Minimum Disruption Service Composition and Recovery in Mobile Ad Hoc Networks

Shanshan Jiang, Yuan Xue, Douglas C. Schmidt

Institute for Software Integrated Systems
Department of Electrical Engineering and Computer Science
Vanderbilt University

Abstract

The dynamic nature of mobile ad hoc networks poses fundamental challenges to the design of service composition schemes that can satisfy the end-to-end quality of service requirements and minimize the effect of service disruptions caused by dynamic link and node failures. Although existing research on mobile ad hoc networks has focused on improving reliability, little existing work has considered service deliveries spanning multiple components. Moreover, service composition strategies proposed for wireline networks (such as the Internet) are poorly suited for highly dynamic wireless ad hoc networks.

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 that support the service path, and *network routing*, which finds the optimal network path that connects these service components. Our framework is based on the *disruption index*, which is a novel concept that characterizes different service disruption aspects, such as frequency and duration, that are not captured adequately by conventional metrics, such as reliability and availability.

Using the definition of disruption index, 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 for ad hoc networks with known mobility plan. Based on the derived analytical insights, we present our MD-SCR 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 based on node location and velocity using linear regression. We use simulations to evaluate the results of our algorithm in various network environments. The results validate that our algorithm can achieve better performance than conventional methods.

Key words: Mobile Ad hoc Network, Service Composition, Reliability

1 Introduction

Mobile ad hoc networks are self-organized wireless networks formed dynamically through collaboration among mobile nodes (1). Since ad hoc networks can be deployed rapidly without the support of a fixed networking infrastructure, they can be applied to a wide range of application scenarios, such as disaster relief and homeland security operations. These diverse application needs have fueled an increasing demand for new functionalities and services. To meet these demands, component-based software development (2) has been used to ensure the flexibility and maintainability of software systems. *Service composition* (3; 4; 5) is a promising technique for integrating loosely-coupled distributed service components into a composite service that provides end users with coordinated functionality, such as web services and multimedia applications.

There is an extensive literature on service composition techniques over wireline networks. For example, (4; 6; 7) focus on finding a service path over wireline networks that satisfies various quality of service (QoS) requirements. Likewise, (8; 9) consider how to provide highly available services. While these results have made critical steps towards constructing high quality service paths in a variety of networking environments, they do not extend directly to service composition in mobile ad hoc networks since intermittent link connectivity and dynamic network topology caused by node mobility is not considered.

To address this open issue, this paper studies service composition over mobile ad hoc networks. In particular, we investigate 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 and incur the minimum disruptions to end users in mobile ad hoc networks*. We focus on two important factors of service disruption—*frequency* and *duration*—that characterize the disruption experienced by end users. 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 common 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

Email address: `shanshan.jiang,yuan.xue,d.schmidt@vanderbilt.edu`
(Shanshan Jiang, Yuan Xue, Douglas C. Schmidt).

algorithms (10; 11) and component deployment mechanisms (12) using the path with maximum reliability for data/service delivery. There are two problems, however, with using reliability as a metric for service composition and recovery design: (1) it does not account for service repair and recovery and (2) 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, 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. 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 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 two 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 repairs and the new nodes emerging in its vicinity.

To address these challenges, this paper presents a new service composition and recovery framework for mobile ad hoc networks to achieve minimum 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. Our framework is based on the *disruption index*. This novel concept characterizes different service disruption aspects, such as frequency and duration, that are captured inadequately by 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 solu-

tion. 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 the sustainability of a service link is modeled through its lifetime and predicted via an estimation function derived using linear regression.

This paper makes three contributions to research on service composition and recovery in mobile ad hoc networks. First, it creates a theoretical framework for service composition and recovery strategies for ad hoc networks that characterize the effect of service disruption. Second, it uses dynamic programming techniques to present the optimal solution to MDSCR problem, which provides important analytical insights for MDSCR heuristic algorithm design. Third, 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 network path repairs.

This paper significantly extends our prior work in (13; 14). In particular, this paper provides a detailed theoretical analysis of the optimal solution of our two-tier MDSCR algorithm. Likewise, we present a comprehensive ns-2 simulation study of disruption indices and the throughput of our MDSCR algorithm and compare it with common algorithms. In addition, (13) is a work-in-progress paper that simply motivates the service disruption issues in dynamic networking environments, whereas this paper provides a detailed analytical and experimental study on mobile ad hoc networks.

The remaining of this paper is organized as follows: Section 2 provides our network and service model; Section 3 describes our service composition and recovery framework for ad hoc networks. Section 4 formulates the MDSCR problem and provides its optimal solution; Section 5 explains our MDSCR heuristic algorithm; Section 6 presents our simulation results and evaluates the performance of our MDSCR algorithm; Section 7 discusses the limitations of our approach in this paper; Section 8 compares MDSCR with related work; and Section 9 presents concluding remarks.

2 System Model

2.1 Mobile Ad Hoc Network Model

We consider a mobile ad hoc network consisting of a set of nodes $\mathcal{N}$. In this network, link connectivity and network topology change with node movement. To model such a dynamic network environment, we first decompose the time horizon $\mathcal{T} = [0, \infty)$ into a set of time instances $\mathcal{T}' = \{\tau_1, \tau_2, ...\}$ so that during

the time interval $[\tau_i, \tau_{i+1})$, the network topology remains unchanged, *i.e.*, the same as the topology at τ_i.

We then model this mobile ad hoc network using a series of graphs indexed by time instances in $\mathcal{T}'$, *i.e.*, $\mathcal{G}_{\mathcal{T}'} = \{\mathcal{G}(\tau), \tau \in \mathcal{T}'\}$. At time τ, the network topology graph is represented by $\mathcal{G}(\tau) = (\mathcal{N}, \mathcal{L}(\tau))\}$, where $\mathcal{L}(\tau)$ represents the set of wireless links at time τ, *i.e.*, for link $l = (n, n') \in \mathcal{L}(\tau)$, nodes n and n' are within the transmission range of each other. [1] We further denote a network path that connects node n_s and n_d in this graph as $\mathcal{P}_{(n_s,n_d)}(\tau) = (n_1, n_2, ...n_m)$, where $(n_j, n_{j+1}) \in \mathcal{L}(\tau)$ for $j = 1, ..., m-1$, and $n_1 = n_s$, $n_m = n_d$. We also use $|\mathcal{P}(\tau)|$ to denote the path length of $\mathcal{P}(\tau)$ (*i.e.*, the number of links in $\mathcal{P}(\tau)$). To simplify the notation, we use $\mathcal{G}, \mathcal{L}, \mathcal{P}$ and omit τ to represent the network topology, link set, and network path at a particular time instance.

Figure 1 shows an example mobile ad hoc network based on the terms defined above. Two snapshots of the network topologies at time instances τ_1 and τ_2
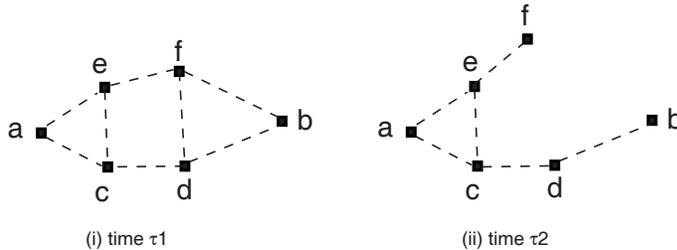

Fig. 1. Example Mobile Ad Hoc Network

are shown in Figure 1(i) and (ii), respectively. Due to the mobility of node f, links (f, d) and (f, b) in $G(\tau_1)$ are no longer available in $G(\tau_2)$.

2.2 Service Model

To characterize the structure of distributed applications that are expected to run in the mobile computing environments, we apply a component-based software model (2). All application components are constructed as *autonomous services* that perform independent operations (such as transformation and filtering) on the data stream passing through them. These services can be connected to form a directed acyclic graph, called a *service graph*.

This paper focuses on so-called *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 $s_k(k = 1, ..., r)$ 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*.

---

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

In a mobile ad hoc network, 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 2 shows an example of service deployment and service composition. A service link is an overlay link that
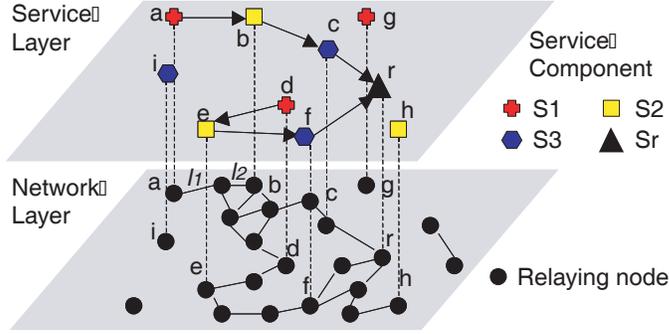

Fig. 2. Example Service Deployment and Service Composition

may consist of several wireless links in the network, *i.e.*, a network path. In Figure 2, $(s_1[a] \rightarrow s_2[b] \rightarrow s_2[c] \rightarrow s_r[r])$ is a service path; the service link $(s_1[a] \rightarrow s_2[b])$ is supported by the network path $\mathcal{P} = (l_1, l_2)$.

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—called the *service link length*—that is defined the number of wireless links traversed by a service link. In particular, we require that the service link length is bounded by H hops. Table 1 summarizes the notations used in this paper.

3 Service Composition and Recovery Framework for Mobile Ad Hoc Network

This section describes our service composition and recovery framework for ad hoc networks.

3.1 Service Composition

Service composition refers to the process of finding a service path that satisfies designated QoS requirements in the network. As shown in Figure 3, service composition in a mobile ad hoc network involves the following two inherently related processes:

- *Service routing*, which selects the service components (out of many replicas)

Notation	Description
$t \in \mathcal{T}$	continuous real time
$\tau \in \mathcal{T}'$	discrete time instance, when topology is changed
$\mathcal{N}$	set of mobile nodes
$\mathcal{G}(\tau)$	network topology graph at time τ
$\mathcal{L}(\tau)$	set of wireless links at time τ
$\mathcal{P} = (n_1, n_2, ... n_m)$	network path
$\mathcal{S} = (s_1 \rightarrow s_2 \rightarrow ... \rightarrow s_r)$	service path
H	service link length requirement
$\pi_\mathcal{S}$	service routing scheme
$\pi_\mathcal{N}$	network routing scheme
$\pi = (\pi_\mathcal{S}, \pi_\mathcal{N})$	service composition and recovery scheme
$\Pi = (\pi(t_1), \pi(t_2), ..., \pi(t_l))$	service composition and recovery policy
$\Phi(\mathcal{G}_{\mathcal{T}'})$	the set of all feasible service composition policies over $\mathcal{G}_{\mathcal{T}'}$
$F(\bar{t})$	disruption penalty function
D	disruption index
$\tilde{D}$	disruption index estimation
$N_{\mathcal{P} \rightarrow \mathcal{P}'}$	number of link substitutions from path $\mathcal{P}$ to path $\mathcal{P}'$
$N_{\pi_\mathcal{S} \rightarrow \pi'_\mathcal{S}}$	number of component substitutions from $\pi_\mathcal{S}$ to $\pi'_\mathcal{S}$
$\mathcal{J}(\pi(t_w))$	minimum disruption index for the service disruption experienced the service from time instance $t_w \in \mathcal{T}$ where composition scheme $\pi(t_w)$ is used
$\tilde{d}_{n \rightarrow n'}(t + \Delta t)$	predicted distance of a service link $(n \rightarrow n')$
$L_{n \rightarrow n'}$	lifetime of service link $(n \rightarrow n')$

Table 1
Key Notations

for the service path. This routing process relies on service component discovery (16; 17) to find the candidate service components, then selects the appropriate ones to compose a service path that satisfies the QoS requirement. 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 hosting nodes for selected service components. Formally, the network routing scheme
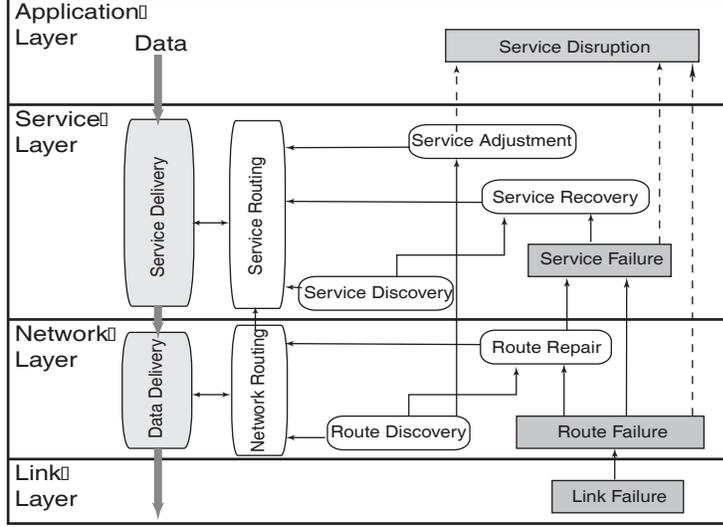
Fig. 3. A Service Composition and Recovery Framework in a Mobile Ad Hoc Network

can be represented as a set of routes $\pi_{\mathcal{N}} = \{\mathcal{P}_{(n_k,n_{k+1})}, k = 1, ..., r - 1\}$ where $\mathcal{P}_{(n_k,n_{k+1})}$ represents the network route that supports the service link $(s_k[n_k] \rightarrow s_{k+1}[n_{k+1}])$.

These two processes interact with each other closely. On one hand, the component selection in the service routing determines the source and destination nodes in the network routing. On the other hand, the path quality in the network routing also affects the selection of service components in the service routing. Collectively, a service composition scheme is represented as $\pi = (\pi_{\mathcal{S}}, \pi_{\mathcal{N}})$.

In an ad hoc network, service failures may occur for multiple reasons. For example, end-to-end QoS requirements of a service may be violated due to network overload; service links may break due to failure of the underlying wireless communication path. This paper focuses on *service failures caused by node mobility*.

3.2 Service Recovery

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, network, or service level. For example, a wireless link failure could be detected at the link-level via IEEE 802.11 ACK frame, or at the network-level through HELLO messages in the routing protocol, such as *AODV* (18).

Similar to service composition, service recovery also involves two processes: (1)

network-level recovery, which repairs the data path between two components, and (2) *service-level recovery*, which replaces one or more service components. The network-level path repair usually depends on the specific ad hoc routing protocol used and relies on the route repair mechanism built within the routing protocol. The 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 (*i.e.*, repaired) path, but also the service path used previously (*i.e.*, 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. For example, network-level recovery may be attempted first without changing any service components. If this recovery fails, then a service-level recovery is initiated. The limitation with using this service recovery strategy, however, is that the new service path may have a poor QoS and/or may fail again soon. Alternatively, we may wish to use service-level recovery directly without trying network-level recovery. Such a strategy, however, will incur more overhead in repairing the failed service links.

Though node mobility can cause service failures, it may provide better service paths by bringing new service components into their vicinity, *i.e.*, within their transmission range. *Service adjustment* is the process of modifying the current service path for better QoS or higher reliability by using a new network path or new component(s) that appear in the vicinity through node mobility. Similar to the dilemma faced by service recovery, however, such changes can disrupt the service, even though they improve the new path's reliability and quality.

4   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 service composition, adjustment, and recovery strategies to achieve minimum service disruptions. This section quantitatively characterizes the impact of service disruption and establishes an optimization-based theoretical framework based on dynamic programming.
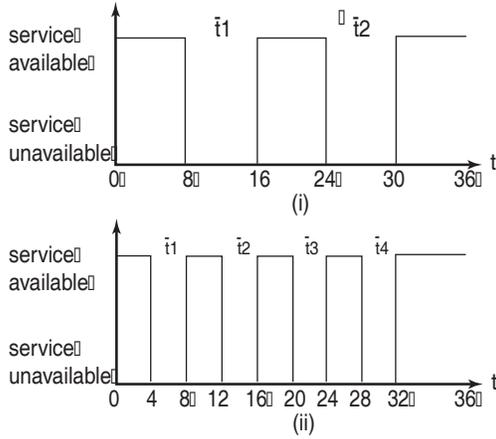
Fig. 4. Example Service Disruption Processes
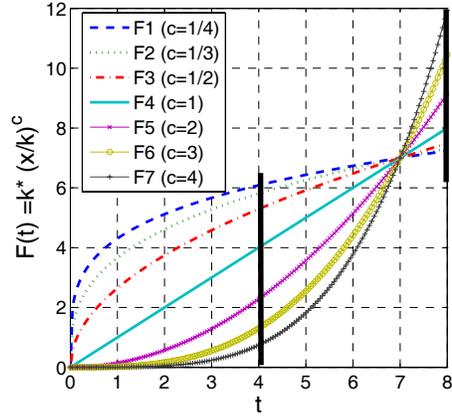
Fig. 5. Example Disruption Penalty Functions (k=7 is the intersection point of all the lines)

4.1 Service Disruption Model

During the service failure and recovery processes, the service is unavailable to the end user, thereby causing service disruption. To analytically investigate service composition and recovery strategies that could provide the most smooth and reliable service delivery, we first need to characterize the impact of service disruption quantitatively.

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, i.e.*, it 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. Figure 4 shows an example of two service disruption processes. In this figure, scenario (i) and (ii) have the same service availability ($\frac{24}{36}$). User-perceived disruption could be different, however, since scenario (ii) has a higher service failure frequency but smaller disruption durations. To model the effect of service disruption precisely, 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

10

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 thus 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 problem of measuring 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 5 shows three basic types of failure penalty functions (*i.e.*, convex, linear, concave). We further define *disruption index D* as a metric to characterize 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 show how the disruption index D characterizes different user-specific disruption effects by choice of $F(\bar{t})$, we calculate the disruption indices for the two service disruption processes in Figure 4 using the different failure penalty functions $F(\bar{t})$ shown in Figure 5. The results are summarized in Table 2.

Fi	Fi(4)	Fi(8)	$D_{Proc(i)}$	$D_{Proc(ii)}$
Fi	Fi(4)	Fi(8)	$D_{Proc(i)}$	$D_{Proc(ii)}$
F1 (convex)	6.0861	7.2376	0.4021	0.6762
F2 (convex)	5.8088	7.3186	0.4066	0.6454
F3 (convex)	5.2915	7.4833	0.4157	0.5879
F4 (linear)	4.0000	8.0000	0.4444	0.4444
F5 (concave)	2.2857	9.1429	0.5079	0.2540
F6 (concave)	1.3061	10.4490	0.5805	0.1451
F7 (concave)	0.7464	11.9417	0.6634	0.0829

Table 2
Disruption Indices Under Different Penalty Functions

Table 2 shows that if $F(\bar{t})$ is a convex 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 concave 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 of computing service availability, we present simple and stable estimations of disruption durations for network-level recovery and service-level recovery, respectively.

4.1.1 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. A typical network-level recovery process in repairing a network path in ad hoc networks (18) involves 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. Formally, let $\mathcal{P}$ and $\mathcal{P}'$ be the paths before and after recovery. We use $N_{\mathcal{P} \to \mathcal{P}'}$ 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

$$N_{\mathcal{P} \to \mathcal{P}'} = |\mathcal{P}'| - |\mathcal{P} \cap \mathcal{P}'| \tag{2}$$

Using the number of wireless link substitutions as an estimate for disruption duration introduced by network-level recovery is consistent with typical network repair operations. For example, there are usually two repair mechanisms in wireless 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]

4.1.2 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 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

---

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

service routing schemes before and after recovery. We use $N_{\pi_\mathcal{S} \to \pi'_\mathcal{S}}$ to represent the number of component substitutions from $\pi_\mathcal{S}$ to $\pi'_\mathcal{S}$, then

$$N_{\pi_\mathcal{S} \to \pi'_\mathcal{S}} = |\pi'_\mathcal{S}| - |\pi_\mathcal{S} \cap \pi'_\mathcal{S}| \tag{3}$$

where $|\pi'_\mathcal{S}| = r$ is the number of components in $\pi'_\mathcal{S}$ and $|\pi_\mathcal{S} \cap \pi'_\mathcal{S}|$ is number of common nodes in these two sets.

Based on the recovery duration estimation, we now proceed to refine the definition of 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 $\pi(t_v)$ to $\pi(t_{v+1})$ is estimated as

$$\bar{t}_k = \beta \times N_{\pi(t_v) \to \pi(t_{v+1})} \tag{4}$$
$$= \beta \times (N^\mathcal{N}_{\pi(t_v) \to \pi(t_{v+1})} + \alpha N^\mathcal{S}_{\pi(t_v) \to \pi(t_{v+1})}) \tag{5}$$

where $N^\mathcal{N}_{\pi(t_v) \to \pi(t_{v+1})}$ and $N^\mathcal{S}_{\pi(t_v) \to \pi(t_{v+1})}$ 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 $\pi(t_v)$ to $\pi(t_{v+1})$ 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 via the component and wireless link substitutions. We denote the estimation of disruption index as $\tilde{D}$:

$$\tilde{D} = \frac{1}{T} \sum_{v=1}^{l-1} F(\beta \times N_{\pi(t_v) \to \pi(t_{v+1})}) \tag{6}$$

$$= \frac{1}{T} \sum_{v=1}^{l-1} F(\beta \times (N^\mathcal{N}_{\pi(t_v) \to \pi(t_{v+1})} + \alpha N^\mathcal{S}_{\pi(t_v) \to \pi(t_{v+1})})) \tag{7}$$

4.2 MDSCR Problem Formulation

Based on the definition of disruption index, 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:

$$\Pi = (\pi(t_1), \pi(t_2), ..., \pi(t_l)) \tag{8}$$

where $0 = t_1 < t_2 < ... < t_l \leq T \in \mathcal{T}$. Π gives the initial service composition scheme $\pi(t_1)$ and all the service recovery schemes $\pi(t_v) \rightarrow \pi(t_{v+1})$, $v = 1, ..., l-1$.

We say service composition $\pi(t_v)$ *is feasible* on network $\mathcal{G}(t_v)$, if and only if all the network paths in $\pi_{\mathcal{N}}(t_v)$ exist on $\mathcal{G}(t_v)$. Moreover, Π *is feasible* if and only if each of its service composition $\pi(t_v)$ is feasible over the network topologies during its lifetime $[t_v, t_{v+1})$, *i.e.*, $\pi(t_v)$ is feasible on all $\mathcal{G}(\tau)$ where $t_v \leq \tau < t_{v+1}$, $\tau \in \mathcal{T}'$.

We denote the set of all feasible service composition policies over $\mathcal{G}_{\mathcal{T}'}$ as $\Phi(\mathcal{G}_{\mathcal{T}'})$. For a feasible service policy $\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'})$, there is a corresponding disruption index, which is defined in Section 4.1 as $\tilde{D}(\Pi)$:

$$\tilde{D}(\Pi) = \frac{1}{T}\sum_{v=1}^{l-1} F(\beta \times N_{\pi(t_v) \rightarrow \pi(t_{v+1})}) \tag{9}$$

The goal of the MDSCR algorithm is to find the best policy $\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'})$ that is feasible for $\mathcal{G}_{\mathcal{T}'}$, so that $\tilde{D}(\Pi)$ is minimized over the lifetime of service $\mathcal{S}$. Formally,

$$\text{MDSCR} : \textbf{minimize } \tilde{D}(\Pi) \tag{10}$$
$$\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'}) \tag{11}$$

At this point, we have established a theoretical framework for the MDSCR problem in mobile ad hoc networks. When the mobility plan is determined *a priori*, the graph series $\mathcal{G}(t)$ is then given. In this case, the MDSCR problem 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 the MDSCR problem, we must therefore consider heuristics that can dependably predict link lifetime and integrate it into service routing and recovery. We next study the optimal MDSCR solution under a known mobility plan (Section 4.3) and derive its analytical properties (Section 4.4). Based on these analytical insights, we then present the location-aided MDSCR heuristic algorithm based on service link lifetime prediction in Section 5.

4.3   Optimal Solution

If $\mathcal{G}_{\mathcal{T}'}$ is given, MDSCR is essentially a dynamic programming problem. Let $\mathcal{J}(\pi(t_w))$ be the minimum disruption index for the service disruptions expe-

rienced by the end user from time instance $t_w \in \mathcal{T}$ where composition scheme $\pi(t_w)$ is used, $i.e.,$

$$\mathcal{J}(\pi(t_w)) = \min_{\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'})} \frac{1}{T} \sum_{v=w}^{l-1} F(\beta \times N_{\pi(t_v) \to \pi(t_{v+1})}) \qquad (12)$$

Obviously $\mathcal{J}(\pi(t_1)) = \min_{\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'})} \tilde{D}(\Pi)$. Based on dynamic programming, we have

$$\mathcal{J}(\pi(t_w)) = \min_{\pi(t_{w+1})} \{\frac{1}{T} F(\beta \times N_{\pi(t_w) \to \pi(t_{w+1})}) + \mathcal{J}(\pi(t_{w+1}))\} \qquad (13)$$

When the mobility plan of the ad hoc network is known, the equation shown above could be used to give the optimal 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. (13) gives the optimal service recovery scheme (minimum disruption service recovery) that changes the service composition from $\pi(t_w)$ to $\pi(t_{w+1})$.

4.4 Analysis

The optimal solution outlined above reveals several interesting properties for MDSCR strategies, as we discuss below.

4.4.1 Reactive Recovery

The first property of an optimal solution is the *reactive adjustment and recovery strategy*. 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 the service composition remains the same on the discovery of new nodes and new service components in the neighborhood (*i.e.*, no service adjustment) and the node failures that are not on the service path. Formally, this property is presented in Theorem 1 below.

Theorem 1: Let $\Pi^* = (\pi^*(t_1), ..., \pi^*(t_l))$ be the optimal MDSCR policy. 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 $\mathcal{G}(\tau_i)$ ($\tau_i \leq t_{w+1} < \tau_{i+1}$, $\tau_i, \tau_{i+1} \in \mathcal{T}'$) at t_{w+1}.

The proof of this theorem is given in the Appendix A.

4.4.2 Reactive service-level recovery

For an optimal solution, the service-level recovery is invoked if and only if the network-level recovery can not repair one of the service links in use, *i.e.*, there is no feasible network path connecting these two service components. This property is formally summarized in Theorem 2 below.

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

The proof of this theorem is given in the Appendix B.

5 MDSCR Heuristic Algorithm

This section explains our MDSCR heuristic algorithm. The analytical results establish several important guidelines for our MDSCR heuristic algorithm. First, a recovery operation will only be triggered upon the failure detection of the wireless link in use. Second, network-level recovery should first be initiated before a service-level recovery is attempted.

5.1 Two-tier MDSCR Algorithm

Based on the analytical results, 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 $\tilde{D}_{\mathcal{S}}$ via service routing, where $\tilde{D}_{\mathcal{S}}$ is defined as

$$\tilde{D}_{\mathcal{S}} = \frac{1}{T} \sum_{v=1}^{g-1} F(\beta \alpha N_{\pi_{\mathcal{S}}(t_v^s) \to \pi_{\mathcal{S}}(t_{v+1}^s)}^{\mathcal{S}}) \tag{14}$$

In particular, the initial service composition solution at the service level is given by solving the following equation:

$$\mathcal{J}(\pi_{\mathcal{S}}(t_1^s)) = \min_{\Pi_{\mathcal{S}} \in \Phi(\mathcal{G}_{\mathcal{T}'})} \frac{1}{T} \sum_{v=1}^{g-1} F(\beta \alpha N_{\pi_{\mathcal{S}}(t_v^s) \to \pi_{\mathcal{S}}(t_{v+1}^s)}^{\mathcal{S}}) \tag{15}$$

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

$$\mathcal{J}(\pi_{\mathcal{S}}(t_w^s)) = \min_{\pi_{\mathcal{S}}(t_{w+1}^s)} \{\frac{1}{T} F(\beta \alpha N_{\pi_{\mathcal{S}}(t_w^s) \to \pi_{\mathcal{S}}(t_{w+1}^s)}^{\mathcal{S}}) + \mathcal{J}(\pi_{\mathcal{S}}(t_{w+1}^s))\} \tag{16}$$

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.

$$\tilde{D}_{\mathcal{N}}(t_w^s \to t_{w+1}^s) = \frac{1}{T} \sum_{t=t_w^s}^{t_{w+1}^s} F(\beta N_{\pi(t) \to \pi(t+1)}^{\mathcal{N}}) \tag{17}$$

The decomposition mechanism presented above separates MDSCR concerns so that the service-level MDSCR and the network-level MDSCR can be treated separately. We focus our discussion below on the service-level MDSCR strategies and rely partially on the existing ad hoc network routing protocols for the network-level MDSCR.

5.2 One-step Lookahead Approximation

Finding the solution to the service-level MDSCR problem is still a challenging issue for ad hoc networks with uncertain mobility plans since complete knowledge of future network topologies is needed. The service recovery decision at t_{w+1}^s requires the knowledge of network topology after this time instance to calculate the future disruption index $\mathcal{J}(\pi_{\mathcal{S}}(t_{w+1}^s))$. To address this problem, we present a one-step look-ahead approximation method where the 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 $L_{n_k \to n_{k+1}}$ be the expected lifetime [3] for the service link $(s_k[n_k] \to$

---

[3] 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

$s_{k+1}[n_{k+1}]$). 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 $\gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} = \sum_{k=1}^{r-1} \frac{1}{L_{n_k \to n_{k+1}}}$. Likewise, $\mathcal{J}(\pi_{\mathcal{S}}(t^s_{w+1}))$ is estimated as

$$\hat{\mathcal{J}}(\pi_{\mathcal{S}}(t^s_{w+1})) = F(\beta\alpha \times E[N^{\mathcal{S}}]) \times \gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} \tag{18}$$

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

$$F(\beta\alpha \times E[N^{\mathcal{S}}]) \times \gamma_{\pi_{\mathcal{S}}(t^s_1)} \tag{19}$$

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

$$\frac{1}{T}F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_w)\to\pi_{\mathcal{S}}(t^s_{w+1})}) + F(\beta\alpha E[N^{\mathcal{S}}])\gamma_{\pi_{\mathcal{S}}(t^s_{w+1})} \tag{20}$$

In Eq. (20), the first term characterizes the recovery duration from the failed service routing scheme $\pi_{\mathcal{S}}(t^s_w)$ to the new service routing scheme $\pi_{\mathcal{S}}(t^s_{w+1})$. The second term characterizes the sustainability of the newly composed service path. Thus minimizing Eq. (20) balances the tradeoff between these two factors faced by service recovery.

5.3 Lifetime Prediction

Another problem with deriving a practical MDSCR solution for Eq. (19) and Eq. (20) involves estimating the service link lifetime. This problem is hard 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.

To address this challenge, we devise a service link lifetime prediction method based on linear regression. [4] In particular, we estimate the lifetime of a network path $L_{n \to n'}$ based on the predicted distance between two components $\tilde{d}_{n \to n'}(t + \Delta t)$, which is calculated based on the current locations of the hosting nodes, their velocities and the prediction time Δt. For a service link $(n \to n')$, let $d_{n \to n'}(t)$ be the distance between its two end nodes, and vector $V_n(t)$, $V_{n'}(t)$

---

that supports this service link is larger than service link length requirement H.
[4] We assume that the mobile nodes in the network are distributed roughly homogeneously.

be their velocities at time t. The predicted distance of service link $(n \to n')$ after time interval Δt is then given as follows:

$$\tilde{d}_{n \to n'}(t + \Delta t) = d_{n \to n'}(t) + \Delta t \times |V_n(t) - V_{n'}(t)| \tag{21}$$

To establish a relation between the predicted distance $\tilde{d}_{n \to n'}(t + \Delta t)$ and the lifetime $L_{n \to n'}$ of a service link $(n \to n')$, we conducted the experiments described below. The network configuration parameters are given in Table 5 in Section 6.1. We plot the relation between the service link lifetime and its predicted distance in Figure 6.
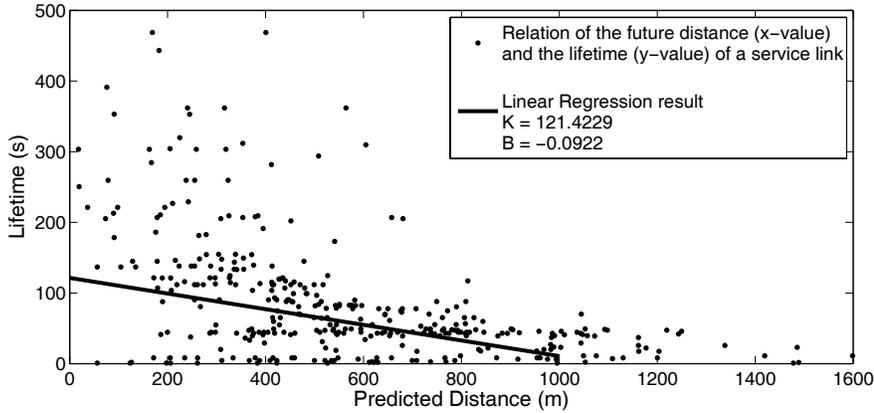
Fig. 6. Lifetime Prediction

The black dot in Figure 6 describes the relation of the predicted distance (x-value) and the lifetime (y-value) of a service link; and the black line is the linear regression result. Using linear regression over the experiment results, the lifetime of a service link is calculated as follows:

$$L_{n \to n'} = K \times \tilde{d}_{n \to n'}(t + \Delta t) + B \tag{22}$$

where $K = 121.4229$ and $B = -0.0922$ are two coefficients of the linear regression in this experiment.

In the simulation study (Section 6), we derive the corresponding coefficients for linear regression for different network configurations, and pick the best prediction time Δt with the largest goodness-of-fit.

5.4   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

Algorithm 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. (19). //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 3
Minimum Disruption Service Composition Algorithm

services (20) for node location and velocity information, as well as service discovery services (17).

Table 3 presents the minimum disruption service composition algorithm. This algorithm has two tiers: top and bottom. The top tier is the service routing that finds the service components with the lowest service link failure rates for the service path. After the service components are determined, the network routing algorithm in the bottom tier will find the network path with the maximum estimated lifetime to connect these components.

Table 4 gives the minimum disruption service recovery algorithm. This algorithm also has two tiers: bottom and top. 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 it fails, however, then the service-level recovery in the top tier will be triggered. The service-level recovery first finds the new service components, which balances the recovery duration and the sustainability for the new service link. It then performs the network path routing between the new service components.

Algorithm 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. (20)

     //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 4

Minimum Disruption Service Recovery

6   Simulation Study

This section evaluates the performance of our MDSCR algorithm via simulation and compares it with other service composition and recovery algorithms.

6.1   Simulation Setup

We conducted the simulations using *ns*-2 (21). In our 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 with a *maximum speed* (default value is $10m/s$) and a *pause time* (default value is $10s$).

number of nodes	50
network size (m^2)	2000×1000
transmission range (m)	250
maximum speed (m/s)	10
pause time (s)	10
number of components in a service path	4
number of component replica $|\mathcal{N}_k|$	8
service link length requirement H	3
α	10
β	1
disruption penalty function	$F(\bar{t}) = \bar{t}$

Table 5
Default Simulation Parameters

The service discovery is simulated based on the results presented in (22) and the network routing protocol is simulated using $AODV$ in ns-2. By default, The service delivers constant bit rate (CBR) traffic at $1 packet/sec$, and the size of the packets is 512 bytes. The simulated service is composed of 4 components and each component has 8 replicas by default. Each service link requires its maximum network path length $H \leq 3$ by default.

Based on the averaged simulation results, we set the values of α to 10 and β to 1. Linear function $F(\bar{t}) = \bar{t}$ is used as the default disruption penalty function. In the simulation, the prediction time is adjusted for each network configuration to achieve the smallest prediction error. Default values of the simulation parameters are given in Table 5.

We compare the performance of our MDSCR algorithm with the *shortest path service composition and recovery* (SPSCR) algorithm (23; 24) and the *random selection service composition and recovery* (RSSCR) algorithm. The shortest path routing algorithm (25) is a common ad hoc routing algorithm that chooses the path with the smallest hop number. The SPSCR algorithm is a natural extension of the 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. The RSSCR algorithm randomly chooses the candidate hosting nodes for the service components in a service path. We use RSSCR as the baseline for comparison since it does not use any optimization strategy.
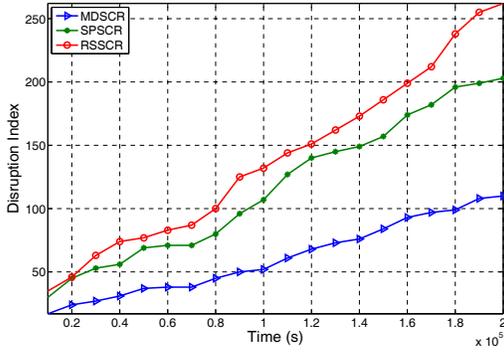
Fig. 7. Disruption Index for MDSCR, SP-SCR, and RSSCR When Service Path Length is 2 Using CBR Traffic
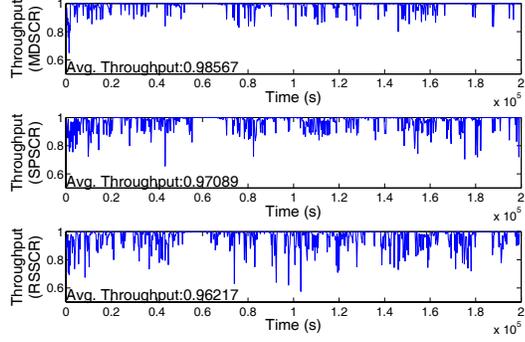
Fig. 8. Throughput for MDSCR, SPSCR, and RSSCR when Service Path Length is 2 Using CBR Traffic

6.2   Basic Comparison

We first conduct the basic comparison of disruption index and throughput for the MDSCR, SPSCR, and RSSCR algorithms. In this experiment, the number of components in a service path is 2. The service link length requirement is restricted by the default network path length requirement in $AODV$, which is 30 hops.

For each experiment, we run the MDSCR, SPSCR, and RSSCR algorithms over the same network scenario, *i.e.*, each node in two runs of the simulation follows the same trajectory. Each CBR traffic simulation runs for $2 \times 10^5 s$. Since the experiment time is extremely long, it can reflect a general network topology.

Figures 7 and 8 show the results of disruption index and throughput for the MDSCR, SPSCR, and RSSCR algorithms using CBR traffic. From Figure 7, we can see that the disruption index is an accumulated value, which increases with time. This figure also shows that the MDSCR algorithm achieves a smaller disruption index compared with the SPSCR and RSSCR algorithms, and thus incurs fewer and shorter disruptions with regard to their frequencies and durations. This result can also be reflected by the instantaneous throughput of the service, which is shown in Figure 8. This figure shows how the MDSCR algorithm achieves higher and smoother throughput in comparison with the SPSCR and RSSCR algorithms.

The reason for these results is that the shortest path may fail quickly for the SPSCR algorithm since some wireless links on the shortest path may be broken shortly after the path is established due to node mobility. Likewise, the RSSCR algorithm performs poorly since it considers neither the length of a service link (as does the SPSCR algorithm) nor the future distance between

23
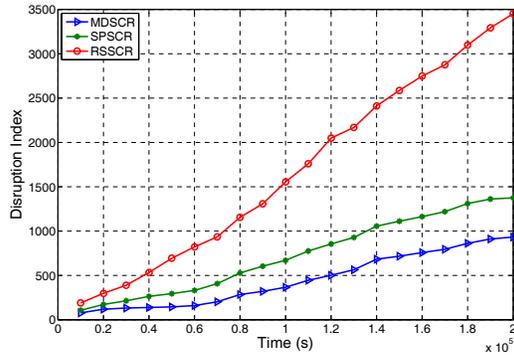
Fig. 9. Disruption Index for MDSCR, SP-SCR, and RSSCR When Service Path Length is 4 Using CBR Traffic
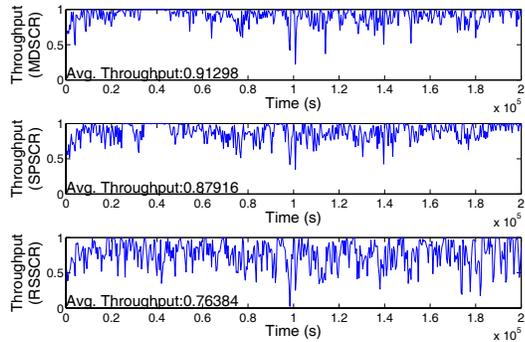
Fig. 10. Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 Using CBR Traffic

service components (as does the MDSCR algorithm).

6.3 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 number of service components from 2 to 4. Figures 9 and 10 show these results.

Comparing Figure 9 with Figure 7, it is clear that the MDSCR algorithm consistently outperforms the SPSCR and RSSCR algorithms under both service path lengths. The throughput comparison in Figures 10 and 8 further validates this result. We also observe that the disruption index increases and the throughput decreases when the synthetic service is composed of more components (*i.e.*, from 2 to 4), which means there is a higher possibility for the service path to be disrupted.

## 6.4 Impact of Service Link Length Requirement H

The service link length requirement H can limit service link selection, and thus may also affect the performance of the service composition and recovery algorithms. Figure 11 shows the results for the service consisting of 2 components with the service link length requirement as 3 hops, using CBR traffic.

Comparing it with Figure 7, we can see that the disruption index increases with more restricted service link length requirement, which means there is a higher possibility for a disconnected service link. The throughput comparison

24
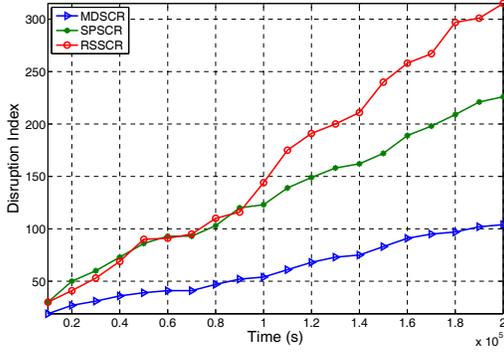
Fig. 11. Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using CBR Traffic
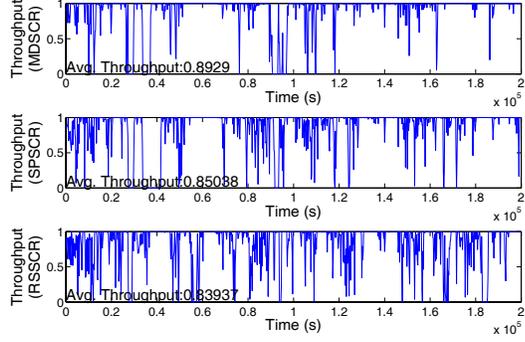
Fig. 12. Throughput for MDSCR, SPSCR, and RSSCR when Service Path Length is 2 and Service Link Length Requirement is 3 Using CBR Traffic
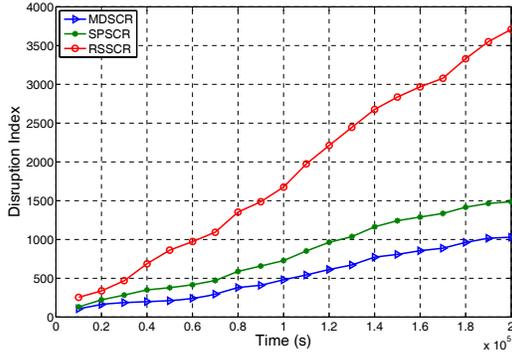
Fig. 13. Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 and Service Link Length Requirement is 3 Using CBR Traffic
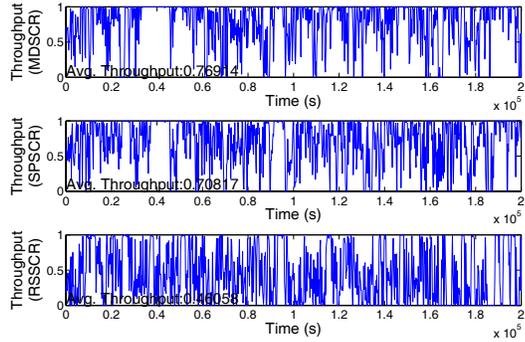
Fig. 14. Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 and Service Link Length Requirement is 3 Using CBR Traffic

in Figures 12 and 8 also verifies this result, *i.e.*, the service throughput is higher and smoother when the service link has no length requirement.

We next conducted experiments with the service consisting of 4 components (service link length requirement remains the same), also using CBR traffic. The results are shown in Figures 13 and 14. By comparing these two figures with Figures 9 and 10, we observe that the disruption index increases and throughput decreases with more restricted service link length requirement.

To further study the impact of service link length requirement H, we introduced the disruption 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 disruption indices of the MDSCR and SPSCR algorithms. We experimented with the MDSCR and SPSCR algorithms over 50 different random network topologies, each of which runs for $2,000s$. We used the average improvement ratio as a metric in our simulation study.
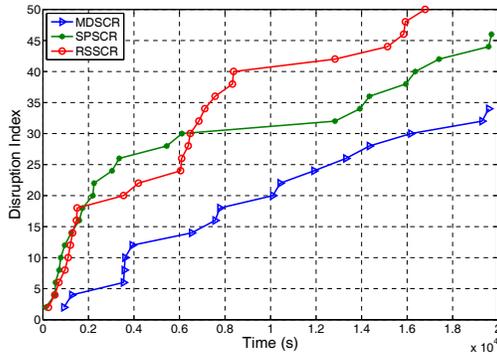
Fig. 15. Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 Using TCP Traffic
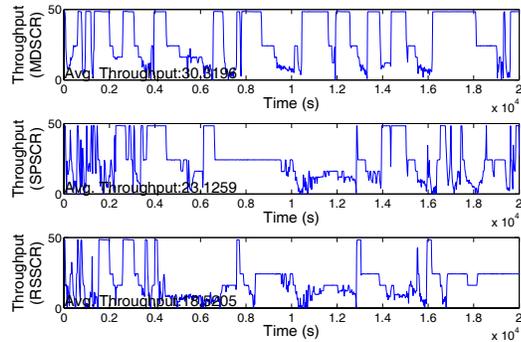
Fig. 16. Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 Using TCP Traffic

We run simulations under different values of H $(1, ..., 5)$ and plot the average improvement ratios in Figure 19. The results show that the MDSCR algorithm outperforms 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 there is no optional service path for most of the time. 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 randomness in the service link lifetime.

6.5 Impact of Traffic Type

The performance of service composition and recovery algorithms heavily depends on the inter-component traffic type, particularly if we consider the throughput of the service in a highly dynamic and lossy network environment. In our simulation study, we use CBR traffic as the default traffic type. Without any loss-based rate adaptation, its throughput directly reflects the impact of service disruption caused by node mobility and link failures. In practice, TCP is also commonly used as a transport protocol for inter-component communication. Here we study the performance of our algorithm over TCP. In our simulation, the packet size is 2 kilobytes. Each simulation runs for $2 \times 10^4 s$. Figures 15 and 16 show the results of disruption index and throughput when service path length is 2 with no service link length requirement. These two figures further validate the results discussed in Section 6.2. From the figures, we could also observe that TCP is more sensitive to the disruptions. This is because its sending rate adapts based on its packet loss/delay and it can not distinguish the queueing loss from the packet loss caused by link failures, which is a common problem of TCP over wireless networks (26).

26
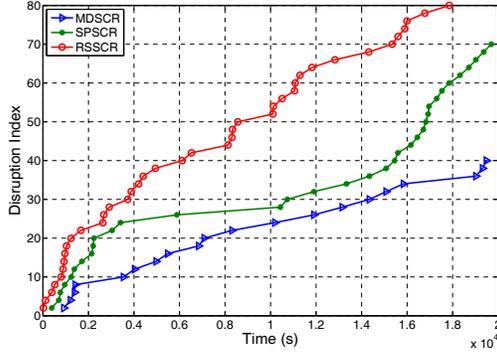
Fig. 17. Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using TCP Traffic
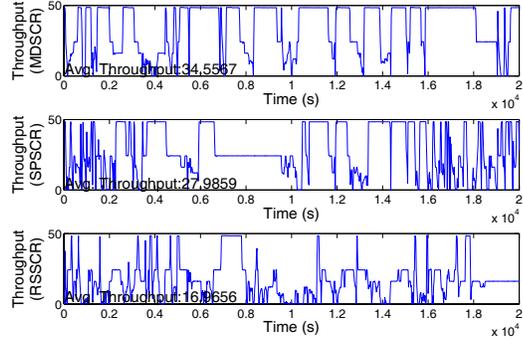
Fig. 18. Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using TCP Traffic

We also conduct experiments with the service consisting of 2 components and service link length requirement is 3, also using TCP traffic. The results are shown in Figures 17 and 18. Comparing these two figures with Figures 15 and 16 shows the same result with regard to the disruption index discussed in Section 6.4, *i.e.*, the disruption index increases with more restricted service link length requirement. Compared with the result discussed in Section 6.4, however, the result of the throughput comparison is opposite. In particular, the service throughput is higher and smoother when the service link length requirement is 3 because the throughput of TCP traffic is also affected by the packet transmission latency, which will decrease with small service link length requirement.

6.6 Impact of Number of Component Replicas

The performance of service composition and recovery algorithms depends intuitively on the service component redundancy in the network (*i.e.*, the number of component replica). We simulate the MDSCR and SPSCR algorithms in networks with different numbers of component replica: $4, ..., 12$, and plot the average improvement ratio of 50 different random network topologies running for $2,000s$ in Figure 20.

Figure 20 shows that the improvement ratio grows steadily as the number of component replica increases. This result indicates that as the number of optional service paths grows, the opportunity for the MDSCR algorithm to select a better service path also increases.

27
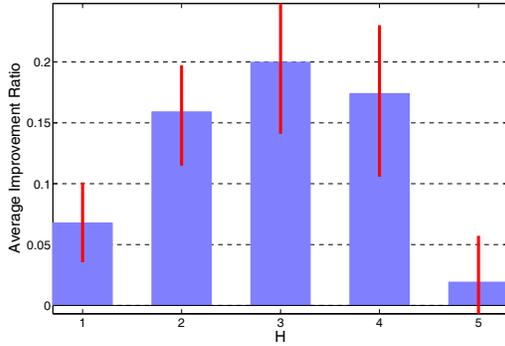
Fig. 19. Impact of Service Link Length Requirement H on Improvement Ratio
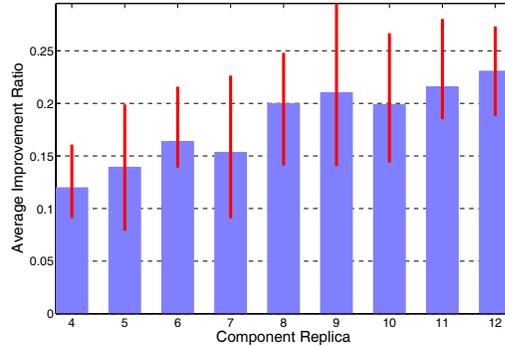
Fig. 20. Impact of Number of Component Replicas on Improvement Ratio
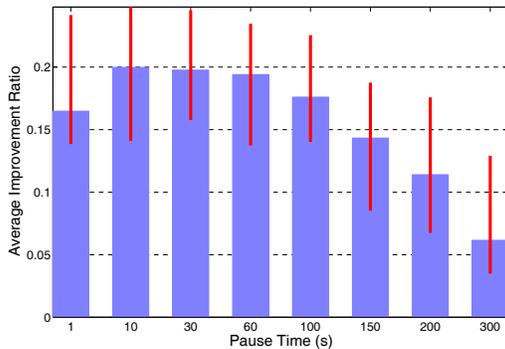
Fig. 21. Impact of Pause Time on Improvement Ratio
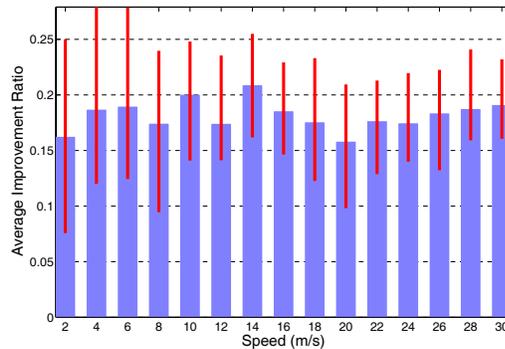
Fig. 22. Impact of Node Speed on Improvement Ratio

6.7 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 mobility configuration to reflect the best prediction results (*i.e.*, the largest goodness-of-fit in linear regression). Each experiment runs over 50 different random network topologies for $2,000s$.

Figures 21 and 22 show that our MDSCR algorithm achieves better performance than the SPSCR algorithm under all mobility scenarios. In particular, our MDSCR algorithm works best with pause time ranging from $10s$ to $100s$, which represents a medium-mobility environment. In this mobility environment, the service link lifetime prediction method provides the best prediction results.

6.8   Impact of F Function

In the simulation described above, the disruption penalty function F takes a linear form. We now study the performance of our MDSCR algorithm under different shapes of the F function. Figure 23 compares the improvement ratios under linear, concave, and convex functions. Each experiment also runs over
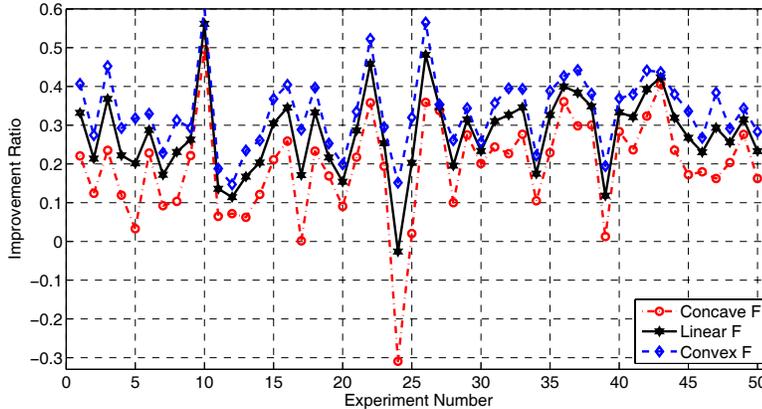

Fig. 23. Improvement Ratio Comparison for Concave, Linear, and Convex Penalty Function F

50 different random network topologies for $2,000s$.

Figure 23 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 the SPSCR algorithm. In the concave region, conversely, the benefits of local recovery are not significant, and the advantages of MDSCR are therefore less prominent.

7   Discussion

There are two general types of network routing strategies for mobile ad hoc networks: (1) *reactive protocols*, such as DSR (27) and $AODV$ (18), and (2) *proactive protocols*, such as $DSDV$ (28). The service composition and recovery strategies presented in this paper use a reactive strategy, *i.e.*, for service-level routing and recovery, a new service path is established only after the current

service path fails. These strategies thus work the best with reactive ad hoc routing protocols.

We use *AODV* for network-level routing and recovery in this paper. Similar to reactive networking protocols, reactive service composition and recovery strategies can cause longer recovery latency. Conversely, since reactive strategies do not require constant maintenance of the service link, so they may incur lower overhead than proactive strategies, especially in a highly dynamic networking environment.

Our service-level composition and recovery algorithm is a centralized algorithm, *i.e.*, a centralized service composition manager contacts the discovery service to locate service components and perform the service composition and recovery computation. This manager could reside with the first component (sender) or the last component (receiver) of the service. Our future work will use a hop-by-hop service path routing that selects components in a distributed manner to reduce the computational overhead of service-level routing.

8 Related Work

Our work is positioned in the overlapping area of service composition for service-oriented networks and reliable network routing in mobile ad hoc networks. This section reviews the existing literature in these two areas to compare and highlight the contribution of our work.

Component-based software development focuses on building software systems by integrating reusable software components (2; 29). At the foundation of this technique is the requirement that all application components are constructed as *autonomous services*, which perform independent operations. *Service composition* is a crucial technology for integrating loosely coupled distributed service components into a composite service that provides a comprehensive function for end users. The existing literature focuses on the following two key issues in service composition:

• **The quality of the composed service path**, which is measured via QoS performance metrics, such as the delay, bandwidth, and reliability. For example, Xu et al. (4) find service paths to optimize the end-to-end resource availability with controlled system overhead. In (6; 7), multiple QoS criteria are aggregated for service path selection and optimization. The scalable service composition is investigated in (30; 31) for large scale systems, by employing distributed and hierarchical routing techniques.

• **Failure recovery in service disruptions**. Raman et al. (8) presents an

architecture for quick service path recovery using service replicas and tuning the process of failure detection, focusing mainly on architectural discussions. Li et al. (9) present a theoretical model for interference-aware service routing in overlay networks.

Our work differs from prior work by considering the intermittent link connectivity and dynamic network topology caused by node mobility in constructing and recovering the service paths.

There is also extensive research on achieving reliable data delivery in mobile ad hoc networks. For example, (10) presents a reliability framework for mobile ad hoc routing, which uses the position and trajectory information of the so-called reliable nodes (in terms of robust and secure) to build the reliable path. Likewise, (11; 32; 33; 34; 35) present reliable routing solutions based on mobility prediction to predict the future availability of wireless links and adapt the mobile routing mechanisms. These studies focus on building stable end-to-end connections at the network layer. In contrast, our work considers the interaction between the service layer and the network layer.

Our work is also closely related to work on the component-based service support for mobile environments. For example, (12) studies how to distribute the software components onto hardware nodes so that the system availability is maximized. It takes into account the overall system availability with regard to connection failures and presents a fast approximative solution. This algorithm is based on the knowledge of connection reliability, which may be impractical since (1) connection reliability is hard to be accurately estimated and (2) even it is able to be measured, reliability is usually a dynamic metric whose value may constantly change with node mobility. Thus it may cause repeated component deployments, especially if the goal is to maximize the overall system availability.

Mobility prediction has also been applied to service component replication strategies (36; 37) to provide continuous service despite of network partition. Moreover, (38) presents a distributed architecture and associated protocols for service composition in mobile environments. The composition protocols are based on distributed brokerage mechanisms and utilize a distributed service discovery process over ad hoc network connectivity. Our work is complimentary to—yet different from—this existing work. First we study the theoretical modeling and algorithm design for service composition and recovery, which is different from the work of (38) that focuses on the architecture design. We also assume that the service components are already deployed over the network, where the existing service deployment and replication strategies (36; 37) could be applied.

31

9   Concluding Remarks

This paper systematically investigates the service composition and recovery strategies that improve the performance of service delivery in mobile ad hoc networks under frequent wireless link failures. It develops a theoretical framework for minimum disruption service composition and recovery based on dynamic programming. Based on the analytical properties of the optimal solution, it further presents a MDSCR heuristic algorithm that provides an effective service composition and recovery solution for ad hoc networks with uncertain node mobility.

We learned the following lessons based on our research conducted thus far:

• **Our simulation results show that the MDSCR algorithm can achieve higher and smoother throughput and smaller disruption index to end users compared with the traditional methods (*e.g.*, the shortest path routing and service composition).** The benefits are particularly notable in scenarios with stringent service link length requirements, networks with medium mobility, and/or the type of impatient users (convex F function). Our future work will validate the performance of the MDSCR heuristic algorithm in our middleware framework (5) and study its performance based on production system deployments in representative mobile ad hoc network domains.

• **The predicted service link lifetime is significantly affected by system dynamics**. When using linear regression to predict the service link lifetime, we need to calculate the results based on different node speeds and pause times.

• **Our experiments involve two steps: prediction and simulation**. The first step is prediction, *i.e.*, we predict the service link lifetime and based on its result, calculate the service path failure rate. The second step is simulation, *i.e.*, we use the prediction results provided by the first step to select the service path and then calculate the disruption index and throughput.

A   Proof of Theorem 1.

Theorem 1: Let $\Pi^* = (\pi^*(t_1), ..., \pi^*(t_l))$ be the optimal MDSCR policy 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 $\mathcal{G}(\tau_i)$ ($\tau_i \leq t_{w+1} < \tau_{i+1}$, $\tau_i, \tau_{i+1} \in \mathcal{T}'$) at t_{w+1}.

Proof: Suppose that the above theorem does not hold and there exist an optimal MDSCR policy A: $\Pi^A = (\pi^A(t_1), ..., \pi^A(t_l))$ where there exist w and two consecutive service compositions $\pi^A(t_w)$ and $\pi^A(t_{w+1})$ so that $\pi^A(t_w)$ is feasible on the network topology $\mathcal{G}(\tau_i)$ $(\tau_i \leq t_{w+1} < \tau_{i+1})$ at t_{w+1}. Let $t_{w+h})$ be the first time instance after t_{w+1} when composition $\pi^A(t_w)$ is not feasible on the network topology at that time $(t_{w+h}))$, and $\pi^A(t_{w+h})$ is the composition used at that time instance. The disruption index for policy A is then given as

$$\tilde{D}^A = \frac{\sum_{v=1}^{l-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})}{T} \tag{A.1}$$

$$= \frac{1}{T}\{\sum_{v=1}^{w-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})}) + \sum_{v=w+h+1}^{l-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})$$

$$+ \sum_{v=w}^{w+h} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})\}$$

Let us consider policy B: $\Pi^B = (\pi^B(t_1), ...\pi^B(t_w), \pi^B(t_{w+h})\pi^B(t_l))$. For each composition in policy B, $\pi^B(t_v) = \pi^A(t_v)$, for $v = t_1, ..., t_w, t_{w+h}, ..., t_l$. The disruption index for policy B is

$$\tilde{D}^B = \frac{\sum_{v=1}^{w} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})}{T} + \frac{\sum_{v=w+h}^{l-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})}{T} \tag{A.2}$$

$$= \frac{1}{T}\{\sum_{v=1}^{w-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})}) + \sum_{v=w+h+1}^{l-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})$$

$$+ F(\beta \times N_{\pi^B(t_w) \to \pi^B(t_{w+h})})\}$$

Obviously, $N_{\pi^B(t_w) \to \pi^B(t_{w+h})} \leq \sum_{v=w}^{w+h} N_{\pi^A(t_v) \to \pi^A(t_{v+1})}$. Since $F(\cdot)$ is a linear or concave function, we have that

$$F(N_{\pi^B(t_w) \to \pi^B(t_{w+h})}) \leq F(\sum_{v=w}^{w+h} N_{\pi^A(t_v) \to \pi^A(t_{v+1})}) \tag{A.3}$$

Thus $\tilde{D}^B \leq \tilde{D}^A$, which is a contradiction, since policy A is claimed as the optimal solution. $\square$

B Proof of Theorem 2.

Theorem 2: Let $\Pi^* = (\pi^*(t_1), ..., \pi^*(t_l))$ be the optimal MDSCR policy. Consider a sub-sequence of service compositions in Π^* where service components are changed. We denote this sub-sequence only with its service routing scheme as $\Pi^*_{\mathcal{S}} = (\pi^*_{\mathcal{S}}(t^s_1), ..., \pi^*_{\mathcal{S}}(t^s_g))$. Then for any two consecutive service compositions in $\Pi^*_{\mathcal{S}}$, $\pi^*_{\mathcal{S}}(t^s_w)$ and $\pi^*_{\mathcal{S}}(t^s_{w+1})$, $\pi^*_{\mathcal{S}}(t^s_w)$ is **not** feasible on the network topology $\mathcal{G}(\tau_i)$ $(\tau_i \leq t^s_{w+1} < \tau_{i+1}, \tau_i, \tau_{i+1} \in \mathcal{T}')$ at t^s_{w+1}.

Proof: Suppose that the above theorem does not hold and there exist an optimal MDSCR policy A whose service routing scheme $\Pi_{\mathcal{S}}^A = (\pi_{\mathcal{S}}^A(t_1^s), ..., \pi_{\mathcal{S}}^A(t_g^s))$. This policy has two consecutive service compositions $\pi^A(t_w)$ and $\pi^A(t_{w+1})$ in $\Pi_{\mathcal{S}}^A$ so that $\pi_{\mathcal{S}}^A(t_w)$ is feasible on the network topology $\mathcal{G}(\tau_i)$ ($\tau_i \leq t_{w+1} < \tau_{i+1}$) at t_{w+1}. Let $t_{w+h})$ be the first time instance after t_{w+1} when composition $\pi^A(t_w)$ is not feasible on the network topology at that time (t_{w+h})), and $\pi^A(t_{w+h})$ is the composition used at that time instance. The disruption index for policy A is then given as

$$\tilde{D}^A = \frac{\sum_{v=1}^{l-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})}{T} \tag{B.1}$$

$$= \frac{1}{T}\{\sum_{v=1}^{w-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})}) + \sum_{v=w+h+1}^{l-1} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})$$

$$+ \sum_{v=w}^{w+h} F(\beta \times N_{\pi^A(t_v) \to \pi^A(t_{v+1})})\}$$

Let us consider policy B: $\Pi^B = (\pi^B(t_1), ...\pi^B(t_w), \pi^B(t_{w+h})\pi^B(t_l))$. For each composition in policy B, $\pi^B(t_v) = \pi^A(t_v)$, for $v = t_1, ..., t_w, t_{w+h}, ..., t_l$. The disruption index for policy B is

$$\tilde{D}^B = \frac{\sum_{v=1}^{w} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})}{T} + \frac{\sum_{v=w+h}^{l-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})}{T} \tag{B.2}$$

$$= \frac{1}{T}\{\sum_{v=1}^{w-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})}) + \sum_{v=w+h+1}^{l-1} F(\beta \times N_{\pi^B(t_v) \to \pi^B(t_{v+1})})$$

$$+ F(\beta \times N_{\pi^B(t_w) \to \pi^B(t_{w+h})})\}$$

Obviously, $N_{\pi^B(t_w) \to \pi^B(t_{w+h})} \leq \sum_{v=w}^{w+h} N_{\pi^A(t_v) \to \pi^A(t_{v+1})}$. Since $F(\cdot)$ is a linear or concave function, we have that

$$F(N_{\pi^B(t_w) \to \pi^B(t_{w+h})}) \leq F(\sum_{v=w}^{w+h} N_{\pi^A(t_v) \to \pi^A(t_{v+1})}) \tag{B.3}$$

Thus $\tilde{D}^B \leq \tilde{D}^A$, which is a contradiction, since policy A is claimed as the optimal solution.
□

References

[1]   R. Ramanathan, J. Redi, A brief overview of ad hoc networks: challenges and directions, IEEE Communications Magazine 40 (5).

[2]   V. Subramonian, G. Deng, C. Gill, J. Balasubramanian, L. Shen, W. Otte, D. C. Schmidt, A. Gokhale, N. Wang, The Design and Performance of

Component Middleware for QoS-enabled Deployment and Configuration of DRE Systems, Journal of Systems and Software: Special Issue on Component-Based Software Engineering of Trustworthy Embedded Systems 80 (5).

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

[4] D. Xu, K. Nahrstedt, Finding Service Paths in a Media Service Proxy Network, in: Proc. of SPIE/ACM MMCN, 2002.

[5] J. Balasubramanian, P. Lardieri, D. C. Schmidt, G. Thaker, A. Gokhale, 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 80 (7).

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

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

[8] B. Raman, R. H. Katz, An Architecture for Highly Available Wide-Area Service Composition, Computer Communications Journal 26 (15).

[9] L. Xiao, K. Nahrstedt, Minimum User-perceived Interference Routing in Service Composition, in: Proc. of IEEE Infocom, 2006.

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

[11] J. Tang, G. Xue, W. Zhang, Reliable Ad Hoc Routing Based on Mobility Prediction, Journal of Combinatorial Optimization 11 (1).

[12] M. Mikic-Rakic, S. Malek, N. Medvidovic, Improving Availability in Large, Distributed Component-Based Systems via Redeployment, in: Proc. of International Working Conference on Component Deployment, 2005.

[13] S. Jiang, Y. Xue, D. C. Schmidt, Disruption-Aware Service Composition and Recovery in Dynamic Networking Environments, in: Workshop on Automating Service Quality (WRAQS), 2007.

[14] S. Jiang, Y. Xue, D. C. Schmidt, Minimum Disruption Service Composition and Recovery in Mobile Ad hoc Networks, in: Proc. of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS), 2007.

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

[16] U. C. Kozat, 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 2 (1).

[17] R. Koodli, C. Perkins, Service Discovery in On-Demand Ad Hoc Net-

works, in: Internet Draft, 2002.

[18] C. E. Perkins, E. M. Belding-Royer, 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] Y. Xue, B. Li, K. Nahrstedt, A Scalable Location Management Scheme in Mobile Ad-hoc Networks, in: Proc. of IEEE Annual Conference on Local Computer Networks, 2001.

[21] http://www.isi.edu/nsnam/ns/doc/index.html.

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

[23] I. B. Arpinar, B. Aleman-Meza, R. Zhang, A. Maduko, Ontology-Driven Web Services Composition Platform, in: Proc. of IEEE International Conference on E-Commerce Technology (CEC), 2004.

[24] T. Yu, Y. Zhang, K. Lin, Efficient Algorithms for Web Services Selection with End-to-end QoS Constraints, Journal of ACM Transactions on the Web (TWEB) 1 (1).

[25] Z. Wang, J. Crowcroft, Analysis of Shortest-path Routing Algorithms in a Dynamic Network Environment, Journal of ACM SIGCOMM Computer Communication Review 22 (2).

[26] S. Karandikar, S. Kalyanaraman, P. Bagal, B. Packer, TCP rate control, Journal of ACM SIGCOMM Computer Communication Review 30 (1).

[27] D. B. Johnson, D. A. Maltz, Y. Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), in: IETF Internet draft, 2003.

[28] C. E. Perkins, P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, in: ACM SIG-COMM'94 Conference on Communications Architectures, Protocols and Applications, 1994.

[29] G. Deng, J. Balasubramanian, W. Otte, D. C. Schmidt, A. Gokhale, DAnCE: A QoS-enabled Component Deployment and Configuration Engine, in: Proc. of Working Conference on Component Deployment, 2005.

[30] J. Jin, K. Nahrstedt, On Exploring Performance Optimizations in Web Service Composition, in: Proc. of ACM/IFIP/USENIX International Middleware, 2004.

[31] X. Gu, K. Nahrstedt, A Scalable QoS-Aware Service Aggregation Model for Peer-to-Peer Computing Grids, in: Proc. of IEEE HPDC, 2002.

[32] S. Jiang, D. He, J. Rao, A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks, in: Proc. of IEEE Infocom, 2001.

[33] J. Tang, G. Xue, W. Zhang, Reliable Routing in Mobile Ad Hoc Networks Based on Mobility Prediction, in: Proc. of IEEE Mobile Ad-hoc and Sensor Systems, 2004.

[34] B. Sathyaraj, R. Doss, Route Maintenance Using Mobility Prediction for Mobile Ad Hoc Networks, in: Proc. of IEEE Mobile Ad-hoc and Sensor Systems, 2005.

[35] N. Sadagopan, F. Bai, B. Krishnamachari, A. Helmy, PATHS: Analysis of

PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols, in: Proc. of ACM MobiHoc, 2003.

[36] B. Li, K. Wang, NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility, IEEE Journal on Selected Areas in Communications 21 (10).

[37] S. Sivavakeesar, G. Pavlou, A. Liotta, Stable Clustering through Mobility Prediction for Large-scale Multihop Intelligent Ad Hoc Networks, in: Proc. of IEEE WCNC, 2004.

[38] D. Chakraborty, A. Joshi, T. Finin, Y. Yesha, Service Composition for Mobile Environments, Journal of Mobile Networks and Applications 10 (4).