiquitous computing environment called *Active Space*. To provide comprehensive functions for end users, *service composition* [4], [5], [6] integrates loosely-coupled distributed service components into a composite service.

This work was supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: Cisco, ESCHER, HP, IBM, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia and United Technologies.

In light of the needs outlined above, this paper studies the problem of service composition over MANETs. There is an extensive literature on service composition techniques over wired networks [5], [7], [8], [9], [10], which has made critical steps towards constructing high quality service paths in a variety of networking environments. The wired network results cannot be extended directly to service composition in MANETs, however, since they do not consider the intermittent link connectivity and dynamic network topology caused by node mobility.

To address this 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 minimum disruptions to end users in MANETs*. We focus on two important factors of service disruption: its frequency and duration, which characterize the disruption experienced by end users. To achieve this goal, we address the following three challenges:

• *How to characterize and measure the impact of service disruptions quantitatively*. *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 [11], [12] and component deployment mechanisms [13]. The basic idea is to use the path with maximum reliability for data/service delivery. We face two problems when using reliability as a metric for service composition and recovery design in MANETs. 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 the packet loss ratio along a path. Its constantly changing value may cause repeated service adjustments in MANETs, 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 a MANET, 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. These 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 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 MANETs. 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 interdependent 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 MANETs to minimize service disruptions. This 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 the *disruption index*, which is a novel construct that characterizes different service disruption aspects, such as frequency and duration, that can not be captured via conventional metrics, such as reliability and availability.

For MANETs with known mobility plans, 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 MANETs with uncertain node mobility. This heuristic algorithm approximates the optimal solution with one-step lookahead prediction, where the sustainability of a service link is modeled through its lifetime and predicted via an estimation function derived using linear regression.

This paper makes the following contributions on service composition and recovery in MANETs: (1) it creates a theoretical framework for service composition and recovery strategies for MANETs that characterize the effect of service disruption; (2) it presents an optimal solution to MDSCR problem based on dynamic programming techniques and provides important analytical insights for MDSCR heuristic algorithm design; (3) it presents a simple yet effective statistical model based on 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 MANETs. Section IV formulates the MDSCR problem and provides its optimal solution; Section V describes the MDSCR heuristic algorithm; Section VI presents our simulation results; and Section VII concludes the paper.

II. Network and Service Model

We consider a MANET consisting of a set of mobile nodes $\mathcal{N}$. In this network, link connectivity and network topology change with node movement. We model this 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.[1] We further denote a network path that connects node n_1 and n_m in this graph as $\mathcal{P}_{(n_1,n_m)}(t) = (n_1, n_2, ...n_m)$, 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 [14]. All application components are constructed as *autonomous services* that perform independent operations (such as video 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 $\mathcal{S} = (s_1 \rightarrow s_2 \rightarrow ... \rightarrow s_r)$, where $s_k (k = 1, ..., r-1)$ 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*.

In a MANET, each service component s_k can be replicated at multiple nodes to improve the service availability [15]. 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 of 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, $(s_1[a] \rightarrow s_2[b] \rightarrow s_2[c] \rightarrow s_r[r])$ is a service path; the service link $(s_1[a] \rightarrow s_2[b])$ is supported by the network path (l_1, l_2).

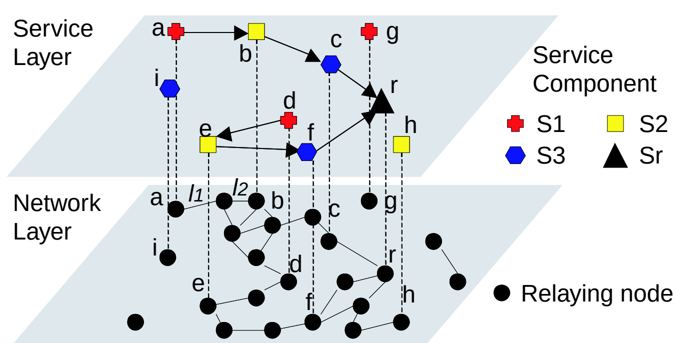

Fig. 1. Example Service Deployment and Service Paths

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

III. Service Composition and Recovery Framework for 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 MANET involves the following two inherently tightly-coupled processes:

[1]For simplicity, we only consider bi-directional wireless links in this work.

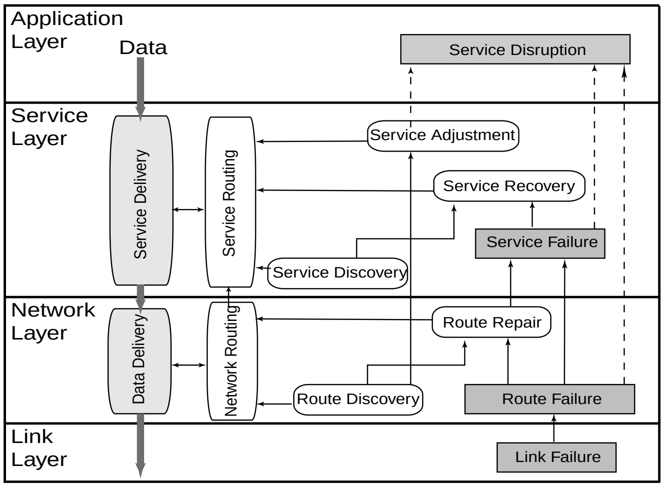

Fig. 2. A Service Composition and Recovery Framework in a Mobile Ad Hoc Network

• *Service routing*, which selects the service components (out of many replicas) for the service path. It relies on service component discovery [16], [17] to find the candidate service components, then selects the appropriate ones to compose a service path. Formally, a service routing scheme is represented as $\pi_{\mathcal{S}} = (s_1[n_1], s_2[n_2], ..., s_r[n_r])$, 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 $\pi_{\mathcal{N}} = \{\mathcal{P}_{(n_k, n_{k+1})}, k = 1, ..., r-1\}$ 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 interact with each other closely. 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 $\pi = (\pi_{\mathcal{S}}, \pi_{\mathcal{N}})$.

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 a MANET, 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, which 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 sometimes 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. Theoretical Framework for Minimum-Disruption Service Composition and Recovery (MDSCR)

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 the end user 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 quantitatively characterizes the impact of service disruption and establishes such an optimization-based theoretical framework based on dynamic programming.

A. Service Disruption Model

A classical way to model service disruption is *service availability*, which is defined as the fraction of service available time during the service lifetime T: $A = \frac{T - \sum_{i=1}^{q}(\bar{t}_i)}{T}$, where q is the number of service disruptions and $\bar{t}_1, \bar{t}_2, ..., \bar{t}_q$ is the sequence of disruption durations. 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*. Specifically, service availability cannot differentiate between one scenario with higher service failure frequency but shorter disruption durations from the other scenario with lower service failure frequency but longer 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 incorporate into service composition and recovery decisions. To establish a theoretical framework that provides realistic insight to implementation of service composition and recovery strategy, therefore, 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 $F(\bar{t})$ (*e.g.*, convex, concave, or linear) reflects its relative sensitivity to disruption duration and frequency. We further define *disruption index* D as a metric that characterizes the impact of service disruption during the entire service lifetime T:

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

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 respectively.

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 MANETs [18] involve discovering an alterative 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.

Using the number of wireless link substitutions as an estimate for disruption duration introduced by network-level recovery is consistent with typical MANET 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.[2]

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.

Based on the recovery duration estimation, we now proceed to refine the definition of the disruption index. Consider a service $\mathcal{S}$ that starts at time instance 0 and ends at T. Let $\pi(t_1), \pi(t_2), ..., \pi(t_l)$ be the sequence of service composition schemes used during the service lifetime, and l be the length of this sequence. The disruption duration $\bar{t}_k$ from service composition $\pi(t_k)$ to $\pi(t_{k+1})$ is estimated as

[2]For simple estimation, we do not consider the impact of route caches here.

$$\begin{aligned} \bar{t}_k &= \beta \times N_{\pi(t_k)\rightarrow\pi(t_{k+1})} && (2)\\ &= \beta \times (N^{\mathcal{N}}_{\pi(t_k)\rightarrow\pi(t_{k+1})} + \alpha N^{\mathcal{S}}_{\pi(t_k)\rightarrow\pi(t_{k+1})}) && (3) \end{aligned}$$

where $N^{\mathcal{N}}_{\pi(t_k)\rightarrow\pi(t_{k+1})}$ and $N^{\mathcal{S}}_{\pi(t_k)\rightarrow\pi(t_{k+1})}$ 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 $\pi(t_{k+1})$ respectively. β is the parameter that converts the number of substitutions to disruption time. $\alpha > 1$, 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

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

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: $\Pi = (\pi(t_1), \pi(t_2), ..., \pi(t_l))$. Note that Π gives initial service composition scheme $\pi(t_1)$ and all the service recovery schemes $\pi(t_k) \rightarrow \pi(t_{k+1})$, $k = 1, ..., l-1$. We denote the set of all feasible service composition policies over MANET $\mathcal{G}$ as $\Phi(\mathcal{G})$. For a feasible service policy $\Pi \in \Phi(\mathcal{G})$, there is a corresponding disruption index $\tilde{D}(\Pi)$. 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,

$$\text{MDSCR}: \quad \textbf{minimize} \quad \tilde{D}(\Pi) \tag{5}$$

$$\Pi \in \Phi(\mathcal{G}) \tag{6}$$

When the mobility plan is determined a priori, the graph series $\mathcal{G}(t)$ is then given. In this case, 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. Thus to derive a practical solution for MDSCR problem, we need to consider heuristics that can dependably predict link lifetime and integrate it into service routing and recovery. In the following sections, we first study the optimal MDSCR solution (Section IV-C) and then present the MDSCR heuristic algorithm (Section V).

C. Optimal Solution

If $\mathcal{G}(t)$ is given, 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.*:

$$\mathcal{J}(\pi(t_w)) = \min_{\Pi\in\Phi(\mathcal{G})} \frac{1}{T}\sum_{k=w}^{l-1} F(\beta \times N_{\pi(t_k)\rightarrow\pi(t_{k+1})}) \tag{7}$$

From Eq. (4) and Eq. (7), based on dynamic programming, we have

$$\mathcal{J}(\pi(t_w)) = \min_{\pi(t_{w+1})} \{\frac{1}{T} F(\beta \times N_{\pi(t_w)\rightarrow\pi(t_{w+1})}) + \mathcal{J}(\pi(t_{w+1}))\} \tag{8}$$

When the mobility plan of the MANET is known, the equation shown above could be used to give the optimal MDSCR solution via standard dynamic programming techniques [19]. 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. (8) gives the optimal service recovery scheme (minimum disruption service recovery) that changes the service composition from $\pi(t_w)$ to $\pi(t_{w+1})$.

V. MDSCR Heuristic Algorithm

A. Two-Tier MDSCR Algorithm

Based on the analytical properties[3] of the optimal MDSCR solution, which are given and proven in [20] due to space constraints, we present the MDSCR heuristic algorithm. We can reduce the complexity of the 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 $\tilde{D}_{\mathcal{S}}$ via service routing, where $\tilde{D}_{\mathcal{S}}$ is defined as

$$\tilde{D}_{\mathcal{S}} = \frac{1}{T}\sum_{k=1}^{g-1} F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_k)\rightarrow\pi_{\mathcal{S}}(t^s_{k+1})}) \tag{9}$$

At time t^s_{w+1} with service routing scheme $\pi_{\mathcal{S}}(t^s_w)$, the service recovery scheme that changes the service path from $\pi_{\mathcal{S}}(t^s_w)$ to $\pi_{\mathcal{S}}(t^s_{w+1})$ is given by solving the following equation:

$$\begin{aligned} &\mathcal{J}(\pi_{\mathcal{S}}(t^s_w)) \qquad (10)\\ = &\min_{\pi_{\mathcal{S}}(t^s_{w+1})}\{\frac{1}{T}F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_w)\rightarrow\pi_{\mathcal{S}}(t^s_{w+1})}) + \mathcal{J}(\pi_{\mathcal{S}}(t^s_{w+1}))\}\end{aligned}$$

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 $\tilde{D}_{\mathcal{N}}$ (defined as follows) during the lifetime of each service link via network routing.

$$\tilde{D}_{\mathcal{N}}(t^s_w \rightarrow t^s_{w+1}) = \frac{1}{T}\sum_{t=t^s_w}^{t^s_{w+1}} F(\beta N^{\mathcal{N}}_{\pi(t)\rightarrow\pi(t+1)}) \tag{11}$$

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

[3]For an optimal solution, a service path is changed if and only if one of the underlying wireless link is broken, which means that the service composition remains the same on the discovery of new service components in the neighborhood and the node failures that are not on the service path; and the service-level recovery is invoked if and only if the network-level recovery can not repair one of the service links in use.

B. One-step Look-ahead Approximation

Finding the solution to the service-level MDSCR problem is still impossible for MANETs with uncertain mobility plan since it needs the complete knowledge of future network topologies. Specifically, the service recovery decision at t^s_{w+1} requires the knowledge of network topology after this time to calculate the future disruption index $\mathcal{J}(\pi_{\mathcal{S}}(t^s_{w+1}))$. 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 $E(N^{\mathcal{S}})$.

Formally, let $L_{n_k\rightarrow n_{k+1}}$ be the expected lifetime[4] for the service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$. The service routing scheme at time t^s_{w+1} is $\pi_{\mathcal{S}}(t^s_{w+1}) = (s_1[n_1], s_2[n_2], ..., s_r[n_r])$. Its failure rate is estimated as $\gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} = \sum_{k=1}^{r-1}\frac{1}{L_{n_k\rightarrow n_{k+1}}}$. Likewise, $\mathcal{J}(\pi_{\mathcal{S}}(t^s_{w+1}))$ is estimated as

$$\hat{\mathcal{J}}(\pi_{\mathcal{S}}(t^s_{w+1})) = F(\beta\alpha \times E[N^{\mathcal{S}}]) \times \gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} \tag{12}$$

The initial service composition strategy is to find $\pi_{\mathcal{S}}(t^s_1)$ to minimize

$$F(\beta\alpha \times E[N^{\mathcal{S}}]) \times \gamma_{\pi_{\mathcal{S}}(t^s_1)} \tag{13}$$

The service-level recovery strategy involves finding a service routing scheme $\pi_{\mathcal{S}}(t^s_{w+1})$ to minimize

$$\frac{1}{T}F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_w)\rightarrow\pi_{\mathcal{S}}(t^s_{w+1})}) + F(\beta\alpha E[N^{\mathcal{S}}])\gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} \tag{14}$$

Eq. (14) 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 for Eq. (13) and Eq. (14) is to estimate the service link lifetime. This problem is non-trivial due to the highly interdependent wireless link failures and the impact from network path repairs. It therefore cannot be solved by traditional network path reliability estimation methods.

To address this challenge, we devise a service link lifetime prediction method based on linear regression. In particular, we estimate the lifetime of a service link $L_{n\rightarrow n'}$ based on the 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 Δt. The lifetime of a service link is computed via linear regression shown as follows.

$$L_{n\rightarrow n'} = K \times \tilde{d}_{n\rightarrow n'}(t+\Delta t) + B \tag{15}$$

where K and B are two coefficients of linear regression.

In the simulation study (Section VI), we derive the corresponding coefficients for linear regression in different network setups, and pick the prediction time Δt with the largest goodness-of-fit.

[4]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.

D. Two-Tier Predictive Heuristic Algorithm

We now summarize the discussions above and present the MDSCR heuristic algorithm. The deployment of our algorithm needs the support of location service [21] for node location and velocity information, and service discovery service [17].

Table I gives the minimum disruption service composition algorithm. This algorithm has two tiers. The top tier is service routing that finds the service components for the service path. After the service components are determined, the network routing algorithm in the bottom tier will find the data path to connect these components.

Table II gives the minimum disruption service recovery algorithm. This algorithm also has two tiers. The bottom 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 returns successfully. If the network-level recovery fails, however, then service-level recovery in the top tier will be triggered.

	Alg. I: Minimum Disruption Service Composition
1	*Top tier: service routing*
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 $\pi_{\mathcal{S}}$ that minimizes Eq. (13). //This could be done based on any minimum cost routing algorithm
2	*Bottom tier: network routing*
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 I

MINIMUM DISRUPTION SERVICE COMPOSITION ALGORITHM

	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	*Bottom tier: network-level recovery*
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	*Top tier: service-level recovery*
	//Assume the current service routing scheme is $\pi_{\mathcal{S}}(t_w^s)$
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 $\pi_{\mathcal{S}}(t_{w+1}^s)$ that minimizes Eq. (14)
	//then perform network routing
2.3	For each service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$ in $\pi_{\mathcal{S}}(t_{w+1}^s)$
	Find the network path with the maximum estimated lifetime and length $\leq H$.
	$\mathcal{P}_{(n_k,n_{k+1})} \leftarrow MLNR(n_k, n_{k+1}, \mathcal{G})$

TABLE II

MINIMUM DISRUPTION SERVICE RECOVERY ALGORITHM

VI. SIMULATION STUDY

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

A. Simulation Setup

In the simulated MANET, 50 nodes are randomly deployed over a $2,000 \times 1,000 m^2$ region. Each node has a transmission range of $250m$. Node mobility follows the random waypoint model with certain *maximum speed* (default value is $10m/s$) and certain *pause time* (default value is $10s$). Each simulation runs for $2,000s$.

The simulated service is composed of 3 components; each component has 8 replicas by default. Each service link requires its maximum network path length $H \leq 3$. In the simulation, the prediction time is adjusted for each network configuration to achieve the smallest prediction error. The service discovery is simulated based on the results presented in [22] and the network routing protocol is simulated using AODV in ns-2. Based on the averaged simulation results, we set the values of α to 10 and β to 1.

We compare the performance of our MDSCR algorithm with the shortest path service composition and recovery (SPSCR) algorithm. Shortest path routing [23] is a common ad hoc routing algorithm that chooses the path with the smallest hop number. A shortest path may fail quickly, however, because some of the wireless links on the shortest path may be broken shortly after the path is established due to node mobility. Here the SPSCR algorithm is a natural extension of shortest path routing algorithm, where the length of a service link is the length of the shortest network path that supports it and the service path with the shortest service link length will be chosen.

For each experiment, we run both MDSCR and SPSCR algorithms 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 compare the throughput of the MDSCR and SPSCR algorithms over the same network mobility configuration. The network parameters used here are based on their default values. The service is sending constant bit rate (CBR) traffic at $1 packet/sec$. The simulation results are shown in Figure 3.

This figure shows that the MDSCR algorithm incurs fewer and shorter disruptions with regard to their frequencies and durations, which leads to a relatively higher and smoother throughput performance.

We then experiment the MDSCR and SPSCR algorithms over 50 different random network topologies. We also use the default values to configure the network parameters. The simulation results are shown in Figure 4. In the figure, the y-axis shows the improvement ratio, which is defined as $\frac{\tilde{D}_{SPSCR} - \tilde{D}_{MDSCR}}{\tilde{D}_{SPSCR}}$, where $\tilde{D}_{MDSCR}$ and $\tilde{D}_{SPSCR}$ are the

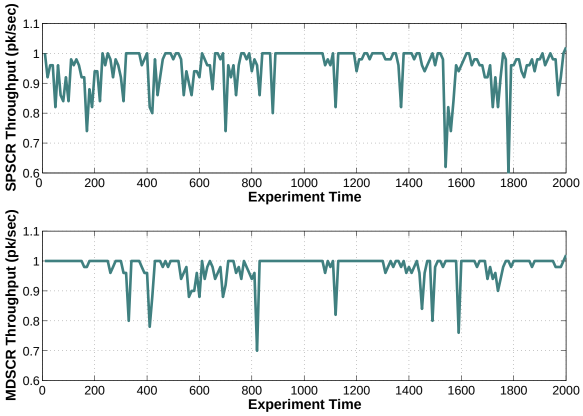

Fig. 3. Throughput comparison with SPSCR and MDSCR

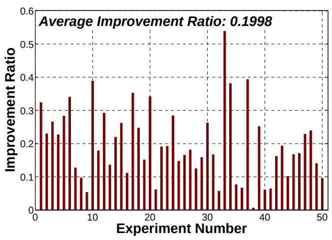

Fig. 4. Improvement ratio under default simulation parameters

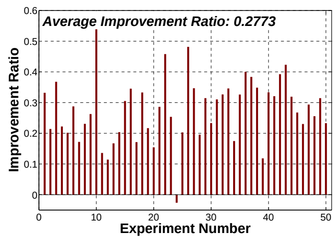

Fig. 5. Improvement ratio with a service path length of 4

disruption indices of the MDSCR and SPSCR algorithms, respectively. Figure 4 shows that the MDSCR algorithm outperforms the SPSCR algorithm in most experiments by an average ratio of 19.98%.

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 plotted in Figure 5. This result shows that the MDSCR algorithm consistently outperforms the SPSCR algorithm under both service path lengths.

Comparing Figure 4 with Figure 5, we also observe that the average improvement ratio 27.73% with longer service path length (4) is better than the one with service path length as 3. This result shows that the benefit of MCSCR algorithm increases relative to SPSCR when the service path length gets longer, *i.e.*, more service components are involved in the service composition. In some very rare cases (only one in our 50 round experiment), the SPSCR 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.

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. Figure 6 compares the improvement ratios under

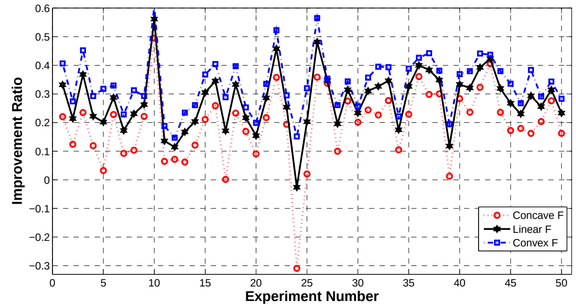

Fig. 6. Improvement ratio comparison with concave, linear, and convex penalty function F

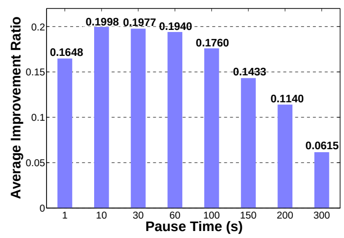

Fig. 7. Impact of pause time on improvement ratio

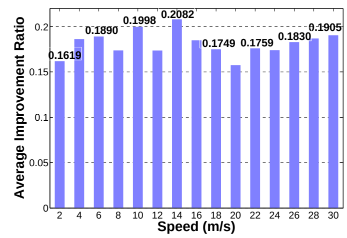

Fig. 8. Impact of node speed on improvement ratio

linear, concave and convex functions when service path length is 4.

This figure shows that the convex function F gives a larger improvement ratio(33.54%) than the linear function(27.73%) and the linear function gives a larger improvement ratio than the concave function(19.20%). 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 SPSCR. 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 SPSCR algorithms under different node speeds and pause times. In particular, we experiment with pause times of $1s$, $10s$, $30s$, $60s$, $100s$, $150s$, $200s$, $300s$ and maximum node speeds of $2m/s$, $4m/s$, $6m/s$, ..., $30m/s$. The prediction time is also adjusted in each experiment to reflect the best prediction results, *i.e.*, the largest goodness-of-fit in linear regression.

Figure 7 and Figure 8 show that our MDSCR algorithm achieves better performance than the SPSCR algorithm under all mobility scenarios. In particular, the MDSCR algorithm works best with pause time ranging from $10s$ to $100s$, which represents a medium-mobility environment. In such mobility environments, the service link lifetime prediction method achieves the best prediction results.

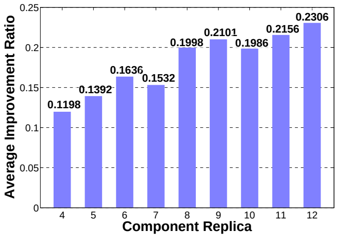

Fig. 9. Impact of number of component replicas on improvement ratio

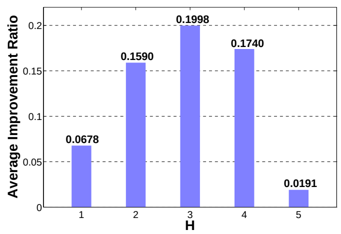

Fig. 10. Impact of service link length requirement H on improvement ratio

F. Impact of Number of Component Replicas

The performance of service composition and recovery algorithms intuitively depends on the service component redundancy in the network, *i.e.*, the number of component replica. We simulate both algorithms with different numbers of component replica: $4, ..., 12$. Figure 9 plots the average improvement ratio over different randomly generated topologies. This figure shows that the improvement ratio from the SPSCR algorithm to MDSCR algorithm grows steadily as the number of component replica increases. This result shows that as the number of optional service paths grows, the opportunity for the MDSCR algorithm to select a better service path increases.

G. Impact of Service Link Length Requirement H

The service link length requirement H can limit the service link selection, and thus may also affect the performance of the service composition and recovery algorithms. To study the impact of service link length requirement H, we run simulations under different values of H $(1, ..., 5)$. These results appear in Figure 10, which shows that the MDSCR algorithm performs better than the SPSCR algorithm for all H values.

The MDSCR algorithm also works best when the maximum service link length requirement is 3. If the service link length requirement is too small (*e.g.*, 1), then for most of the time, there is no optional service path. Conversely, if the service link length requirement is too large (*e.g.*, 5), 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 the randomness in the service link lifetime.

VII. Concluding Remarks

This paper systematically investigates the service composition and recovery strategies that improve the performance of service delivery in MANETs under frequent wireless link failures. It develops a theoretical framework for minimum disruption service composition and recovery based on dynamic programming, and presents a *minimum-disruption service composition and recovery* (MDSCR) heuristic algorithm that provides an effective service composition and recovery solution for MANETs. Our simulation results show that the MDSCR algorithm can provide much less disruption to end users than traditional methods, such as shortest path routing and service composition.

References

[1] M. Roman, C. K. Hess, R. C., A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces," *Journal of IEEE Pervasive Computing*, pp. 74–83, 2002.

[2] S. Chetan, J. Al-Muhtadi, R. Campbell, and M. D. Mickunas, "Mobile Gaia: A Middleware for Ad-hoc Pervasive Computing," in *Proc. of IEEE Consumer Communications and Networking Conference*, 2005.

[3] C. Shankar, J. Al-Muhtadi, R. Campbell, and M. D. Mickunas, "A Middleware for Enabling Personal Ubiquitous Spaces," in *Proc. of System Support for Ubiquitous Computing Workshop at Sixth Annual Conference on Ubiquitous Computing*, 2004.

[4] X. Gu and K. Nahrstedt, "Dynamic QoS-Aware Multimedia Service Configuration in Ubiquitous Computing Environments," in *Proc. of IEEE ICDCS*, 2002.

[5] D. Xu and K. Nahrstedt, "Finding Service Paths in a Media Service Proxy Network," in *Proc. of SPIE/ACM MMCN*, 2002.

[6] 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*, vol. 80, no. 7, 2007.

[7] Y. Liu, A. H.H. Ngu, and L. Zeng, "QoS Computation and Policing in Dynamic Web Service Selection," in *Proc. of International Conference on WWW*, 2004.

[8] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality Driven Web Services Composition," in *Proc. of International Conference on WWW*, 2003.

[9] B. Raman and R. H. Katz, "An Architecture for Highly Available Wide-Area Service Composition," *Computer Communications Journal*, vol. 26, no. 15, 2003.

[10] L. Xiao and K. Nahrstedt, "Minimum User-perceived Interference Routing in Service Composition," in *Proc. of IEEE Infocom*, 2006.

[11] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," in *Proc. of IEEE Infocom*, 2003.

[12] J. Tang, G. Xue, and W. Zhang, "Reliable Ad Hoc Routing Based on Mobility Prediction," *Journal of Combinatorial Optimization*, vol. 11, no. 1, 2006.

[13] 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.

[14] 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 on Component-Based Software Engineering of Trustworthy Embedded Systems*, vol. 80, no. 5, 2007.

[15] K. Wang and B. Li, "Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks," in *Proc. of IEEE Infocom*, 2002.

[16] U. C. Kozat and L. Tassiulas, "Service Discovery in Mobile Ad Hoc Networks: An Overall Perspective on Architectural Choices and Network Layer Support Issues," *Journal of Ad Hoc Networks*, vol. 2, no. 1, 2004.

[17] R. Koodli and C. Perkins, "Service Discovery in On-Demand Ad Hoc Networks," in *Internet Draft*, 2002.

[18] C. E. Perkins, E. M. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," in *IETF Internet draft*, 2003.

[19] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 2000.

[20] 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.

[21] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-hoc Networks," in *Proc. of IEEE Annual Conference on Local Computer Networks*, 2001.

[22] U.C. Kozat and L. Tassiulas, "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks," in *Proc. of IEEE Infocom*, 2003.

[23] Z. Wang and J. Crowcroft, "Analysis of Shortest-path Routing Algorithms in a Dynamic Network Environment," *Journal of ACM SIGCOMM Computer Communication Review*, vol. 22, no. 2, 1992.