NETWORK AND SERVICE FAILURE RESTORATION AND PREVENTION IN MULTI-HOP WIRELESS AND MOBILE NETWORKS

By

Shanshan Jiang

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Computer Science

December 2009

Nashville, Tennessee

Approved:

Professor Yuan Xue
Professor Douglas C. Schmidt
Professor Gabor Karsai
Professor Aniruddha Gokhale
Professor William H. Robinson

To my loving son, Hanlun Li, endlessly inspiring
and
To my beloved husband, Bo Li, infinitely supportive

ACKNOWLEDGEMENTS

I am greatly appreciative and thankful to my advisor, Dr. Yuan Xue, for her scientific vision, detailed guidance, great patience, and tremendous encouragement throughout these years. Dr. Xue has guided me through my transition from an incoming graduate student to an outgoing researcher. In addition, to my mentor, Dr. Douglas C. Schmidt, thank you for helping me improve both my research ability and technical expertise. I also thank Dr. Gabor Karsai, Dr. Aniruddha Gokhale, and Dr. William H. Robinson for serving on my dissertation committee. I appreciate their guidance and advice.

I also want to acknowledge NSF TRUST (The Team for Research in Ubiquitous Secure Technology) for supporting me during my dissertation research.

I would also like to thank my colleagues here at Institute for Software Integrated Systems, especially the members of VANETS group and DOC group, for embodying the spirit of open and altruistic academic collaboration, through which we have all produced innovative research results.

Most importantly, I want to express my deepest gratitude to my loving family. I am indebted to my mom and dad for their support, encouragement, and belief in me. Special thanks to my husband, Bo Li, who makes all my work worthwhile. This work is dedicated to them.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Wireless Networks

Over the recent years, wireless networking technologies such as cellular networks, wireless local area networking (WLAN), and wireless mesh networks are becoming an integral part of our communication environment due to its convenience and flexibility. Such a strong proliferation in the use of wireless technology all over the world creates new research and business opportunities for both its producers and consumers [26].

This dissertation investigates a significant branch of wireless networks – multi-hop wireless networks. A multi-hop wireless network is formed by a collection of wireless nodes which are capable of communicating with each other over a certain range of wireless spectrum. These nodes cooperate to relay traffic throughout the network via multiple hops. Since a multi-hop network can be deployed rapidly without the support of a fixed networking infrastructure, it can be found in a wide range of commercial and military applications, such as residential Internet access, smart home/office, first responder networks, security/monitoring systems, etc.

Problem Description and Research Goal

In this dissertation, we study the problem of *failure restoration and prevention* in multi-hop wireless networks. Wireless networks are more prone to failures than their wireline counterparts. This dissertation considers the following types of failures in wireless networks.

- *Node Failure.* Usually deployed in unplanned, sometimes hostile environments, wireless nodes are particularly vulnerable to failures due to a variety of reasons such as power exhaustion, software and hardware faults, natural disasters, malicious attacks, and human errors.

- *Channel Failure.* Built over an open, shared wireless medium, wireless networks suffer from all kinds of disturbances in the wireless channel, which may be caused by environmental noises and interferences or malicious attacks. In particular, when the jamming attack is launched, the jamming nodes may consume part or all of the channel capacity and cause permanent or intermittent channel failures.

- *Link/Route Failure.* Different from wired link, a wireless link is an abstraction of the communication between two nodes that are within the communication range of each other. When these two nodes move out of range, the wireless link they form will fail. In a multi-hop wireless network where mobile nodes serve as the route relays, such link failures will further lead to route failures.

- *QoS Failure.* Wireless networks typically operate over the shared wireless spectrum with limited and variable channel bandwidth. Packet deliveries in this environment can experience high losses and unpredictable delays. Real-time applications in wireless networks suffer from QoS (Quality of Service) failures [7, 78], when their performance requirements (in terms of packet delivery ratio, delay, throughput, etc) cannot be satisfied.

Compared with traditional wireline networks, the unique characteristics of wireless networks [54, 32, 30, 47, 25, 66, 23, 60, 11, 16, 79, 51, 7, 78] pose several new challenges when dealing with failures outlined above.

- *Shared Channel Resource Model.* In wireless networks, wireless links in the neighborhood interfere with each other and share the channel resource in a location dependent fashion [27]. This is fundamentally different from wireline networks whose communication link capacities are independent from each other. Thus the impact of the node failure in wireless networks is different from the node failure in wireline networks. In wireline networks, node failure is usually translated into the failures of a set of wireline links, which causes a loss of network capacity and thus degraded performance. In wireless networks, when a node fails, the channel resource remains unchanged in that neighborhood. The other adjacent nodes need to take over its relaying role and reroute the traffic. The impact of node failures to the global network performance is thus much harder to evaluate. Furthermore, when wireless channel failure happens, the wireless communication resource is lost on the failed channel. The network can recover its service by switching to a new channel when multiple channels are available. Such a failure recovery strategy is not available for wireline networks.

- *Dynamic Mobile Network Model.* In wireline network, link failures are usually assumed to be independent from each other and follow Poisson arrival process [69]. The link failure in wireless mobile networks is mostly caused by node mobility. Such link failures are correlated with each other. This implies that when a wireless link fails, it is hard to locate an alternative link in its immediate neighborhood. The link failure arrivals also do not follow a Poisson process, but are rather closely related to the node mobility pattern.

The above discussion shows that the failure restoration and prevention approaches proposed for wireline networks (such as the Internet) [22, 52, 70, 39, 45, 13, 8] cannot be directly applied to the wireless environment. The issue of failure restoration and prevention needs a fresh treatment in wireless networks. This dissertation investigates the network and service failure restoration and prevention strategies in multi-hop wireless networks. Its goal is to minimize the service disruption caused by various wireless failures and maximize the network performance after failure restoration.

Research Approach and Dissertation Contribution

This dissertation studies the failure restoration and prevention mechanisms for a multi-hop wireless network at three levels: (1) route restoration at the network level, (2) QoS failure prevention at the network and transport level, and (3) service restoration at the application level. We investigate both stationary and mobile multi-hop wireless networks. For multi-hop wireless stationary networks, we focus on the node failures, channel failures, and QoS failures. For multi-hop wireless mobile networks, we focus on the mobility-caused link failures.

To derive the best failure restoration solution, we employ an optimization-based approach. In particular, we formulate the network restoration strategies in multi-hop wireless stationary networks under node and channel failures as a set of linear programming problems which reflect the unique wireless resource model. For multi-hop wireless mobile networks, we formulate the service restoration strategies as a dynamic programming problem which explicitly incorporates the dynamic network topology model of the mobile network. We investigate the QoS failure prevention strategies over a wireless sensor network through system prototyping and experiment. We build QoS support mechanisms including differentiated scheduling and queue management in this network and use a remote healthcare system as an example real-time application to evaluate its performance.

Network Restoration for Multi-hop Wireless Stationary Networks

This dissertation first presents the network-level route restoration solutions that can minimize the performance degradation in the event of node failures and jamming attacks.

- *Network Restoration under Node Failures.* We study the following failure scenarios within this category: (1) *Single Node Failure.* Exactly one node of the network fails. Thus all traffic to, from and through the node is affected. This may happen as a result of human errors, software and hardware faults, and power exhaustion. (2) *Dispersed Multiple Node Failure.* Several nodes fail independently. This is due to the same reason as single node failure except multiple nodes fail simultaneously. (3) *Regional Multiple Node Failure.* The nodes within a certain geographic region fail. This may happen as a result of malicious attacks and natural disasters.

- *Network Restoration under Jamming Attacks.* We investigate the network restoration solutions via the joint design of traffic rerouting, channel re-assignment, and scheduling for a multi-hop wireless network with multi-channel multi-radio support. Efficient routing and channel assignment schemes can relieve the interference caused by both normal network nodes and jamming nodes.

We formulate the maximum performance network restoration problem under the above failure scenarios as a set of linear programming problems. In particular, we consider different network restoration strategies, from *global restoration* to *local restoration*, which can support a range of tradeoffs between the restoration latency and network throughput after restoration. Two performance degradation indices, *transient disruption index (TDI)* and *throughput degradation index (THI)*, are defined to quantitatively evaluate the impact of both node failures and jamming attacks during and after restoration.

Service Restoration for Multi-hop Wireless Mobile Networks

This dissertation also presents the application-level service composition and restoration framework which achieves minimum service disruptions in multi-hop wireless mobile 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 restoration (MDSCR) as a dynamic programming problem and analyze the properties of its optimal solution for wireless mobile networks with known mobility plan. Based on the derived analytical insights, we present our MDSCR heuristic algorithm for wireless mobile 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. This approach addresses the correlated wireless link failures in the multi-hop wireless mobile networks.

QoS Failure Prevention for Wireless Remote Healthcare System

Finally, this dissertation presents a QoS support mechanism for multi-hop wireless networks that prevents QoS failures. The QoS support mechanism integrates an XML-based QoS specification, admission control policy, and differentiated scheduling and queue management. The QoS support mechanism is built into a heterogenous wireless sensor network consisting of medical sensors and wireless gateways, called *Carenet*. A remote healthcare system is built over this QoS-enable wireless sensor network to capture and transmit in real time the medical data of the patients to their care providers. One of the most important requirements of this system is to assure the timely and robust delivery of the life-critical medical data in the resource-constrained wireless sensor networking environment. We use this remote healthcare system as an example real-time application to evaluate the performance of our integrated QoS support mechanism in *Carenet*.

The *main contributions* of this dissertation are as follows. First, we develop two optimization-based theoretical frameworks – a linear programming framework for route restoration in wireless stationary networks and a dynamic programming framework for service restoration in wireless mobile networks. These frameworks capture the unique characteristics of wireless networks including the shared channel resource model and the dynamic network topology model. Second, we present a set of novel metrics that quantitatively define the performance of network and service restoration strategies. The metrics capture two important aspects of failure recovery – restoration overhead and after-restoration network performance. Third, extensive simulation study and system experiments are conducted to evaluate the impact of various failures and the performance of different failure restoration and prevention strategies on multi-hop wireless networks. These performance results provide valuable guidelines towards practical system design for multi-hop wireless networks.

Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter II provides the network restoration strategies for multi-hop wireless stationary networks under node failures and jamming attacks, defines the performance degradation model, and evaluates the performance of our network restoration algorithms. Chapter III provides our service model over a multi-hop wireless mobile network, presents our service composition and restoration framework and the disruption model, formulates our Minimum Disruption Service Composition and Restoration (MDSCR) problem and gives its optimal solution, explains our MDSCR heuristic algorithm, and describes our simulation results. Chapter IV introduces the design of our remote healthcare system – *CareNet*, gives the implementation of the system prototype, describes the details of our proposed QoS support scheme, and presents the results of our experiment study. And Chapter V presents the concluding remarks.

Table I.1 summarizes a list of acronyms used in this dissertation.

ACE	ADAPTIVE Communication Environment
AODV	Ad hoc On-Demand Distance Vector Routing
CBR	Constant Bit Rate
ECG	Eelectrocardiogram
ILP	Integer Linear Programming
LP	Linear Programming
MAC	Medium Access Control
MDSCR	Minimum-Disruption Service Composition and Restoration
NS-2	Network Simulator 2
QoS	Quality of Service
RSSCR	random selection service composition and restoration
RTS/CTS	Request To Send/Clear To Send
SPSCR	shortest path service composition and restoration
TCP	Transmission Control Protocol
TDI	Transient Disruption Index
THI	THroughput degradation Index
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Table I.1: Acronyms.

CHAPTER II

MAXIMUM PERFORMANCE NETWORK RESTORATION FOR MULTI-HOP WIRELESS STATIONARY NETWORKS

Introduction

Usually being deployed in unplanned, sometimes hostile environments, multi-hop wireless networks are vulnerable to node failures due to a variety of reasons such as power exhaustion, node departures, software and hardware faults, natural disasters, and human errors. And built upon an open, shared wireless medium, multi-hop wireless networks are also particularly vulnerable to jamming attacks. The ability to deal with node failures and jamming attacks, and maintain an acceptable level of performance degradation is thus a crucial issue in the design of multi-hop wireless networks.

This chapter firstly investigates the network restoration strategies upon node failures. In particular, we have the following failure scenarios: (1) *Single Node Failure* – Exact one node of the network fails. Thus all traffic to, from and through the node is affected. This may happen as a result of human errors, software and hardware faults, and power exhaustion. (2) *Dispersed Multiple Node Failure* – A part of the nodes fail independently. This is due to the same reason as single node failure with multiple nodes failing simultaneously. (3) *Regional Multiple Node Failure* – The nodes within a certain geographic region fail. This may happen as a result of malicious attacks and natural disasters. We use the similar solutions used in the jamming defense strategies.

This chapter secondly investigates the jamming defense strategies via the joint design of traffic rerouting, channel re-assignment, and scheduling in a multi-hop multi-channel wireless network. When jamming occurs, the traffic going through that jamming area is disrupted. The network either switches to different channels other than those of the jammers, and/or its traffic needs to be rerouted around the jamming area. Our network restoration scheme needs to discover alternate paths and channels.

In particular, we consider different restoration strategies, from *global restoration* to *local restoration*, which can support a range of tradeoffs between the restoration latency and network performance after restoration. In *global restoration*, all flows in the network will be rerouted and/or re-assigned to new channels in response to the node failures and jamming attacks. *Local restoration* uses a set of detour paths and channels to route around the failed nodes and the jamming area locally. The local restoration strategy can typically restore service much faster than the global restoration strategy because the restoration is locally activated; while in the global restoration, all flows in the network have to be notified with the traffic disruption information.

The goal of this chapter is to investigate *the network restoration strategies that can minimize the performance degradation in the event of node failures and jamming attacks.* In order to achieve this goal, we apply an optimization-based approach, which formulates the network restoration strategies under the global and local restoration strategies as linear programming problems. In particular, we define the minimum flow throughput *scaling factor* as the network restoration performance metric and seek to maximize its value under these two strategies. Our formulation explicitly incorporates the unique characteristics of wireless network including the wireless interference, and for multi-channel wireless network, the channel assignment. And based on the LP solutions, we provide a greedy scheduling algorithm, which schedules the active edges and channels at each time slot, and a greedy static channel assignment algorithm to avoid the channel switching overhead. To the best of our knowledge, this is the first work that studies the network restoration problem in multi-hop wireless networks using an optimization-based approach.

The main contributions of this chapter are as follows. First, we developed an optimization-based framework for network restoration strategies under node failures and jamming attacks in multi-hop wireless networks. Second, we define two novel indices, *transient disruption index* and *throughput degradation index*, that quantitatively evaluate the performance of network restoration strategies. Third, we provide a greedy scheduling algorithm, which schedules the network traffic, and for the jamming attack problem, the jamming traffic. Fourth, we implement the solutions of different optimization-based network restoration strategies, and provide extensive performance evaluations in multi-hop wireless network under different network restoration strategies and different node failure and jamming attack scenarios.

Related Work

It is worth nothing that there are existing works on routing restoration over wireline networks [22, 52, 70, 39, 45, 13, 8] under link failures. While these works also use an optimization-based approach, their results, however, cannot be extended directly to routing restoration in multi-hop wireless networks due to the different network resource types and different network failure scenarios. First, the wireline and wireless networks are built based on fundamentally different types of network resources. Wireline networks rely on the communication links. When link fails, it will result in the loss of network capacity and thus degraded performance. Multi-hop wireless networks are self-organized by the wireless nodes communicating with each other over wireless spectrum. The wireless link is only an abstraction of such communication. The impact of a node failure in wireless network is different from the node failure in wireline networks, which is usually translated into the failures of a set of wireline links. In wireline networks, the link capacities are independent from each other. In wireless networks, wireless communications share the channel resource in a location-dependent fashion. When a node fails, the channel resource remains unchanged in that neighborhood. The other adjacent node needs to take over its relaying role and reroute the traffic. The potential network capacity loss highly depends on the locations of the adjacent nodes and the restoration strategy. This issue has not been quantitatively investigated so far.

Several complementary approaches are proposed in recent works to address the jamming issue. For example, the work of [73] considers how to detect jamming where congested and jammed scenarios can be differentiated. It introduces the notion of consistency checking, where the packet delivery ratio is used to classify a radio link that has poor utility and signal strength consistency check is performed to classify whether poor link quality is due to jamming. The work of [72] studies the jamming defense strategy over a single-radio multi-channel network and presents two channel-surfing strategies, where the wireless channels are re-assigned or dynamically switched under jamming attacks. The work of [10] designs a jamming-resistant MAC protocol for single-hop wireless networks and the work of [14] evaluates the throughput performance degradation of the IEEE 802.11 MAC protocol under various jamming models, including periodic or memoryless jammers, and channel-oblivious or channel-aware jammers.

System Model

This section provides our network, node failure, and jamming models.

Network Model

We consider a multi-hop wireless network and model it as a directed graph $G = (V, E, C)$, where $v \in V$ represents a wireless node in the network. If this network supports multi-radio multi-channel, C denotes a set of orthogonal wireless channels[1] and we assume each node v is equipped with $\kappa(v)$ radios. Otherwise, C denotes a single wireless channel and $\kappa(v) = 1$.

In this wireless network, we assume that the normal behavior of a wireless node at the MAC layer follows the IEEE 802.11 wireless standard. All nodes have a uniform transmission range denoted by R_T and a uniform interference range denoted by R_I ($R_I \leq R_T$). A transmission edge $e = (v, v') \in E_T$ is formed if the distance between its two nodes $r(v, v')$ satisfies $r(v, v') \leq R_T$; an interference edge $e = (v, v') \in E_I$ is formed if the distance between its two nodes $r(v, v')$ satisfies $R_T < r(v, v') \leq R_I$; and $E = E_T \bigcup E_I$. We assume that the data bit rate (wireless channel capacity) is the same for all edge using channel c and denote it as ϕ_c. In IEEE 802.11 with RTS-CTS exchange, the sending node need to hear the MAC layer acknowledgement from the receiving node, therefore, it requires both the sending node and receiving node to be free of interference. Therefore, packet transmission from node v to v' is successful if and only if (1) there is an transmission edge $e = (v, v') \in E_T$; (2) node v and v' have radios that support a common channel c; (3) any other node $v'' \in V$ within the interference range of the

[1] For example, in the IEEE 802.11b standard, $|C| = 3$.

sending node v or the receiving node v', *i.e.*, $e = (v, v'') \in E_T \bigcup E_I$ or $e = (v', v'') \in E_T \bigcup E_I$, is not transmitting on channel c. Further we define *interference set* $I(e), e \in E_T$ which contains the transmission edges that interfere with transmission edge e.

We assume any two nodes can communicate with each other in our multi-hop wireless network. We call the traffic between any pair of nodes as a *flow* and denote it as $f \in F$, where F is the set of all flows. The sending node of flow f is denoted as s_f and the receiving node of flow f is denoted as r_f. We use d_f to denote the demand of flow f. The traffic of flow f will be routed over multiple paths and multiple channels. We denote the amount of flow f's traffic being routed on edge e over channel c as $x_f(e, c)$. The amount of all flows' traffic on edge e over channel c is then given by $\sum_{f \in F} x_f(e, c)$.

Node Failure Model

Now we consider a multi-hop wireless network with node failures. Let $V^{\mathcal{N}}$ be the set of failed nodes. For simplicity, we assume that the source and destination of a flow will not fail. Further, we use $E_T^{\mathcal{N}}$ to denote the set of failed wireless transmission edges and $E_I^{\mathcal{N}}$ to denote the set of failed wireless interference edges caused by the node failures. The new network, after all the failed nodes and edges are removed, is denoted as $G^* = G - (V^{\mathcal{N}}, E_T^{\mathcal{N}} \bigcup E_I^{\mathcal{N}})$.

Jamming Model

Now we consider a multi-hop wireless network under jamming attacks. $j_c \in J_c$ represents a wireless jamming node at channel c, where J_c is the set of all the jammers detected at channel c and J is the set of all the jammers over all the channels. It has a constant traffic generating rate $0 \leq G_{j_c} \leq \phi_c$. We assume all the jamming nodes have a uniform jamming range R_J. We do not consider the underlying MAC protocol used by the jamming nodes and assume that they are smart jammers that can totally occupy the channels when sending jamming traffic. We use $J_c(e), e \in E_T \bigcup E_I$ to denote the set of jammers who have one or both of the two end nodes of the edge e within its jamming range. We also use $E_T(j_c)$ to denote the set of transmission edges whose sending or receiving node is within the jamming range of j_c. We assume that the jamming range of all the jammers does not overlap with each other.

Table II.1 summarizes the notations used in this chapter.

Routing and Channel Assignment without Restoration

We first study the routing and channel assignment problem in a multi-hop wireless network when there is no node failures or jamming attacks. Here, we aim to achieve the maximum throughput. Understanding this problem helps us to find a best strategy that can minimize the performance degradation to defend against node failures and jamming attacks. Since the network

Notation	**Description**
$G = (V, E, C)$	network model
V	node set
E	edge set
C	channel set
V	node set
E	edge set
C	channel set
$\kappa(v)$	number of radios node v equipped
ϕ_c	channel capacity
R_T	transmission range
R_I	interference range
$e = (v, v') \in E_T$	transmission edge
$e = (v, v') \in E_I$	interference edge
$f \in F$	flow
s_f	sending node of flow f
r_f	receiving node of flow f
d_f	traffic demand of flow f
$x_f(e, c)$	amount of flow f's traffic being routed on edge e over channel c
$V^{\mathcal{N}}$	set of failed nodes
$E_T^{\mathcal{N}}$	set of failed wireless transmission edges
$E_I^{\mathcal{N}}$	set of failed wireless interference edges
$G^* = G - (V^{\mathcal{N}}, E_T^{\mathcal{N}} \bigcup E_I^{\mathcal{N}})$	new network model under node failures
$j_c \in J_c$	wireless jamming node at channel c
G_{j_c}	traffic generating rate of jamming node j_c
R_J	jamming range
$J_c(e), e \in E_T \bigcup E_I$	set of jammers who have one or both of the two end nodes of the edge e within its jamming range
$E_T(j_c)$	set of transmission edges whose sending or receiving node is within the jamming range of j_c
λ	minimum flow throughput scaling factor

Notation	Description
$\lambda^{\mathcal{N}}$	minimum flow throughput scaling factor under node failures
$f' \in F'$	truncation flow of flow f in the new network G^*
$d_{f'}$	truncation flow demand
$x_{f'}(e,c)$	fraction of flow f' routing over edge e and channel c
$\lambda_S^{\mathcal{J}}$	new minimum flow throughput scaling factor after scheduling
$h \in H$	augument flow of flow f in the new network G^*
d_h	augument flow demand
$x_h(e,c)$	fraction of flow h routing over edge e and channel c
λ_h	augument flow scaling factor
$pre(v^{\mathcal{N}})$	set of nodes that sending data of f directly to $v^{\mathcal{N}}$
$post(v^{\mathcal{N}})$	set of nodes that receiving data of f directly from $v^{\mathcal{N}}$
b_f	set of bypass flows
$x_{b_f(p,q)}(e,c)$	traffic demand of $b_f(p,q)$ routed over edge e and channel c
λ_b	bypass flow scaling factor
$\lambda^{\mathcal{J}}$	minimum flow throughput scaling factor under jamming attacks
$in_f(j_c)$	set of nodes within the jamming area of j_c
$pre_f(j_c)$	set of nodes sending data of f directly to one or more nodes in $in_f(j_c)$
$post_f(j_c)$	set of nodes receiving data of f directly from one or more nodes in $in_f(j_c)$
$\lambda_S^{\mathcal{J}}$	new minimum flow throughput scaling factor after scheduling

Table II.1: Key Notations.

performance in a wireless network depends on the achievable channel capacity which in turn relies on the underlying scheduling algorithm, the optimal routing and channel assignment problem is typically considered jointly with scheduling. Under our network and traffic model, optimizing the performance of a multi-hop wireless network via the joint design of routing, channel assignment, and scheduling can be formulated as an integer linear programming problem (ILP) [6, 40, 44], where the objective is to maximize the traffic throughput and the constraints come from the fairness requirements and the wireless channel capacity. To make the integer linear programming problem tractable, existing approaches [6, 40, 44] usually solve its LP (linear programming) relaxation and then scale the solution to achieve feasibility. Based on the results presented in [40, 44], the necessary conditions of channel assignment and scheduling for a multi-radio multi-channel wireless network are summarized as follows:

$$\forall v \in V, \sum_{c\in C} \sum_{\substack{v'\in V,\\ e=(v,v')|(v',v)\in E_T}} \sum_{f\in F} \frac{x_f(e,c)}{\phi_c} \le \kappa(v) \tag{II.1}$$

$$\forall c \in C, \forall e=(v,v') \in E_T \cup E_I, \sum_{\substack{e'=(v,v'')|(v'',v)|\\ (v',v'')|(v'',v')\in E_T}} \sum_{f\in F} \frac{x_f(e',c)}{\phi_c} \le 1 \tag{II.2}$$

Inequality (II.1) gives the ***node radio constraint***. Recall that a wireless node $v \in V$ has $\kappa(v)$ radios, and thus can only support $\kappa(v)$ simultaneous communications. Inequality (IV.2) shows the ***channel congestion constraint*** over an individual channel. It says that for any channel c, the total traffic being routed on any of the transmission edges incident on each of any two interfered nodes should be no more than the channel capacity ϕ_c.

A common metric that characterizes the throughput of a given routing with respect to a certain traffic demand set is the minimum flow throughput *scaling factor*. This is the minimum, over all flows, of the actual flow throughput being routed divided by its throughput demand. Formally, the minimum flow throughput scaling factor λ among all the flows F is defined as follows.

$$\lambda = \min_{f \in F} \lambda(f), \mathrm{w}here \qquad \text{(II.3)}$$
$$\lambda(f) = \frac{1}{d_f}(\sum_{c \in C} \sum_{\substack{v \in V, \\ e=(v,r_f)}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v \in V, \\ e=(r_f,v)}} x_f(e,c))$$

Note that in Equation (II.3), $\sum_{c \in C} \sum_{\substack{v \in V, \\ e=(v,r_f)}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v \in V, \\ e=(r_f,v)}} x_f(e,c)$ is the amount of traffic received at the destination node r_f of flow f over all the channels.

The goal of the optimal multi-hop wireless routing problem is to maximize λ, where at least λd_f amount of throughput can be routed for flow f. This routing optimization problem is formulated as the following linear programming (LP) problem:

$$\text{maximize} \quad \lambda \tag{II.4}$$

subject to

$$\forall v \in V, \sum_{c \in C} \sum_{\substack{v' \in V, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \frac{x_f(e,c)}{\phi_c} \leq \kappa(v) \tag{II.5}$$

$$\forall c \in C, \forall e = (v,v') \in E_T \cup E_I, \sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T}} \sum_{f \in F} \frac{x_f(e',c)}{\phi_c} \leq 1 \tag{II.6}$$

$$\forall f \in F, \forall v \in V - \{s_f, r_f\}, \sum_{c \in C} \sum_{\substack{v' \in V, \\ e=(v',v) \in E_T}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v' \in V \\ e=(v,v') \in E_T}} x_f(e,c) = 0 \tag{II.7}$$

$$\forall f \in F, \sum_{c \in C} \sum_{\substack{v \in V \\ e=(v,r_f) \in E_T}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v \in V, \\ e=(r_f,v) \in E_T}} x_f(e,c) = \lambda d_f \tag{II.8}$$

$$\forall f \in F, \forall c \in C, \forall e \in E_T, x_f(e,c) \geq 0 \tag{II.9}$$

In this formulation, Equation (II.5) and (II.6) come from the necessary conditions of channel assignment and scheduling. Equation (II.7) and (II.8) are the flow conservation conditions. This formulation is a linear programming problem, which can be solved by either a LP solver [5] or a fast approximation algorithm [20]. [2]

[2] In our simulation study, we solve the LP problem in a centralized fashion. In practice, LP algorithms can be implemented in both centralized way or in distributed way. There

Optimal Network Restoration Strategies under Node Failures

This section studies the network restoration strategies under node failures, which include global, end-to-end, and local restorations. We formulate the optimal restoration problem under each strategy using linear programming, from which the best after-restoration throughput performance can be derived.

Global Restoration

We first consider the global restoration strategy. In this strategy, all the flows will be rerouted after the node failure in order to achieve the optimal routing performance in terms of scaling factor in the new network. Formally, let $\lambda^{\mathcal{N}}$ be the minimum flow throughput scaling factor of G^*. The global network restoration strategy is formulated as follows.

$$\text{maximize} \quad \lambda^{\mathcal{N}} \tag{II.10}$$

subject to

$$\forall v \in V - V^{\mathcal{N}}, \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \frac{x_f(e,c)}{\phi_c} \leq \kappa(v) \tag{II.11}$$

$$\forall c \in C, \forall e = (v,v') \in E_T - E_T^{\mathcal{N}} \cup E_I - E_I^{\mathcal{N}}, \sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T - E_T^{\mathcal{N}}}} \sum_{f \in F} \frac{x_f(e',c)}{\phi_c} \leq 1 \tag{II.12}$$

are several works talking about how to solve the lp problem in a distributed way, such as [68, 48].

$$\forall f \in F, \forall v \in V - V^{\mathcal{N}} - \{s_f, r_f\},$$

$$\sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v',v) \in E_T - E_T^{\mathcal{N}}}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}} \\ e=(v,v') \in E_T - E_T^{\mathcal{N}}}} x_f(e,c) = 0 \tag{II.13}$$

$$\forall f \in F, \sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}} \\ e=(v,r_f) \in E_T - E_T^{\mathcal{N}}}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}}, \\ e=(r_f,v) \in E_T - E_T^{\mathcal{N}}}} x_f(e,c) = \lambda d_f \tag{II.14}$$

$$\forall f \in F, \forall c \in C, \forall e \in E_T - E_T^{\mathcal{N}}, x_f(e,c) \geq 0 \tag{II.15}$$

This formulation is similar to the previous formulation (II), except that all the failed nodes and edges are removed. This formulation gives the greatest flexibility in choosing the restoration path.

End-to-end Restoration

Now we proceed to study the end-to-end restoration strategy. Different from the global restoration strategy, this strategy will only reroute the flows that are affected by the node failure(s), *i.e.*, those flows whose paths traverse the failed node(s). Two steps are involved in deriving the optimal end-to-end restoration strategy. First we need to find the flows that are affected by the node failure, so called *unaffected flow truncation.* Then we need to route the affected flows using *flow augmentation restoration.* These two steps are discussed below.

```
Algorithm: Computing a Truncated Flow
In    A flow $f$, $x_f(e,c)$, and a failed edge $e_T^{\mathcal{N}} = (a,b)$ with $a$ or $b \in V^{\mathcal{N}}$
Out   A truncated flow $f'$, $x_{f'}(e,c)$
1     For each node $v \in V$, compute a topological order $t(v)$, such that $\forall v, v', if(v,v') \in E_T, t(v) < t(v')$
2     For all edges $e = (i,j) \neq (a,b), x_{f'}(e,c) = 0$
3     Consider node $v \in V$ sequentially, starting at node $b$ and continuing according to increasing topological order
3.1     Let $In(v)$ be the sum of $x_f(e,c)$ on all incoming edges to $v$
3.2     Let $Out(v)$ be the sum of $x_f(e,c)$ on all outgoing edges from $v$
3.3     For each outgoing edge $e$ from $v$, set $x_{f'}(e,c) = In(v)x_f(e,c)/Out(v)$
4     Consider node $v \in V$ sequentially, starting at node $a$ and continuing according to decreasing topological order
4.1     Let $Out(v)$ be the sum of $x_f(e,c)$ on all outgoing edges from $v$
4.2     Let $In(v)$ be the sum of $x_f(e,c)$ on all incoming edges to $v$
4.3     For each incoming edge $e$ to $v$, set $x_{f'}(,ce) = Out(v)x_f(e,c)/In(v)$
```

Table II.2: Algorithm for Computing a Truncated Flow.

Unaffected Flow Truncation

We define the truncation of flow f in the new network G^* to be a new flow whose affected paths (*i.e.*, paths that utilize edges in $E_T^{\mathcal{N}}$) of flow f are removed and denote it as $f' \in F'$. For each flow $f \in F$, we have the truncated flow demand $0 \leq d_{f'} \leq d_f$ and the fraction of flow f routing over edge e and channel c, $0 \leq x_{f'}(e, c) \leq x_f(e, c)$.

Table II.2 presents the algorithm for computing a truncated flow with a single edge failure. This truncated flow computation is independently applied to all flows. For node failures we apply this procedure iteratively edge-by-edge to the edges that are adjacent to the failed nodes.

Optimal Flow Augmentation Restoration

We define the augmented flow in the new network G^* as the flow that need to be routed in addition to the truncated flow in order to ensure the original traffic demand of each flow, and denote it as $h \in H$. For each flow $f \in F$, we have the augmented flow demand $d_h = d_f - d_{f'}$. The traffic demand of h that is routed over e is denoted as $x_h(e, c)$. Obviously, the augmented flow h and its original flow f have the same source and destination nodes, i.e., $s_h = s_f$, $r_h = r_f$. Because the augmented flows need to share the wireless channel capacity with the truncated flows, both of them need to be scaled again. This scaling factor is denoted as λ_h; obviously, $\lambda_h \leq 1$. The end-to-end network restoration problem is formulated as follows:

$$\text{maximize} \quad \lambda^{\mathcal{N}} \tag{II.16}$$

subject to

$$\forall v \in V - V^{\mathcal{N}}, \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \frac{\lambda_h x_{f'}(e',c) + \sum_{h \in H} x_h(e',c))}{\phi_c} \leq \kappa(v) \tag{II.17}$$

$$\forall c \in C, \forall e = (v,v') \in E_T - E_T^{\mathcal{N}} \cup E_I - E_I^{\mathcal{N}},$$

$$\sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T - E_T^{\mathcal{N}}}} \sum_{f \in F} \frac{\lambda_h x_{f'}(e',c) + \sum_{h \in H} x_h(e',c))}{\phi_c} \leq 1 \tag{II.18}$$

$$\forall h \in H, \forall v \in V - V^{\mathcal{N}} - \{s_h, r_h\},$$

$$\sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v',v) \in E_T - E_T^{\mathcal{N}}}} x_h(e,c) - \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}} \\ e=(v,v') \in E_T - E_T^{\mathcal{N}}}} x_h(e,c) = 0 \tag{II.19}$$

$$\forall h \in H, \sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}} \\ e=(v,r_h) \in E_T - E_T^{\mathcal{N}}}} x_h(e,c) - \sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}}, \\ e=(r_h,v) \in E_T - E_T^{\mathcal{N}}}} x_h(e,c) = \lambda_h d_h \tag{II.20}$$

$$\forall h \in H, \forall c \in C, \forall e \in E_T - E_T^{\mathcal{N}}, x_h(e,c) \geq 0 \tag{II.21}$$

$$\lambda^{\mathcal{N}} = \lambda \lambda_h \tag{II.22}$$

In this formulation, $\lambda_h x_{f'}(e',c)$ is the scaled traffic of the truncated flow f' that is routed over edge e and channel c; Equation (II.19) and (II.20) are the flow conservation conditions for the augmented flows H and $\lambda_h d_h$ is the scaled traffic demand of an augmented flow h. $\lambda^{\mathcal{N}}$ is calculated as the scaling factor λ of the network without node failures multiplied by the new scaling factor λ_h.

Local Restoration

In this restoration strategy, the affected flow paths will be rerouted locally. First we need to find the bypass flows that need to be routed away from the failed nodes. For these flows, their immediate upstream and downstream nodes surrounding the failed area should remain unchanged, by definition.

	Algorithm: Computing a Bypass Flow
In	A flow f, $x_f(e,c)$, and a failed node $v^{\mathcal{N}}$
Out	A set of bypass flows b_f, traffic demands d_{b_f}
1	For each node $v \in V$, if $(v, v^{\mathcal{N}}) \in E_T$, add v to the $pre(v^{\mathcal{N}})$ set
2	For each node $v' \in V$, if $(v^{\mathcal{N}}, v') \in E_T$, add v' to the $post(v^{\mathcal{N}})$ set
3	For each node $v' \in post(v^{\mathcal{N}})$, compute the ratio of the traffic of flow f, passing node $v^{\mathcal{N}}$, that is received at node v': $ratio_f(v^{\mathcal{N}}, v') = \frac{x_f(v^{\mathcal{N}},v',c)}{\sum_{\forall w \in post(v^{\mathcal{N}})} x_f(v^{\mathcal{N}},w,c)}$
4	For each node $v \in pre(v^{\mathcal{N}}), v' \in post(v^{\mathcal{N}})$, compute a sub-bypass flow $b_{v,v'}$, the traffic demand of $b_{v,v'}$: $d_{b_{f(v,v')}} = x_f(v, v^{\mathcal{N}}, c) ratio_f(v^{\mathcal{N}}, v', c)$

Table II.3: Algorithm for Computing a Bypass Flow.

Bypass Flows

For a failed node $v^{\mathcal{N}}$ and a flow f, we denote $pre(v^{\mathcal{N}})$ as the set of nodes that sending data of f directly to $v^{\mathcal{N}}$ and $post(v^{\mathcal{N}})$ as the set of nodes that receiving data of f directly from $v^{\mathcal{N}}$, and define a set of bypass flows b_f in the new network G^*. $b_f(v, v')$ is the bypass flow of flow f from node v to node v', which is defined as

$$\forall v \in pre(v^{\mathcal{N}}), \forall v' \in post(v^{\mathcal{N}}), b_f(v, v') \in b_f \tag{II.23}$$

Table II.4 presents the algorithm for computing a bypass flow with a single node failure. This bypass flow computation is independently applied to all flows and to all failed nodes.

Optimal Bypass Restoration

$x_{b_f(p,q)}(e, c)$ is the traffic demand of $b_f(p, q)$ that is routed over edge e and channel c. The minimum flow throughput scaling factor of the bypass flows is denoted as λ_b. The local network restoration problem is formulated as follows:

$$\text{maximize} \quad \lambda^{\mathcal{N}} \tag{II.24}$$

subject to

$$\forall v \in V - V^{\mathcal{N}}, \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \frac{\lambda_b x_f(e,c) + \sum_{\substack{v^{\mathcal{N}} \in V^{\mathcal{N}}, \\ p \in pre(v^{\mathcal{N}}), \\ q \in post(v^{\mathcal{N}})}} x_{b_{f(p,q)}}(e,c)}{\phi_c} \leq \kappa(v) \tag{II.25}$$

$$\forall c \in C, \forall e = (v,v') \in E_T - E_T^{\mathcal{N}} \cup E_I - E_I^{\mathcal{N}},$$

$$\sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T - E_T^{\mathcal{N}}}} \sum_{f \in F} \frac{\lambda_b x_f(e',c) + \sum_{\substack{v^{\mathcal{N}} \in V^{\mathcal{N}}, \\ p \in pre(v^{\mathcal{N}}), \\ q \in post(v^{\mathcal{N}})}} x_{b_{f(p,q)}}(e',c)}{\phi_c} \leq 1 \tag{II.26}$$

$$\forall f \in F, \forall v^{\mathcal{N}} \in V^{\mathcal{N}}, p \in pre(v^{\mathcal{N}}), q \in post(v^{\mathcal{N}}), \forall v \in V - V^{\mathcal{N}} - \{p,q\},$$

$$\sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}}, \\ e=(v',v) \in E_T - E_T^{\mathcal{N}}}} x_{b_{f(p,q)}}(e,c) - \sum_{c \in C} \sum_{\substack{v' \in V - V^{\mathcal{N}} \\ e=(v,v') \in E_T - E_T^{\mathcal{N}}}} x_{b_{f(p,q)}}(e,c) = 0 \tag{II.27}$$

$$\forall f \in F, \forall v^{\mathcal{N}} \in V^{\mathcal{N}}, p \in pre(v^{\mathcal{N}}), q \in post(v^{\mathcal{N}}),$$

$$\sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}} \\ e=(v,q) \in E_T - E_T^{\mathcal{N}}}} x_{b_{f(p,q)}}(e,c) - \sum_{c \in C} \sum_{\substack{v \in V - V^{\mathcal{N}}, \\ e=(q,v) \in E_T - E_T^{\mathcal{N}}}} x_{b_{f(p,q)}}(e,c) = \lambda_b d_{b_{f(p,q)}} \tag{II.28}$$

$$\forall f \in F, \forall c \in C, \forall v^{\mathcal{N}} \in V^{\mathcal{N}}, p \in pre(v^{\mathcal{N}}), q \in post(v^{\mathcal{N}}), \forall e \in E_T - E_T^{\mathcal{N}},$$

$$x_{b_{f(p,q)}}(e,c) \geq 0 \tag{II.29}$$

$$\lambda^{\mathcal{N}} = \lambda \lambda_b \tag{II.30}$$

In this formulation, $\lambda_b x_f(e',c)$ is the scaled traffic of flow f that is routed over edge e and channel c; Equation (II.27) and (II.28) are the flow conservation conditions for the bypass flows and $\lambda_b d_{b_{f(p,q)}}$ is the scaled traffic demand of a bypass flow $b_f(p, q)$. $\lambda^{\mathcal{N}}$ is calculated as the scaling factor λ of the network without node failures multiplied by the new scaling factor λ_b.

Optimal Network Restoration Strategies under Jamming Attacks

This section studies the network restoration strategies under jamming attacks. The jamming attack issue is similar to node failure issue, except that instead of reducing the network resource, we add more traffic into the network.

In our multi-hop wireless network, when jamming attacks happen, the throughput performance of the network traffic around the jamming nodes is degraded. The disrupted network traffic can be rerouted to use other intermittent nodes away from the jamming area, and/or switched to another channel instead of using the jammed channel. In order to calculate how to do the rerouting as well as channel reassignment, based on the discussion of the optimal multi-hop wireless routing and channel assignment problem, we proceed to study the network restoration strategies under jamming attacks.

In our previous introduced necessary conditions of channel assignment and scheduling, Inequality (IV.2) shows the ***channel congestion constraint*** without jamming attacks. For a wireless network under jamming attacks, the available network bandwidth is consumed partially by the jamming nodes. Therefore, we need to include the jamming traffic into the ***channel congestion constraint***, which is defined as follows:

$$\forall c \in C, \forall e = (v, v') \in E_T \cup E_I, \quad \sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T}} \sum_{f \in F} \frac{x_f(e', c)}{\phi_c}$$

$$\leq 1 - \sum_{j_c \in J_c(e)} \frac{G_{j_c}}{\phi_c} \tag{II.31}$$

Inequality (II.31) together with Inequality (II.1) gives the modified necessary conditions of channel assignment and scheduling for a multi-hop wireless network under jamming attacks.

We consider the network restoration via joint traffic rerouting and channel re-assignment under global and local restoration strategies. We formulate the optimal restoration problem under each strategy using linear programming, from which the best after-restoration throughput performance can be derived.

Global Restoration

We first consider the global restoration strategy. In this strategy, all the flows will be rerouted when there are jamming attacks, in order to achieve the optimal routing performance in terms of scaling factor in the new network. Formally, let $\lambda^{\mathcal{J}}$ be the minimum flow throughput scaling factor of the new network. The global restoration strategy is formulated as follows.

$$\text{maximize} \quad \lambda^{\mathcal{J}} \tag{II.32}$$

subject to

$$\forall v \in V, \sum_{c \in C} \sum_{\substack{v' \in V, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \frac{x_f(e,c)}{\phi_c} \leq \kappa(v) \tag{II.33}$$

$$\forall c \in C, \forall e = (v,v') \in E_T \cup E_I, \sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T}} \sum_{f \in F} \frac{x_f(e',c)}{\phi_c} \leq 1 - \sum_{j_c \in J_c(e)} \frac{G_{j_c}}{\phi_c} \tag{II.34}$$

$$\sum_{c \in C} \sum_{\substack{v' \in V, \\ e=(v',v) \in E_T}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v' \in V \\ e=(v,v') \in E_T}} x_f(e,c) = 0 \tag{II.35}$$

$$\forall f \in F, \sum_{c \in C} \sum_{\substack{v \in V \\ e=(v,r_f) \in E_T}} x_f(e,c) - \sum_{c \in C} \sum_{\substack{v \in V, \\ e=(r_f,v) \in E_T}} x_f(e,c) = \lambda^{\mathcal{J}} d_f \tag{II.36}$$

$$\forall f \in F, \forall c \in C, \forall e \in E_T, x_f(e,c) \geq 0 \tag{II.37}$$

This formulation is similar to the previous formulation (II) except Inequality (II.34). This formulation gives the greatest flexibility in choosing the restoration routes and channels.

Local Restoration

In this restoration strategy, the affected flow paths will be rerouted locally. First we need to find the bypass flows that need to be partially routed away from the jamming area. For these flows, their immediate upstream and downstream nodes surrounding the jamming area should remain unchanged.

Bypass Flows

For a jamming node j_c and a flow f, we denote $in_f(j_c)$ as the set of nodes that are within the jamming area of j_c, $pre_f(j_c)$ as the set of nodes sending data of f directly to one or more nodes in $in_f(j_c)$ and $post_f(j_c)$ as the set of nodes receiving data of f directly from one or more nodes in $in_f(j_c)$. We also define a set of bypass flows b_f of flow f in the new network. $b_f(v, v', j_c)$ is a bypass flow of flow f caused by jamming node j_c, with sending node v and receiving node v' , which is defined as

$$\begin{aligned} &\forall j_c \in J_c, \forall v \in pre_f(j_c), \forall v' \in post(j_c), \\ &b_f(v, v', j_c) \in b_f \qquad \text{(II.38)} \end{aligned}$$

Table II.4 presents the algorithm for computing the bypass flows of a given flow for a single jamming node under a given channel. This bypass flow computation is independently applied to all flows, all jamming nodes, and all channels.

Optimal Bypass Restoration

We use $x_{b_f(v,v',j_c)}(e, c)$ to denote the traffic demand of $b_f(v, v', j_c)$ that is routed over edge e and channel c. Because the bypass flows need to share the wireless channel capacity with the original flows, both of them need to

	Algorithm: Computing Bypass Flows of Flow f
In	A flow f, a channel c, $x_f(e, c)$, and a jamming node j_c
Out	A set of bypass flows b_f, traffic demands d_{b_f}
1	For each node $v \in V$, if $r(v, j_c) \leq R_J$, add v to the $in_f(j_c)$ set
2	For each node $v, v' \in V$, if $(v, v') \in E_T$ and $v' \in in_f(j_c)$, add v to the $pre_f(j_c)$ set
3	For each node $v, v' \in V$, if $(v, v') \in E_T$ and $v \in in_f(j_c)$, add v' to the $post_f(j_c)$ set
4	For each node $v \in post_f(j_c)$, compute the ratio of the traffic of flow f, passing the jamming area of j_c, that is received at node v': $ratio_f(j_c, v) = \frac{\sum_{v' \in in_f(j_c), e=(v',v)} x_f(e,c)}{\sum_{v' \in in_f(j_c), w \in post_f(j_c), e=(v',w)} x_f(e,c)}$
5	For each node $v \in pre(j_c)$, compute the demand of the traffic of flow f, entering the jamming area of j_c, that is sent from node v: $d_f(v, j_c) = \sum_{v'' \in in_f(j_c), e=(v,v'')} x_f(e, c)$
6	For each node $v \in pre(j_c), v' \in post(j_c)$, compute a sub-bypass flow b_{v,v',j_c}, the traffic demand of b_{v,v',j_c}: $d_{b_f}(v, v', j_c) = d_f(v, j_c) \cdot ratio_f(j_c, v')$

Table II.4: Algorithm for Computing Bypass Flows.

be scaled again. This scaling factor is denoted as λ^b. The local restoration problem is formulated as follows:

$$\text{maximize} \quad \lambda^{\mathcal{J}} \tag{II.39}$$

subject to

$$\forall v \in V, \sum_{c \in C} \sum_{\substack{v' \in V, \\ e=(v,v')|(v',v) \in E_T}} \sum_{f \in F} \Big(\frac{\lambda^b x_f(e,c)}{\phi_c} + \sum_{\substack{j_c \in J_c, \\ u \in pre(j_c), \\ u' \in post(j_c)}} \frac{x_{b_f(u,u',j_c)}(e,c)}{\phi_c} \Big) \leq \kappa(v) \tag{II.40}$$

$$\forall c \in C, \forall e = (v,v') \in E_T \cup E_I, \sum_{\substack{e'=(v,v'')|(v'',v)| \\ (v',v'')|(v'',v') \in E_T}} \sum_{f \in F} \Big(\frac{\lambda^b x_f(e',c)}{\phi_c}$$

$$+ \sum_{\substack{j_c \in J_c, \\ u \in pre(j_c), \\ u' \in post(j_c)}} \frac{x_{b_f(u,u',j_c)}(e',c)}{\phi_c} \Big) \leq 1 - \sum_{j_c \in J_c(e)} \frac{G_{j_c}}{\phi_c} \tag{II.41}$$

$\forall f \in F, \forall j_c \in J_c, u \in pre(j_c), u' \in post(j_c), \forall v \in V - \{u, u'\}$,

$$\sum_{c \in C} \sum_{\substack{w \in V, \\ e=(w,v) \in E_T}} x_{b_f(u,u',j_c)}(e,c) - \sum_{c \in C} \sum_{\substack{w \in V, \\ e=(v,w) \in E_T}} x_{b_f(u,u',j_c)}(e,c) = 0 \tag{II.42}$$

$\forall f \in F, \forall j_c \in J_c, u \in pre(j_c), u' \in post(j_c)$,

$$\sum_{c \in C} \sum_{\substack{v \in V, \\ e=(v,u') \in E_T}} x_{b_f(u,u',j_c)}(e,c) - \sum_{c \in C} \sum_{\substack{v \in V, \\ e=(u',v) \in E_T}} x_{b_f(u,u',j_c)}(e,c) = \lambda^b d_{b_f}(u,u',j_c) \tag{II.43}$$

$$\forall f \in F, \forall j_c \in J_c, u \in pre(j_c), u' \in post(j_c), \forall c \in C, \forall e \in E_T, x_{b_f(u,u',j_c)}(e,c) \geq 0 \tag{II.44}$$

$$\lambda^{\mathcal{J}} = \lambda\lambda^{b} \tag{II.45}$$

In this formulation, Inequality (II.40) and (II.41) come from the necessary conditions of channel assignment and scheduling for both the original flows and the bypass flows. Equation (II.42) and (II.43) are the flow conservation conditions for the bypass flows. $\lambda^b x_f(e, c)$ is the scaled traffic of flow f that is routed over edge e and channel c, and $\lambda^b d_{b_f}(u, u', j_c)$ is the scaled traffic demand of a bypass flow $b_f(u, u', j_c)$. $\lambda^{\mathcal{J}}$ is calculated as the scaling factor λ of the network without jamming nodes multiplies the new scaling factor λ^b.

Note that since we use multiple channels, a flow that is jammed by a jamming node j_c under channel c can use all the available channels for rerouting.

Scheduling with Dynamic Channel Assignment

[3]Both the global restoration and the local restoration strategies are based on linear programming, which give an upper bound on the achievable network throughput. We use the results from the LP solutions to schedule which edges and channels are active at each time slot. We consider the dynamic channel assignment problem, where a radio may need to switch to a different channel at different time slots. Dynamic channel assignment provides the maximum flexibility in channel assignment and scheduling. Since the scheduling problem is NP-hard, we use a greedy approach to solve it.

[3] We focus on the jamming attack issue in this section; the node failure issue is similar to it.

After we solve the LP problems for the global restoration and the local restoration strategies, we have a set of flows assigned to edges that have been assigned to different channels. We now begin to schedule both the network traffic on the edges and the jamming traffic. The algorithm is shown in Table II.5. In this algorithm, $I(e^*)$ is the set of transmission edges that interfere with edge e^* and $E(j_c^*)$ is the set of transmission edges that are within the jamming range of jammer j_c^*.

We use N to denote the maximum number of time slots taken by all the edge-channel pairs. The new scaling factor $\lambda_S^{\mathcal{J}}$ after scheduling is calculated as:

$$\lambda_S^{\mathcal{J}} = \frac{\lambda^{\mathcal{J}}}{N \cdot \tau} \tag{II.46}$$

Static Channel Assignment

[4]Although dynamic channel assignment provides the maximum flexibility in channel assignment and scheduling, it also results in channel switching overhead. We further consider the static edge channel assignment problem, where a channel is assigned to an edge at the beginning and will remain fixed over all time slots. The static channel assignment problem is also NP-hard and we use the greedy approach to solve this problem.

[4] We still focus on the jamming attack issue in this section.

```
Algorithm: Greedy Scheduling
In   Calculated $x_f(e,c)$ using LP
Out  Each transmission edge is associated with a set of colors from the
     smallest to the highest, which denotes the time slots the edge is
     scheduled

Calculate the amount of all flows' traffic on transmission edge $e$
over channel $c$: $\forall c \in C, \forall e \in E_T, x(e,c) = \sum_{f \in F} x_f(e,c)$
// Initialize the edge-channel color set
$\forall c \in C, \forall e \in E_T$, associate a null color set to the pair $(e,c)$
// Initialize the node color set
$\forall v \in V$, associate a null color set to the node $v$
// Initialize $x'(e,c)$, which denotes the residual traffic on edge e
$\forall c \in C, \forall e \in E, x'(e,c) = x(e,c)$
// Initialize $G'_{j_c}$, which denotes the residual traffic on jamming $j_c$
$\forall c \in C, \forall j_c \in J_c, G'_{j_c} = G_{j_c}$
// Schedule all the network traffic and jamming traffic
While $\sum_{c \in C} \sum_{e \in E} x'(e,c) + \sum_{c \in C} \sum_{j_c \in J_c} G'_{j_c} \geq 0$
  // Consider edge e with the highest residual traffic
  $max\_edge\_traf = \max_{e \in E} x'(e,c)$
  $(e^*, c^*) = arg \max_{e \in E} x'(e,c)$
  // Consider jammer $j_c$ with the highest residual traffic
  $max\_jam\_traf = \max_{j_c \in J} G_{j_c}$
  $j_c^* = arg \max_{j_c \in J} G_{j_c}$
  If $max\_edge\_traf \geq max\_jam\_traf$
    // Schedule the network traffic on the edge
    $\forall e' \in I(e^*)$, find the smallest color $k_1$, that has not
    been added in the color set of the pair $(e', c^*)$
    $e^* = (v, v')$, find the smallest color $k_2$, that has not
    occurred $\kappa(v)$ times in the color set of the node $v$ and
    has not occurred $\kappa(v')$ times in the color set of the node $v'$
    $k = max(k_1, k_2)$
    $\forall e' \in I(e^*)$, add color $k$ to the color set of the pair $(e', c^*)$
    add color $k$ to the color set of the nodes $v$ and $v'$
    $x'(e^*, c^*) = x'(e^*, c^*) - \phi_c \tau$, where $\tau$ is the length of a
    time slot
```

```
Algorithm: Greedy Scheduling
  Else
    // Schedule the jamming traffic
    $\forall e' \in E(j_c^*)$, find the smallest color $k$ that has not
    been added in the color set of the pair $(e', c)$
    $\forall e' \in E(j_c^*)$, add color $k$ to the color set of the pair $(e', c)$
    $G'_{j_c^*} = G'_{j_c^*} - \phi_c \tau$, where $\tau$ is the length of a time slot
  end If
end While
```

Table II.5: Algorithm for Greedy Scheduling.

Constraint Set

Note that in the definition of the necessary conditions of channel assignment and scheduling, the ***node radio constraint*** and the ***channel congestion constraint*** have a common structure. On the left sides of Inequality (II.1) and (II.31), we have L sets, each of which is composed of (edge, channel) pairs; on the right sides of these inequalities, we have L fixed values, where L is the number of all the expanded inequalities without the $\forall$ sign.

We use $S_1, S_2, ..., S_L$ to denote the sets of (link, channel) pairs and use $\beta_{S_1} - G_{S_1}, \beta_{S_2} - G_{S_2}, ..., \beta_{S_L} - G_{S_L}$ to denote their corresponding values. If S_i comes from the ***node radio constraint***, $\beta_{S_i} = \kappa(v)\phi_c$; if S_i comes from the ***channel congestion constraint***, $\beta_{S_i} = \phi_c$. If S_i comes from the ***node radio constraint***, $G_{S_i} = 0$; if S_i comes from the ***channel congestion***

constraint, $G_{S_i} = \sum_{j_c \in J_c(e)} G_{j_c}$. Therefore, the general form of Inequality (II.1) and (II.31) using constraint sets is defined as follows:

$$\forall i \in 1, 2, ..., L, \sum_{(e,c) \in S_i} x_f(e,c) \leq \beta_{S_i} - G_{S_i} \tag{II.47}$$

Static Channel Assignment

We use a greedy approach in solving the static channel assignment problem. Our static channel assignment algorithm is shown in Table II.6. We calculate the amount of all flows' traffic over all the channels on edge e and denote it as $x(e)$. For simplicity, we assume that only one channel can be assigned to a given edge. Therefore, $x(e)$ is assigned to one particular channel assigned to edge e. The input $x_f(e,c)$ of the algorithm is the amount of flow f's traffic being routed on edge e over channel c $x_f(e,c)$, calculated using LP. The output $x(e,c)$ is the amount of traffic being assigned to edge e over one particular channel c. The basic idea of our static channel assignment algorithm is to distribute the load on the constraint sets as much as possible among the given channels.

Once we get the results of static channel assignment, we can use the similar scheduling algorithm described in Section II to schedule both the network and jamming traffic.

```
Algorithm: Static Channel Assignment
In   Calculated $x_f(e,c)$ using LP
Out  New assigned $x(e,c)$
Calculate the amount of all flows' traffic on edge $e$ over channel $c$: $\forall c \in C, \forall e \in E_T, x(e,c) = \sum_{f \in F} x_f(e,c)$
Calculate the amount of all flows' traffic over all the channels on edge $e$: $\forall e \in E_T, x(e) = \sum_{c \in C} x(e,c)$
// $T(e,c)$ denotes the constraint sets that contain the pair $(e,c)$
// $l_S$ denotes the total traffic that has been assigned to constraint set $S$; it is originally equal to the jamming traffic
$l_S = G_S$
// $E_{left}$ denotes the set of the unassigned edges
$E_{left} = E$
While $\sum_{e \in E} x(e) \geq 0$
  For $\forall e \in E_{left}$
    $\forall c \in C, m(e,c) = \max_{S \in T(e,c)} l_S/\beta_S$
    $w(e) = \min_{c \in C} m(e,c)$
    $b(e) = arg \min_{c \in C} m(e,c)$
  end For
  $e^* = arg \min_{e \in E_{left}} w(e)$
  Assign $e^*$ to channel $b(e^*)$
  $\forall S \in T(e^*, b(e^*)), l_S = l_S + x(e^*)$
  $x(e^*, b(e^*)) = x(e^*); \forall c \neq b(e^*), x(e^*, c) = 0$
  $x(e^*) = 0$
  $E_{left} = E_{left} - \{e^*\}$
end While
```

Table II.6: Algorithm for Balanced Static Channel Assignment.

Performance Degradation Model

A fundamental research challenge for choosing the restoration strategy is to understand its tradeoff between the time and overhead involved in repairing the failed traffic path(s) and the traffic throughput and network congestion after restoration. To study this issue, we first define two novel indices, *transient disruption index (TDI)*, which is based on the repair overhead for the failed traffic path(s) during restoration, and *throughput degradation index (THI)*, which characterizes throughput degradation of the new network after restoration.

Transient Disruption Index (TDI)

We use the number of modified routing table entries as an estimate of the repair overhead for the failed path(s). For local repair, only the boundary nodes outside the failed area will try to find the alternative paths in the vicinity. Local repair therefore involves fewer routing table entry modifications and less restoration time. For global repair, the source node initiates a new route discovery, which takes more time than local repair and involves more routing table entry modifications. We use $r_v(c, v')$ to denote a routing table entry of node v's routing table. At a given channel c, it is calculated as the ratio of the total traffic of all the flows sending from node v to its next-hop node v' to the total traffic of all the flows receiving at node v. Its corresponding routing table entry for the new network under is denoted as

$r_v^*(c, v')$. All the routing table entries of the nodes in the network G is denoted as $r(G)$. The transient disruption index (TDI) can be quantitatively defined as follows:

$$TDI = \frac{1}{|r(G)|} \sum_{\substack{c \in C, \\ v \in V, v' \in V, v \neq v'}} r_v(c, v') \neq r_v^*(c, v') \qquad \text{(II.48)}$$

Throughput Degradation Index (THI)

We use the changes of the minimum flow throughput scaling factor λ as an estimate of the throughput degradation of the new network. For local repair, only the flows affected by the failed area will be rerouted. Local repair therefore achieves partially optimal utilization of the network. For global repair, all flows in the network will be considered in order to get an optimal utilization of the network. The throughput degradation index (THI) can be defined as a function of the minimum flow throughput *scaling factor* $\lambda^{\mathcal{J}}$ of the new network[5] and the original optimal minimum flow throughput *scaling factor* λ:

$$THI = 1 - \frac{\lambda^{\mathcal{J}}}{\lambda} \qquad \text{(II.49)}$$

[5] For the node failure issue, it is $\lambda^{\mathcal{N}}$.

Performance Evaluation under Node Failures

This section evaluates the performance of these three optimal network restoration strategies under different node failure scenarios.

Simulation Setup

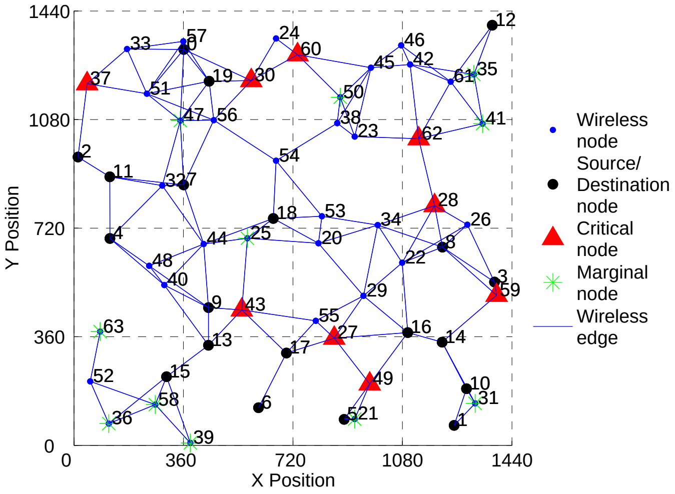

Figure II.1: Example Multi-hop Wireless Network.

In the simulated multi-hop wireless network, 64 wireless nodes are randomly deployed over a 1440×1440 m^2 region. Each node has a transmission range of 250 m and an interference range of 250 m. The channel capacity c is set as 1 $Mbps$. The simulated network topology is shown in Figure II.1.

There are 10 flows in the network with randomly selected sources (node number 0-9) and destinations (node number 10-19). All the flows have the same traffic demand of 10 $pkts/sec$ with packet size of $1k$ Bytes.

We evaluate the performance of global restoration, end-to-end restoration, and local restoration under three failure scenarios:

- Single node failure, where only one node fails in the network;
- Dispersed multiple node failures, where multiple uncorrelated nodes fail;
- Regional multiple node failures, where a set of geographically close nodes fail at the same time. This scenario may happen as a result of physical attack in this region.

Simulation Results

Single Node Failure

To qualitatively characterize the *transient disruption index (TDI)* and *throughput degradation index (THI)* upon different node failure scenarios, we first select each node one by one as the failed node. For simplicity, we do not consider the source and destination nodes and the nodes that cause disconnectivity of the network (node number 49 in the simulated network). Figure II.2 and Figure II.3 show the comparison of TDI and THI for the original network and the new network with global, end-to-end, and local

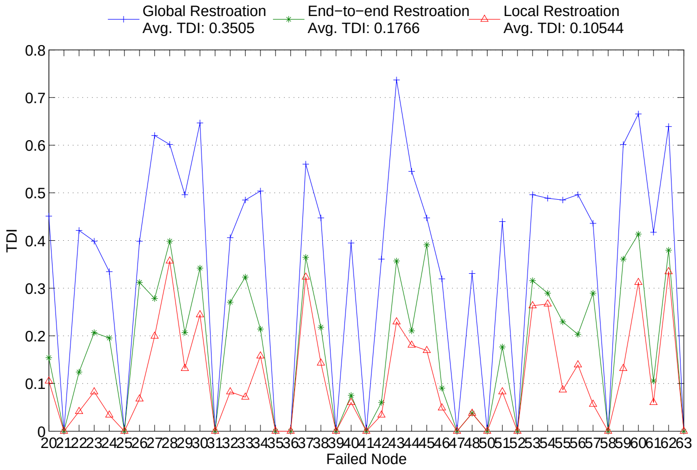

Figure II.2: Comparison of TDI with Single Node Failure.

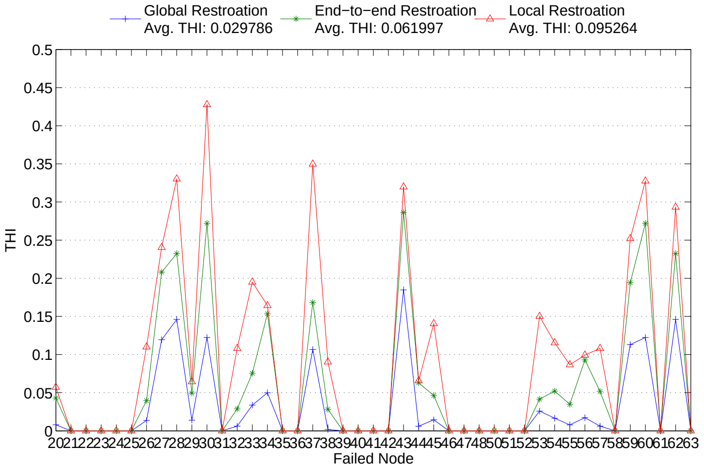

Figure II.3: Comparison of THI with Single Node Failure.

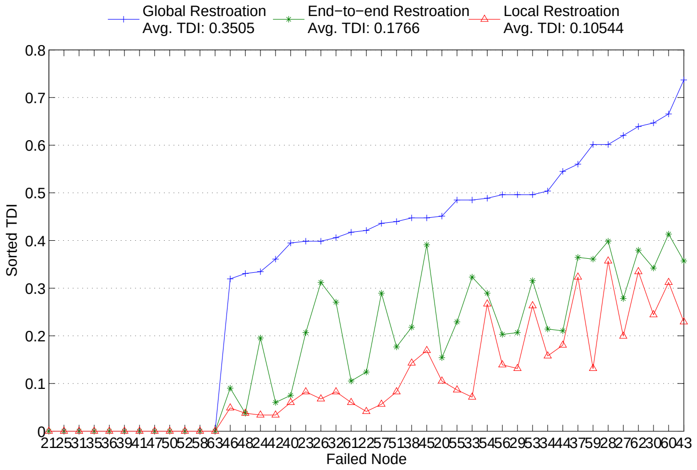

Figure II.4: Comparison of Sorted TDI with Single Node Failure.

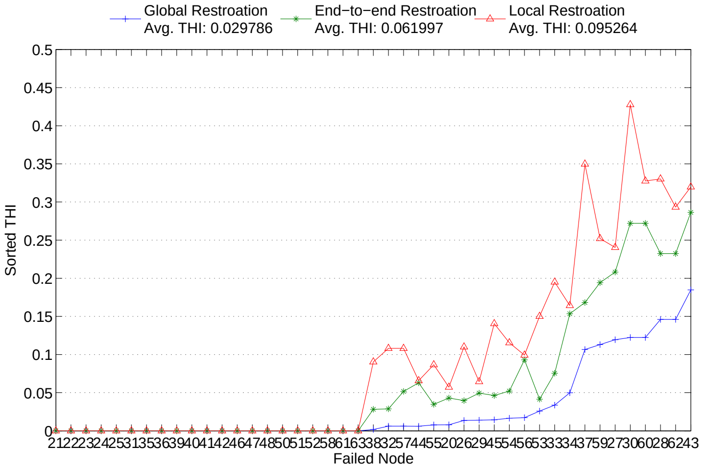

Figure II.5: Comparison of Sorted THI with Single Node Failure.

restorations of single node failure, respectively. To get a clearer observation, we sort the values of TDI and THI in Figure II.4 and Figure II.5. From these four figures, we can see that the failures of some nodes do not change the value of λ and do not modify the routing results; they are marked as green star in Figure II.1. The reason is that they are marginal nodes that do not have traffic passing through them. And we can see that the failures of another group of the nodes do not change the value of λ, however, they modify the routing results. The reason is that the traffic passing through them is redirected to their adjacent nodes without any performance decrease (as discussed in Section II). We can also see that the failures of a third part of the nodes dramatically change the value of λ; they are marked as red triangle in Figure II.1. The reason is that they are critical nodes and have lots of traffic passing through; most of them are the joint points of different network partitions. Comparing these three strategies, we could easily see that the global restoration has lower average THI than the end-to-end strategy, which also has lower average THI than the local strategy; however, the global restoration has higher average TDI than the end-to-end strategy, which also has higher average TDI than the local strategy.

Dispersed Multiple Node Failures

In this set of simulation studies, we investigate the performance of network restoration strategies under simultaneous node failures. In our study,

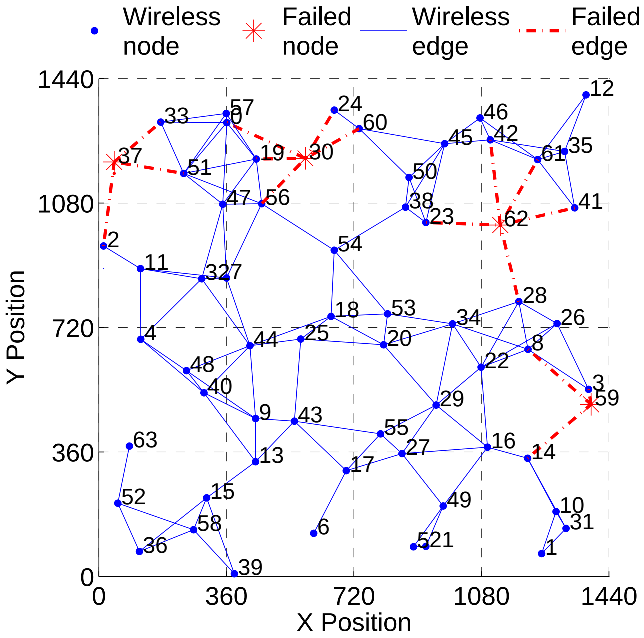

Figure II.6: Example Multi-hop Wireless Network with Dispersed Multiple Node Failures.

Restoration Strategy	TDI	THI
Global Restoration	65.79%	27.30%
End-to-end Restoration	63.16%	31.51%
Local Restoration	57.89%	35.27%

Table II.7: TDI and THI of Global Restoration, End-to-end Restoration, and Local Restoration under Dispersed Multiple Node Failures.

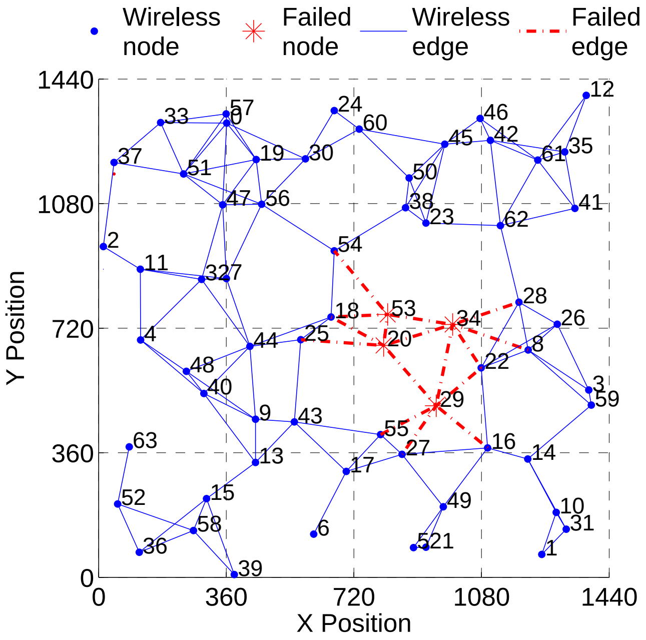

Figure II.7: Example Multi-hop Wireless Network with Regional Multiple Node Failures.

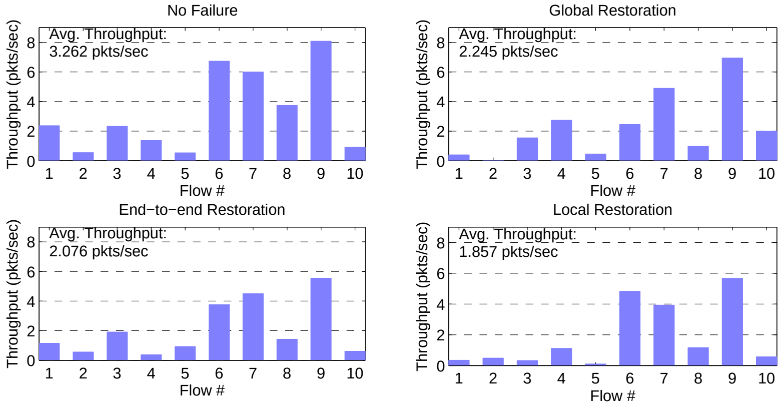

Figure II.8: Throughput Comparison of No Failure, Global Restoration, End-to-end Restoration, and Local Restoration under Dispersed Multiple Node Failures.

several nodes other than the source and destination nodes are randomly independently selected as the failed nodes. The simulated network topology with dispersed multiple node (30, 37, 59, and 62) failures is shown in Figure II.6.

Table II.7 presents the results of TDI and THI of global restoration, end-to-end restoration, and local restoration under dispersed multiple node failures. From this table, we can see that the global restoration causes more TDI and less THI than the end-to-end restoration, and the end-to-end restoration in turn causes more TDI and less THI than the local restoration.

Figure II.8 shows the throughput comparison of the original network and the failed network with global, end-to-end, and local restorations of dispersed multiple node failures. The packet size is 1000 bytes. From this figure, we can see that the average throughput of the global restoration (2.245 $pkts/sec$) is greater than the end-to-end restoration (2.076 $pkts/sec$), and the end-to-end restoration is greater than the local restoration (1.857 $pkts/sec$).

Regional Multiple Node Failures

In this set of simulation studies, we investigate the performance of network restoration strategies under regional node failures. In our study, a certain geographic region is randomly selected which contains all the failed nodes. The simulated network topology with regional multiple node (20, 29, 34, and 53) failures is shown in Figure II.7.

Table II.8 presents the results of TDI and THI of global restoration, end-to-end restoration, and local restoration under regional multiple node failures. From this table, we can see that the global restoration causes more (less) transient disruption (throughput degradation) than the end-to-end restoration, and the end-to-end restoration in turn causes more (less) transient disruption (throughput degradation) than the local restoration. By comparing the values of TDI in Table II.7 and Table II.8, we can see that the transient disruption changes more dramatically under regional multiple node failures than under dispersed multiple node failures. This is because with local restoration, we only need to consider the nodes directly connect to the failed region under regional multiple node failures; however, we need to consider the nodes that have connection to all the failed nodes under dispersed multiple node failures. With the end-to-end restoration, we only need to consider the flow paths enter the failed region under regional multiple node failures; however, we need to consider the flows passing through all the failed nodes under dispersed multiple node failures.

Figure II.9 shows the throughput comparison of the original network and the failed network with global, end-to-end, and local restorations of regional multiple node failures. From this figure, we can see that the average throughput of the global restoration (3.095 $pkts/sec$) is greater than the end-to-end restoration (2.192 $pkts/sec$), and the end-to-end restoration is greater than the local restoration (2.039 $pkts/sec$). By comparing the values of THI in Table II.7 and Table II.8 and the throughput results in Figure II.8 and Figure II.9, we can see that the THI and the average throughput change more dramatically from the global restoration to the end-to-end restoration under regional multiple node failures than under dispersed multiple node failures, while the difference between the end-to-end restoration and the local restoration is not very much. This is because regional multiple node failures greatly affect a part of the flows; furthermore, since the adjacent node of a failed node is probably another failed node in the region, the amount of substituted end-to-end flows and bypass flows are strongly limited.

Restoration Strategy	*TDI*	*THI*
Global Restoration	69.17%	11.89%
End-to-end Restoration	43.98%	30.37%
Local Restoration	29.32%	33.45%

Table II.8: *TDI* and *THI* of Global Restoration, End-to-end Restoration, and Local Restoration under Regional Multiple Node Failures.

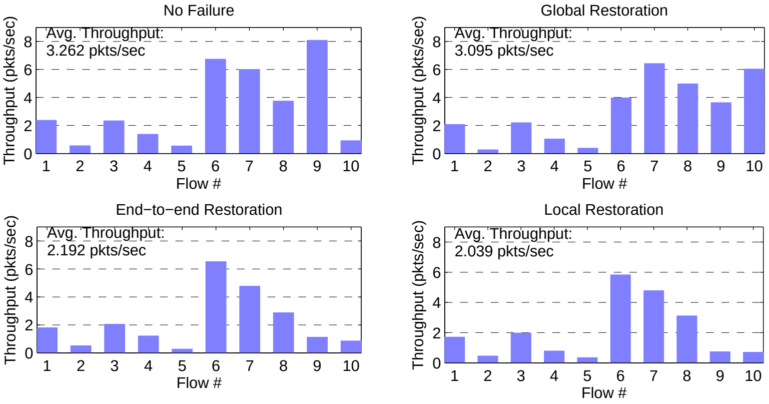

Figure II.9: Throughput Comparison of No Failure, Global Restoration, End-to-end Restoration, and Local Restoration under Regional Multiple Node Failures.

Performance Evaluation under Jamming Attacks

This section evaluates the performance of our optimal network restoration strategies under different network and jamming attack scenarios.

Simulation Setup

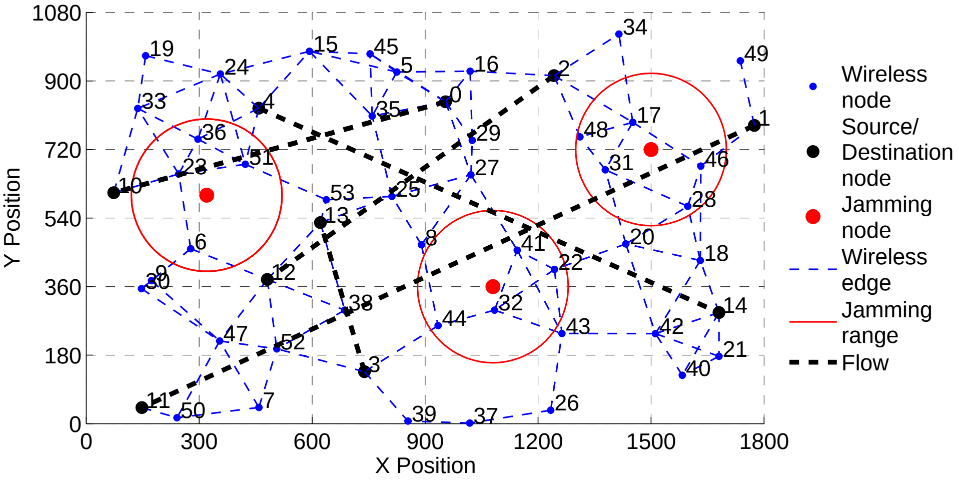

Figure II.10: Example Multi-hop Wireless Network.

In the simulated multi-hop wireless network, 54 wireless nodes are randomly deployed over a 1800×1080 m^2 region. Each node has a transmission range of 250 m and an interference range of 250 m. The channel capacity $\phi_c(c \in C)$ is set as 1 $Mbps$. We have 3 randomly distributed jamming nodes in the network, each of which has a jamming range of 100 m or 200 m. The traffic generating rates of the jammers are from 0.2 $Mbps$ to 0.8 $Mbps$. The simulated network topology is shown in Figure II.10. There are 5 flows in the network with randomly selected sources (node number 0-4) and destinations (node number 10-14). All the flows have the same traffic demand of 1 $Mbps$.

We evaluate the performance of the global restoration and local restoration under two scenarios:

- *Single channel*, where all the network nodes and jamming nodes use the same channel.
- *Multiple channels*, where all the network nodes and jamming nodes use multiple channels and $|C| = 5$. Each network node is equipped with multiple radios. Jammers are able to send jamming traffic over all the channels.

Simulation Results

Single Channel

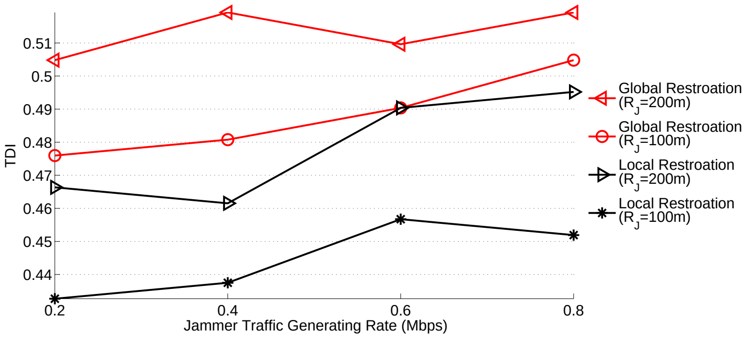

Figure II.11: TDI of Global Restoration and Local Restoration under Single Channel Scenario.

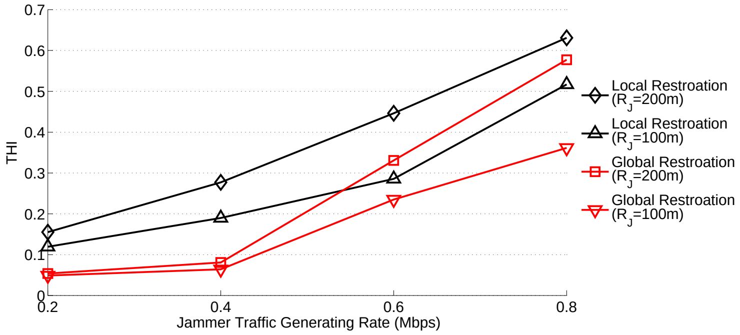

Figure II.12: THI of Global Restoration and Local Restoration under Single Channel Scenario.

We first calculate the values of TDI and THI of global restoration and local restoration under single channel scenario with various jamming traffic generating rates and various jamming ranges. The simulation results of TDI and THI are shown in Figure II.11 and II.12, respectively.

Multiple Channels

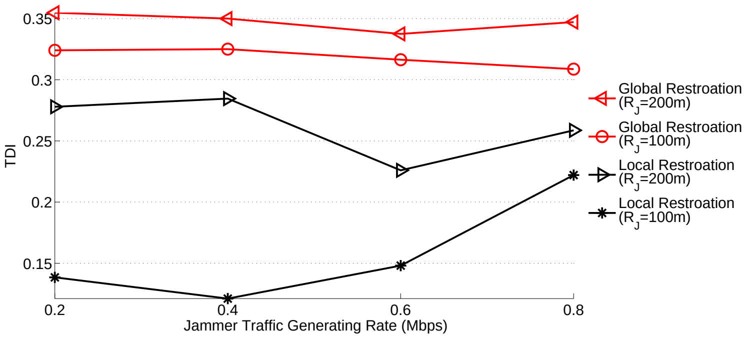

Figure II.13: TDI of Global Restoration and Local Restoration under 5-Channel-3-Radio Scenario using Dynamic Channel Assignment.

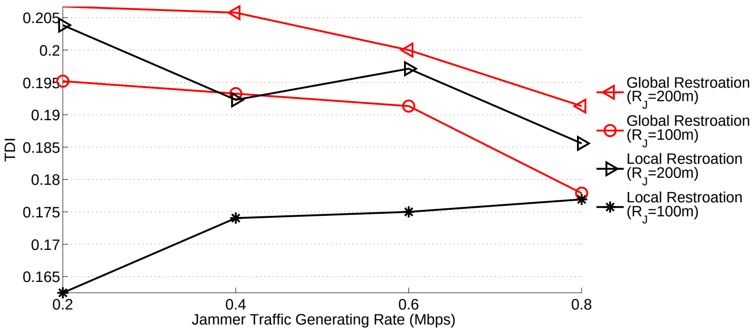

Figure II.14: TDI of Global Restoration and Local Restoration under 5-Channel-3-Radio Scenario using Static Channel Assignment.

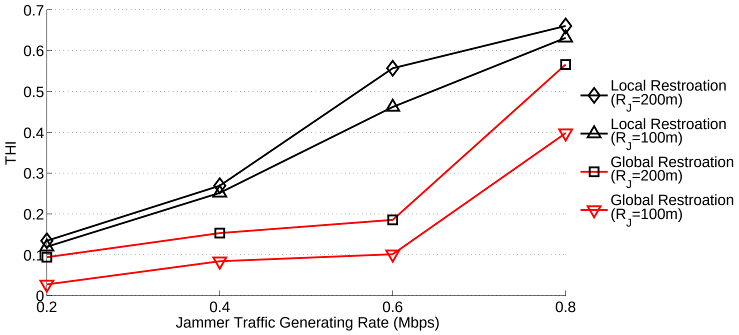

Figure II.15: THI of Global Restoration and Local Restoration under 5-Channel-3-Radio Scenario using Dynamic Channel Assignment.

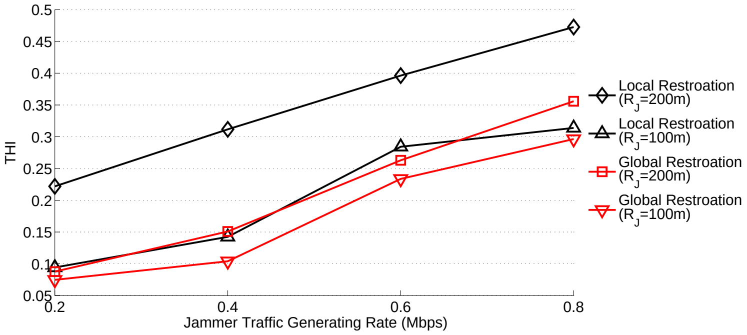

Figure II.16: THI of Global Restoration and Local Restoration under 5-Channel-3-Radio Scenario using Static Channel Assignment.

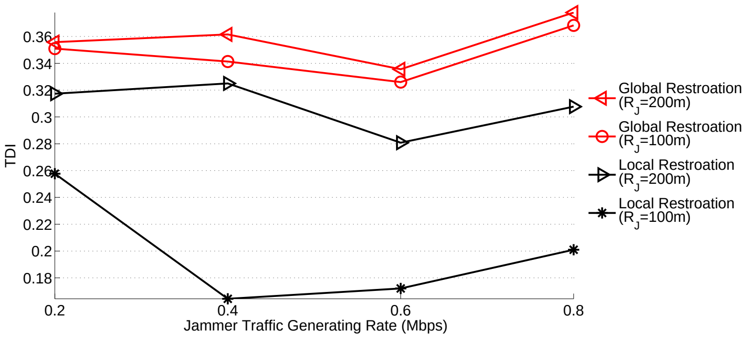

Figure II.17: TDI of Global Restoration and Local Restoration under 5-Channel-5-Radio Scenario using Dynamic Channel Assignment.

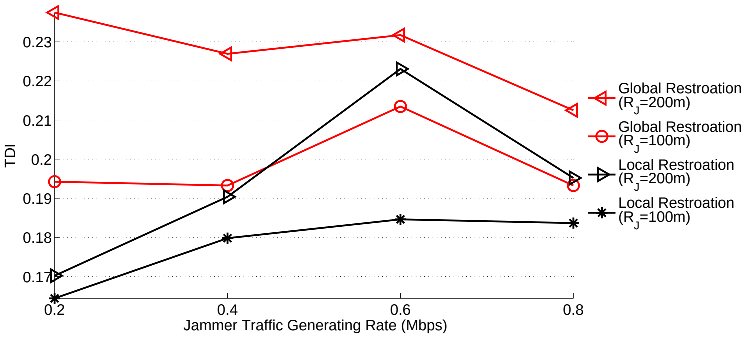

Figure II.18: TDI of Global Restoration and Local Restoration under 5-Channel-5-Radio Scenario using Static Channel Assignment.

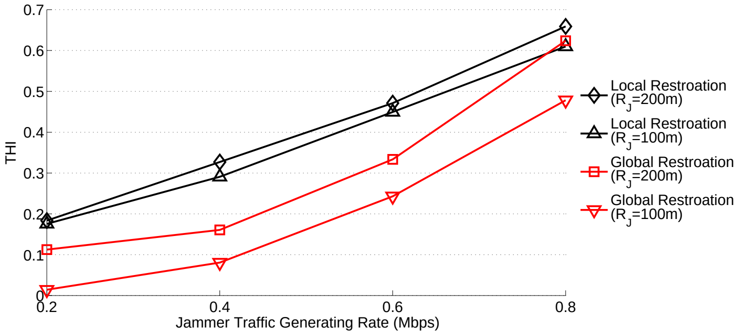

Figure II.19: THI of Global Restoration and Local Restoration under 5-Channel-5-Radio Scenario using Dynamic Channel Assignment.

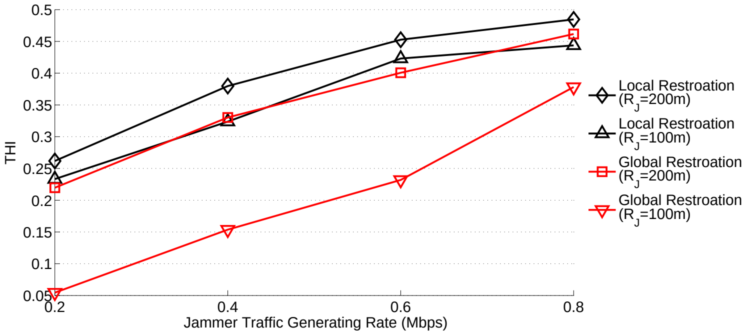

Figure II.20: THI of Global Restoration and Local Restoration under 5-Channel-5-Radio Scenario using Static Channel Assignment.

We then calculate the values of TDI and THI of global restoration and local restoration under multiple channels scenario with various jamming traffic generating rates and various jamming ranges, using both dynamic channel assignment and static channel assignment. The results of 5-channel-3-radio scenario are shown in Figure II.13, II.14, II.15, and II.16, and the results of 5-channel-5-radio scenario are shown in Figure II.17, II.18, II.19, and II.20, respectively.

From these figures, we can see that the transient disruption of the global restoration is much higher than that of the local restoration; however, the throughput degradation is lower in the global restoration. And when the jamming range is $200m$, the values of both TDI and THI are higher than when the jamming range is $100m$. From these figures, we can also see that the transient disruption of both the global and local restorations is not changed too much as the traffic generating rate of the jammers increased. This is because no matter how fast the jamming rate is, the routing table entries always need to be modified; and the number of the modifications is not affected by the jamming rate. The throughput degradation of both the global and local restorations increases as the traffic generating rate of the jammers increased.

Comparison of TDI and THI under Various Scenarios

TDI	Global Restoration ($R_J 200m$)	Global Restoration ($R_J 100m$)	Local Restoration ($R_J 200m$)	Local Restoration ($R_J 100m$)
5-channel 3-radio	34.74%	31.85%	26.18%	15.75%
5-channel 5-radio	35.77%	34.66%	30.77%	19.88%

Table II.9: Average TDI Comparison using Dynamic Channel Assignment.

TDI	Global Restoration ($R_J 200m$)	Global Restoration ($R_J 100m$)	Local Restoration ($R_J 200m$)	Local Restoration ($R_J 100m$)
5-channel 3-radio	21.10%	18.94%	19.47%	17.21%
5-channel 5-radio	22.72%	19.86%	19.87%	17.81%

Table II.10: Average TDI Comparison using Static Channel Assignment.

We further compare the average values of TDI and THI over all the jamming traffic sending rates under multiple channels scenarios with different jamming ranges R_J. The results are shown in Table II.9, II.10, II.11, and II.12, respectively. Comparing the results in Table II.9 and II.10 between the first line and the second line, we can see that the transient disruption under 5-channel-3-radio scenario is lower than that of 5-channel-5-radio scenario; this is because the network is more complicated under 5-channel-5-radio scenario. Comparing the results in Table II.9 and II.10, we can see that the transient disruption under static channel assignment is lower than that under dynamic channel assignment; this is because dynamic channel assignment involves channel switching overhead.

THI	Global Restoration ($R_J 100m$)	Global Restoration ($R_J 200m$)	Local Restoration ($R_J 100m$)	Local Restoration ($R_J 200m$)
5-channel 3-radio	15.26%	24.96%	36.60%	40.51%
5-channel 5-radio	20.40%	30.76%	38.14%	41.01%

Table II.11: Average THI Comparison using Dynamic Channel Assignment.

Comparing the results in Table II.11 and II.12 between the first line and the second line, we can see that the throughput degradation of the 5-channel-3-radio scenario is lower than that of the 5-channel-5-radio scenario. Based on

THI	Global Restoration ($R_J 100m$)	Global Restoration ($R_J 200m$)	Local Restoration ($R_J 100m$)	Local Restoration ($R_J 200m$)
5-channel 3-radio	17.71%	22.43%	20.86%	35.06%
5-channel 5-radio	20.45%	36.30%	35.59%	39.47%

Table II.12: Average THI Comparison using Static Channel Assignment.

the necessary conditions of channel assignment and scheduling, the throughput performance are based on two factors: the number of radios and the available network capacity besides jamming. Under the 5-channel-3-radio scenario, the number of radios also limits the achievable throughout, so the impact of jamming is not very dominant.

Comparison of λ under Various Scenarios

As we mentioned earlier, the result derived from the linear programming formulation of the network restoration problem gives an upper bound on the achievable network throughput. And dynamic channel assignment provides the maximum flexibility in channel assignment and scheduling, so it achieves higher network throughput than static channel assignment. Previously, we have compared the performance of our optimal network restoration strategies under different network and jamming attack scenarios. In order to investigate the relationship among dynamic channel assignment, static channel assignment and the performance upper bound, we compare the value of the minimum flow throughput scaling factor λ under various scenarios.

λ	single channel single radio	5-channel 3-radio	5-channel 5-radio
Linear Programming	0.2555	1.0	1.2777
Dynamic Channel Assignment	0.0979	0.3189	0.3811
Static Channel Assignment	0.0786	0.1463	0.1677

Table II.13: Comparison of λ without Jamming Attacks.

The original values of λ using linear programming, dynamic channel assignment, and static channel assignment for single channel, 5-channel-3-radio, and 5-channel-5-radio scenarios without jamming attacks are shown in Table II.13. Comparing the results from top to bottom, we can see that under all the network configurations, the values of λ using linear programming (performance upper bound) are higher than those using dynamic channel assignment, which in turn are better than those using static channel assignment. The performance gap is caused by the feasibility of scheduling under two different radio operation models (i.e., dynamically switching and fixed binding).

Table II.14, II.15, and II.16 further compare the average values of λ over all the jamming traffic sending rates using different restoration strategies under different network configuration scenarios. Comparing the values of λ

λ	single channel single radio	5-channel 3-radio	5-channel 5-radio
Global Restoration ($R_J = 100m$)	0.1984	0.8750	0.9922
Global Restoration ($R_J = 200m$)	0.1738	0.7813	0.8690
Local Restoration ($R_J = 100m$)	0.1644	0.5977	0.6389
Local Restoration ($R_J = 200m$)	0.1385	0.5695	0.6389

Table II.14: Average λ Comparison using Linear Programming.

λ	single channel single radio	5-channel 3-radio	5-channel 5-radio
Global Restoration ($R_J = 100m$)	0.0806	0.2702	0.3033
Global Restoration ($R_J = 200m$)	0.0724	0.2393	0.2639
Local Restoration ($R_J = 100m$)	0.0707	0.2022	0.2357
Local Restoration ($R_J = 200m$)	0.0610	0.1897	0.2248

Table II.15: Average λ Comparison using Dynamic Channel Assignment.

λ	single channel single radio	5-channel 3-radio	5-channel 5-radio
Global Restoration ($R_J = 100m$)		0.1204	0.1334
Global Restoration ($R_J = 200m$)		0.1150	0.1085
Local Restoration ($R_J = 100m$)		0.1158	0.1080
Local Restoration ($R_J = 200m$)		0.0950	0.1015

Table II.16: Average λ Comparison using Static Channel Assignment.

shown in the same cell in Table II.14, II.15, and II.16, we can see that the global restoration scheme performs better than the local restoration scheme under all scenarios[6]. We also observe that when the number of radios increases from 3 to 5 in a 5-channel network, the network restoration scheme with dynamic channel assignment shows improved performance. This shows that the dynamic channel assignment could well explore the additional radio resources in the restoration. Yet in the case of static channel assignment, increasing the number of radios may not bring much additional performance gain due to its greedy channel assignment strategy which may use a sub-optimal assignment scheme.

[6] Under single channel single radio scenario, there is no need to consider the static channel assignment.

CHAPTER III

MINIMUM DISRUPTION SERVICE COMPOSITION AND RESTORATION FOR MULTI-HOP WIRELESS MOBILE NETWORKS

Introduction

Multi-hop wireless mobile networks, such as mobile ad hoc networks [56], are self-organized networks formed dynamically through collaboration among mobile nodes, which can be applied to a wide range of application scenarios. These diverse application needs have fueled an increasing demand for new functionalities and services. To meet these demands, component-based software development [62] has been used to ensure the flexibility and maintainability of software systems. *Service composition* [29, 71, 12] 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. The importance of service composition has been widely recognized due to its high flexibility in allowing development and deployment of customized applications from primitive services.

There is an extensive literature on service composition techniques over wireline networks. For example, [71, 49, 77] focus on finding a service path over wireline networks that satisfies various service requirements. Likewise, [55, 69] 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 wireless mobile networks since intermittent link connectivity and dynamic network topology caused by node mobility is not considered.

To address this open issue, in this chapter, we study the service composition schemes to cope with node mobility in multi-hop wireless mobile 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 restoration strategies that enable highly reliable service delivery and incur the minimum disruptions to end users in multi-hop wireless mobile 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 wireless mobile routing algorithms [75, 64] and component deployment mechanisms [50] 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 restoration design: (1) it does not account for service repair and restoration 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 a wireless mobile 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 restoration 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 restoration strategies.* Node mobility is a major cause of service failures in wireless mobile 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, we present a new service composition and restoration framework for multi-hop wireless mobile 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 wireless mobile networks with known mobility plan, we formulate the problem of *minimum-disruption service composition and restoration* (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 wireless mobile 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.

The main contributions of this chapter are three folds. First, it creates a theoretical framework for service composition and restoration strategies for wireless mobile 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.

Related Work

Our work is positioned in the overlapping area of service composition for service-oriented networks and reliable network routing in wireless mobile 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 [62, 21]. 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 service performance metrics, such as the delay, bandwidth, and reliability. For example, Xu et al. [71] find service paths to optimize the end-to-end resource availability with controlled system overhead. In [49, 77], multiple service criteria are aggregated for service path selection and optimization. The scalable service composition is investigated in [36, 28] for large scale systems, by employing distributed and hierarchical routing techniques.

- **Failure restoration in service disruptions**. Raman et al. [55] presents an architecture for quick service path restoration using service replicas and tuning the process of failure detection, focusing mainly on architectural discussions. Li et al. [69] 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 wireless mobile networks. For example, [75] presents a reliability framework for wireless mobile network routing, which uses the position and trajectory information of the so-called reliable nodes (in terms of robust and secure) to build reliable path. Likewise, [64, 34, 63, 58, 57] 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, [50] 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 if 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 [46, 59] to provide continuous service despite of network partition. Moreover, [17] 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 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 restoration, which is different from the work of [17] 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 [46, 59] could be applied.

System Model

This section provides our network and service model.

Network Model

We consider a multi-hop wireless mobile 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 multi-hop wireless mobile 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

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

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 III.1 shows an example multi-hop wireless mobile network based on the terms defined above.

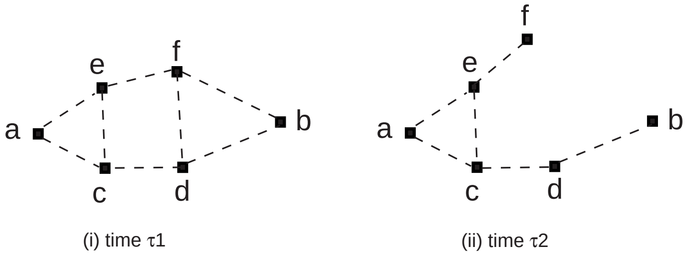

Figure III.1: Example Multi-hop Wireless Mobile Network.

Two snapshots of the network topologies at time instances τ_1 and τ_2 are shown in Figure III.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 $G(\tau_2)$.

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 [62]. 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*.

We focus 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*.

In a wireless mobile network, each service component s_k can be replicated at multiple nodes to improve the service availability [65]. 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 III.2 shows an example of service deployment and service composition.

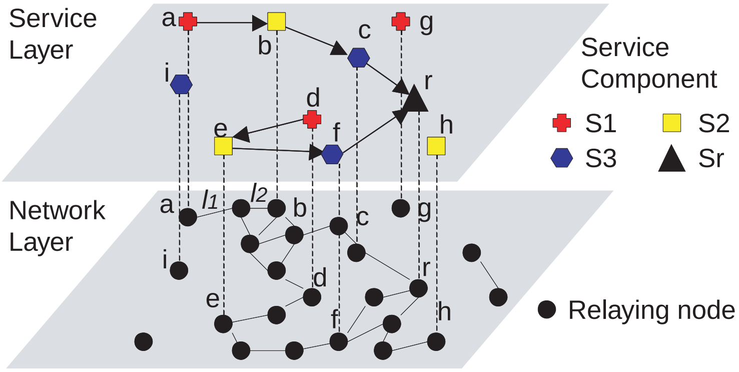

Figure III.2: Example Service Deployment and Service Composition.

A service link is an overlay link that may consist of several wireless links in the network, *i.e.*, a network path. In Figure III.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 service requirements. To focus the discussion on the impact of service failures caused by node mobility, we consider a simple metric—called the *service link length*—that is defined as 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 III.1 summarizes the notations used in this chapter.

Service Composition and Restoration Framework

This section describes our service composition and restoration framework for wireless mobile networks.

Service Composition

Service composition refers to the process of finding a service path that satisfies designated service requirements in the network.

As shown in Figure III.3, service composition in a wireless mobile network involves the following two inherently related processes:

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
$\mathcal{S} = (s_1 \to s_2 \to ... \to s_r)$	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 restoration scheme
$\Pi = (\pi(t_1), \pi(t_2), ..., \pi(t_l))$	service composition and restoration 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
$N_{\mathcal{P} \to \mathcal{P}'}$	number of link substitutions from path $\mathcal{P}$ to path $\mathcal{P}'$
$N_{\pi_{\mathcal{S}} \to \pi'_{\mathcal{S}}}$	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
$\tilde{d}_{n \to n'}(t + \Delta t)$	predicted distance of a service link $(n \to n')$
$L_{n \to n'}$	lifetime of service link $(n \to n')$

Table III.1: Key Notations.

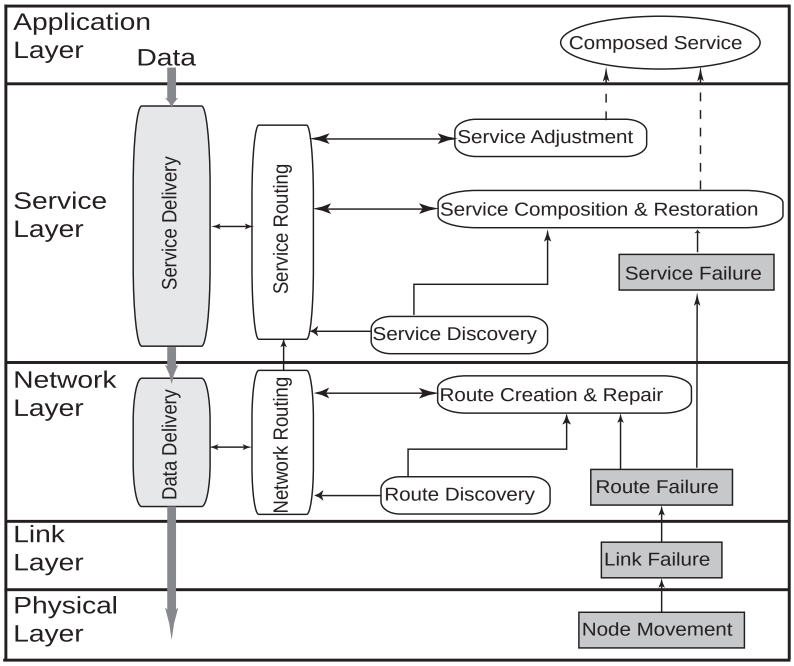

Figure III.3: A Service Composition and Restoration Framework in a Wireless Mobile Network.

- *Service routing*, which selects the service components (out of many replicas) for the service path. This routing process relies on service component discovery [42, 41] to find the candidate service components, then selects the appropriate ones to compose a service path that satisfies the service 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 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 wireless mobile network, service failures may occur for multiple reasons. For example, end-to-end service requirements may be violated due to network overload; service links may break due to failure of the underlying wireless communication path. This chapter focuses on *service failures caused by node mobility.*

Service Restoration

To sustain service delivery, the service path must be repaired. This repair process essentially *recomposes the service path* and is called *service restoration.* Service restoration 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* (Ad hoc On-Demand Distance Vector Routing) [53].

Similar to service composition, service restoration also involves two processes: (1) *network-level restoration*, which repairs the data path between two components, and (2) *service-level restoration*, which replaces one or more service components. The network-level path repair usually depends on the specific wireless routing protocol used and relies on the route repair mechanism built within the routing protocol. The service-level restoration involves discovery of new components and establishment of a new service path.

Service restoration 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 restoration duration, we prefer a service path that could maximally reuse the current nodes/components. For example, network-level restoration may be attempted first without changing any service components. If this restoration fails, then a service-level restoration is initiated. The limitation with using this service restoration strategy, however, is that the new service path may have a poor service and/or may fail again soon. Alternatively, we may wish to use service-level restoration directly without trying network-level restoration. 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 service 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 restoration, however, such changes can disrupt the service, even though they improve the new path's reliability and quality.

Theoretical Framework

A fundamental research challenge for service restoration is *how to best tradeoff the time and overhead involved in service restoration 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 restoration 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.

Service Disruption Model

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

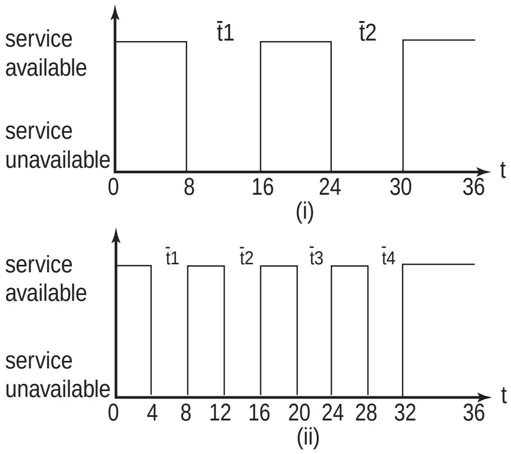

Figure III.4: Example Service Disruption Processes.

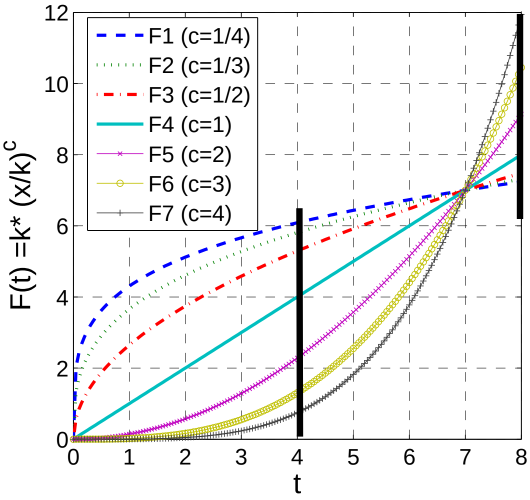

Figure III.5: Example Disruption Penalty Functions (k=7 is the intersection point of all the lines).

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 III.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 failure time and restoration 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 restoration decisions. To establish a theoretical framework that provides realistic insight to implementation of service composition and restoration 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 III.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:

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

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 III.4 using the different failure penalty functions $F(\bar{t})$ shown in Figure III.5. The results are summarized in Table III.2.

Table III.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 with disruptions with long duration. For a linear disruption

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 III.2: Disruption Indices Under Different Penalty Functions.

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 restoration and service-level restoration, respectively.

Estimation for Network-level Restoration

For network-level restoration, the service components remain the same, *i.e.*, we only need to repair the network path that connects them. A typical network-level restoration process in repairing a network path in wireless mobile networks [53] 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 restoration. Formally, let $\mathcal{P}$ and $\mathcal{P}'$ be the paths before and after restoration. 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

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

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

Estimation for Service-level Restoration

A service-level restoration 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 restoration thus takes much more time than network-level restoration. Similar to network-level restoration, the duration of service-level restoration depends largely on

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

the searching/replacing scope of the service components. We can therefore use the number of substituted components to estimate its restoration duration. Formally, let $\pi_{\mathcal{S}}$ and $\pi'_{\mathcal{S}}$ be the service routing schemes before and after restoration. 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

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

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 restoration 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

$$\begin{aligned} \bar{t}_k &= \beta \times N_{\pi(t_v) \to \pi(t_{v+1})} & \text{(III.4)} \\ &= \beta \times (N^{\mathcal{N}}_{\pi(t_v) \to \pi(t_{v+1})} + \alpha N^{\mathcal{S}}_{\pi(t_v) \to \pi(t_{v+1})}) & \text{(III.5)} \end{aligned}$$

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 restoration (if any) and the number of substituted components in service-level restoration (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 $\tilde{D}$:

$$
\begin{aligned}
\tilde{D} &= \frac{1}{T}\sum_{v=1}^{l-1} F(\beta \times N_{\pi(t_v)\to\pi(t_{v+1})}) && \text{(III.6)} \\
&= \frac{1}{T}\sum_{v=1}^{l-1} F(\beta \times (N^{\mathcal{N}}_{\pi(t_v)\to\pi(t_{v+1})} + \alpha N^{\mathcal{S}}_{\pi(t_v)\to\pi(t_{v+1})})) && \text{(III.7)}
\end{aligned}
$$

MDSCR Problem Formulation

Based on the definition of disruption index, we now formulate the *minimum disruptive service composition and restoration* (MDSCR) problem. First, we define a service composition and restoration policy as a sequence of service composition schemes:

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

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 restoration schemes $\pi(t_v) \to \pi(t_{v+1})$, $v = 1, ..., l-1$.

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 III as $\tilde{D}(\Pi)$:

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

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,

$$\text{MDSCR}: \quad \textbf{minimize} \quad \tilde{D}(\Pi) \tag{III.10}$$

$$\Pi \in \Phi(\mathcal{G}_{\mathcal{T}'}) \tag{III.11}$$

At this point, we have established a theoretical framework for the MDSCR problem in multi-hop wireless mobile 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 restoration. We next study the optimal MDSCR solution under a known mobility plan and derive its analytical properties. Based on these analytical insights, we then present the location-aided MDSCR heuristic algorithm based on service link lifetime prediction.

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

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

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

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

When the mobility plan of the network is known, the equation shown above could be used to give the optimal solution via standard dynamic programming techniques [15]. 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. (III.13) gives the optimal service restoration scheme (minimum disruption service restoration) that changes the service composition from $\pi(t_w)$ to $\pi(t_{w+1})$.

Analysis

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

Reactive Restoration

The first property of an optimal solution is the *reactive adjustment and restoration 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 [35].

Reactive Service-level Restoration

For an optimal solution, the service-level restoration is invoked if and only if the network-level restoration 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 $\Pi^*_{\mathcal{S}} = (\pi^*_{\mathcal{S}}(t^s_1), ..., \pi^*_{\mathcal{S}}(t^s_g))$. 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}, *i.e.*, there exists a service link in $\pi^*_{\mathcal{S}}(t^s_w)$ which has no feasible network path in $\mathcal{G}(\tau_i)$, when $\alpha \gg 1$.

The proof of this theorem is given in [35].

MDSCR Heuristic Algorithm

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

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

$$\tilde{D}_{\mathcal{S}} = \frac{1}{T}\sum_{v=1}^{g-1} F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_v)\rightarrow\pi_{\mathcal{S}}(t^s_{v+1})}) \qquad \text{(III.14)}$$

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

$$\mathcal{J}(\pi_{\mathcal{S}}(t^s_1)) = \min_{\Pi_{\mathcal{S}}\in\Phi(\mathcal{G}_{\mathcal{T}'})} \frac{1}{T}\sum_{v=1}^{g-1} F(\beta\alpha N^{\mathcal{S}}_{\pi_{\mathcal{S}}(t^s_v)\rightarrow\pi_{\mathcal{S}}(t^s_{v+1})}) \qquad \text{(III.15)}$$

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

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

The network-level MDSCR is the secondary problem. It tries to minimize the disruption index caused by network-level restoration 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.

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

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 wireless network routing protocols for the network-level MDSCR.

One-step Lookahead Approximation

Finding the solution to the service-level MDSCR problem is still a challenging issue for wireless networks with uncertain mobility plans since complete knowledge of future network topologies is needed. The service restoration decision at t_{w+1}^s requires the knowledge of network topology after this time instance to calculate the future disruption index $\mathcal{J}(\pi_{\mathcal{S}}(t^s_{w+1}))$. 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 $(s_k[n_k] \to s_{k+1}[n_{k+1}])$. 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

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

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

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

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

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

[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 that supports this service link is larger than service link length requirement H.

In Eq. (III.20), the first term characterizes the restoration 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. (III.20) balances the tradeoff between these two factors faced by service restoration.

Lifetime Prediction

Another problem with deriving a practical MDSCR solution for Eq. (III.19) and Eq. (III.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 $V_n(t)$, $V_{n'}(t)$ be their velocities at time t. The predicted distance of service link $(n \to n')$ after time interval Δt is then given as follows:

[4] We assume that the mobile nodes in the network are distributed roughly homogeneously.

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

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 III.5 in Section III. We plot the relation between the service link lifetime and its predicted distance in Figure III.6.

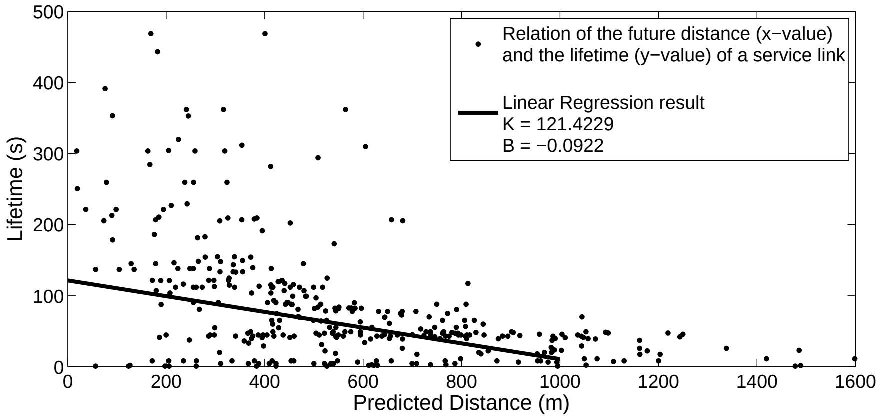

Figure III.6: Lifetime Prediction.

The black dots in Figure III.6 describe 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:

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

```
     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 $L_{n_k \rightarrow n_{k+1}}$.
1.2  Find the service routing scheme $\pi_{\mathcal{S}}$ that minimizes Eq. (III.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 III.3: Minimum Disruption Service Composition Algorithm.

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

In the simulation study, 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.

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 services [74] for node location and velocity information, as well as service discovery services [41].

```
    **Algorithm II**: Minimum Disruption Service Restoration
    //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 restoration**
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
        //Network-level restoration succeeds
        Return the path.
    //Network-level restoration fails, try service-level restoration
2   **Top tier: service-level restoration**
    //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 $L_{n_k \rightarrow n_{k+1}}$.
    //Then perform network routing
2.2 Find the service routing scheme $\pi_{\mathcal{S}}(t_{w+1}^s)$ that minimizes Eq. (III.20)
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 III.4: Minimum Disruption Service Restoration.

Table III.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 III.4 gives the minimum disruption service restoration algorithm. This algorithm also has two tiers: bottom and top. The bottom tier is the network-level restoration, which is triggered by the failure of a wireless link on the current service path. If the network-level restoration succeeds, the algorithm returns successfully. If it fails, however, then the service-level restoration in the top tier will be triggered. The service-level restoration first finds the new service components, which balances the restoration duration and the sustainability for the new service link. It then performs the network path routing between the new service components.

Simulation Study

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

Simulation Setup

We conducted the simulations using ns-2 [2]. In our simulated multi-hop wireless mobile network, 50 nodes are randomly deployed over a $2,000 \times 1,000$ m^2 region. Each node has a transmission range of 250 m. Node mobility follows the random waypoint model with a *maximum speed* (default value is 10 m/s) and a *pause time* (default value is 10 s).

The service discovery is simulated based on the results presented in [43] 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 III.5.

We compare the performance of our MDSCR algorithm with the *shortest path service composition and restoration* (SPSCR) algorithm [9, 76] and the *random selection service composition and restoration* (RSSCR) algorithm. The shortest path routing algorithm [67] is a common multi-hop network routing algorithm that chooses the path with the smallest hop number. The

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 $\lvert\mathcal{N}_k\rvert$	8
service link length requirement H	3
α	10
β	1
disruption penalty function	$F(\bar{t}) = \bar{t}$

Table III.5: Default Simulation Parameters.

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.

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 2×10^5 seconds. Since the experiment time is extremely long, it can reflect a general network topology.

Figures III.7 and III.8 show the results of disruption index and throughput for the MDSCR, SPSCR, and RSSCR algorithms using CBR traffic.

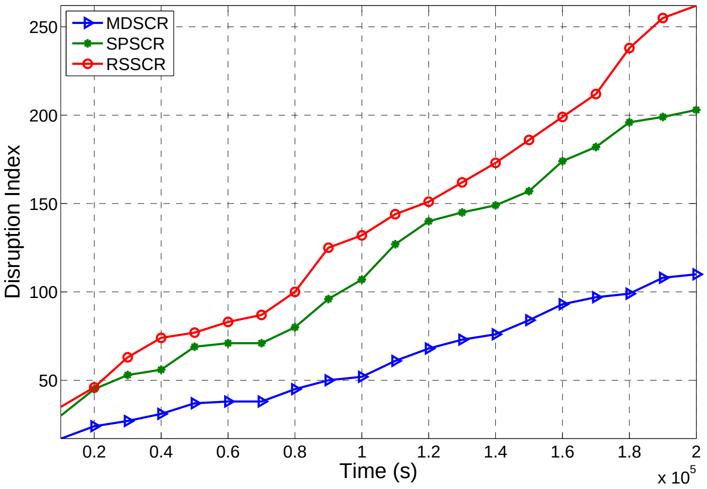

Figure III.7: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 Using CBR Traffic.

From Figure III.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

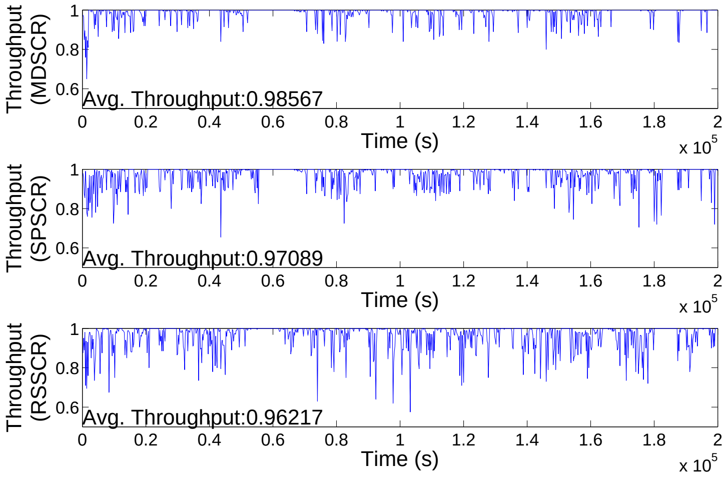

Figure III.8: Throughput for MDSCR, SPSCR, and RSSCR when Service Path Length is 2 Using CBR Traffic.

regard to their frequencies and durations. This result can also be reflected by the instantaneous throughput of the service, which is shown in Figure III.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 service components (as does the MDSCR algorithm).

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 III.9 and III.10 show these results.

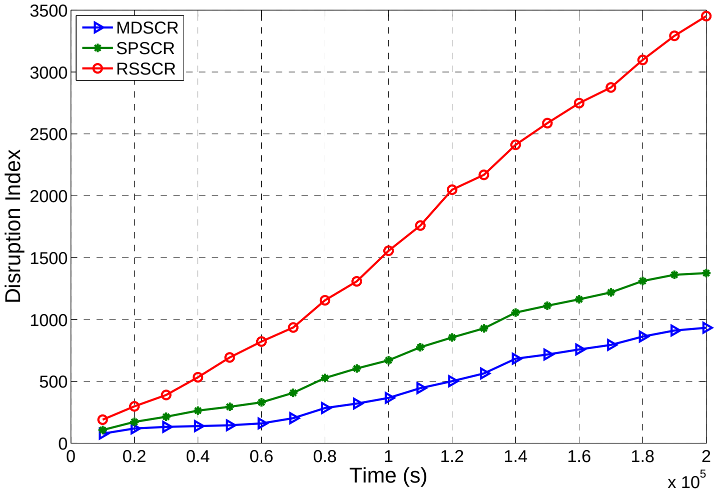

Figure III.9: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 Using CBR Traffic.

Comparing Figure III.9 with Figure III.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 III.10 and III.8 further validates this result. We also observe that the disruption index increases and the throughput decreases when the synthetic service is

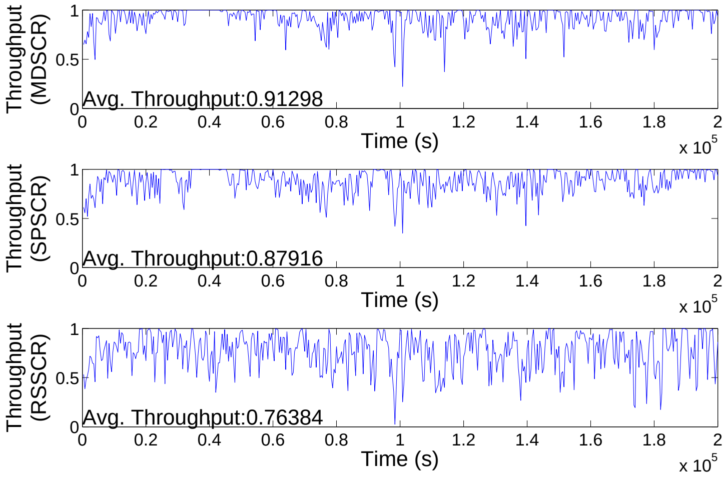

Figure III.10: Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 Using CBR Traffic.

composed of more components (*i.e.*, from 2 to 4), which means there is a higher possibility for the service path to be disrupted.

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 restoration algorithms. Figure III.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 III.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

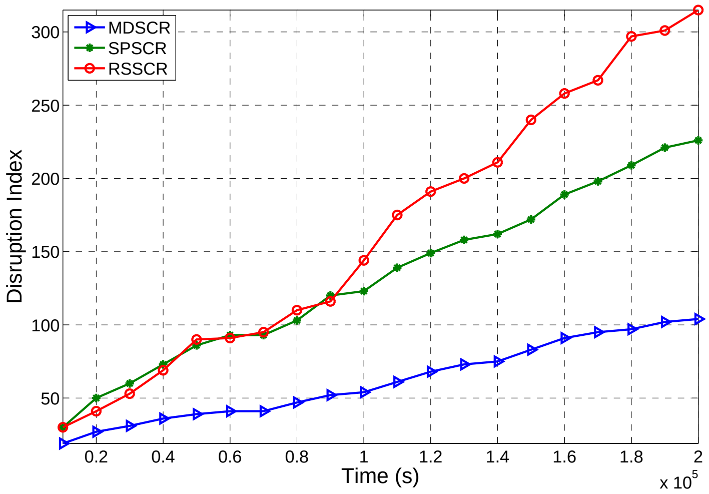

Figure III.11: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using CBR Traffic.

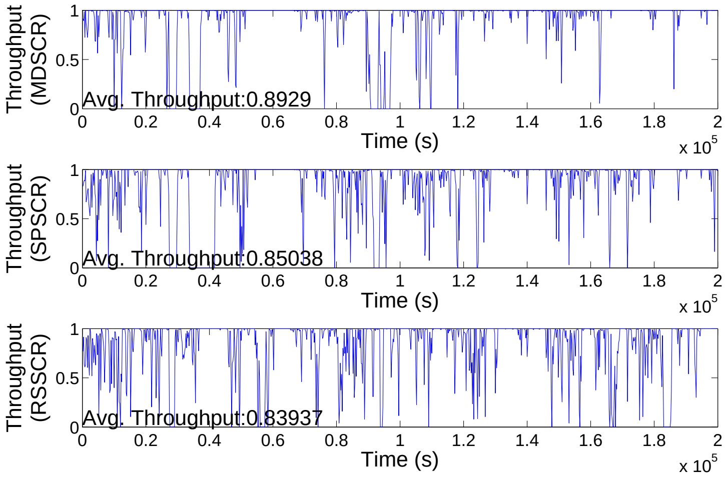

Figure III.12: Throughput for MDSCR, SPSCR, and RSSCR when Service Path Length is 2 and Service Link Length Requirement is 3 Using CBR Traffic.

comparison in Figures III.12 and III.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 III.13 and III.14.

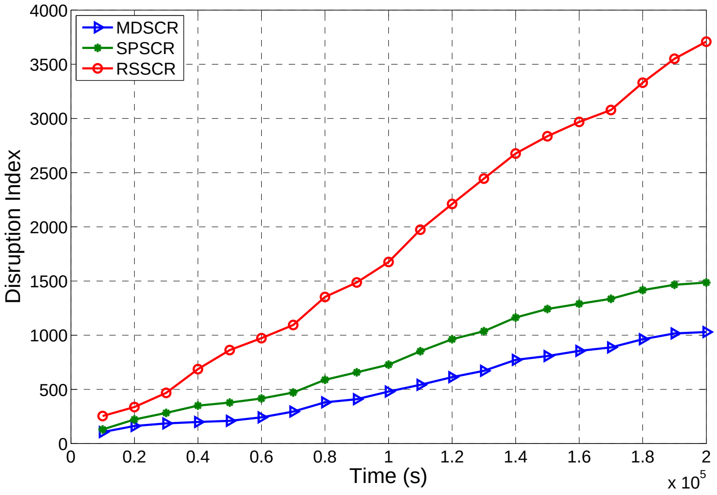

Figure III.13: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 and Service Link Length Requirement is 3 Using CBR Traffic.

By comparing these two figures with Figures III.9 and III.10, we observe that the disruption index increases and throughput decreases with a 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 $\frac{\tilde{D}_{SPSCR}-\tilde{D}_{MDSCR}}{\tilde{D}_{SPSCR}}$,

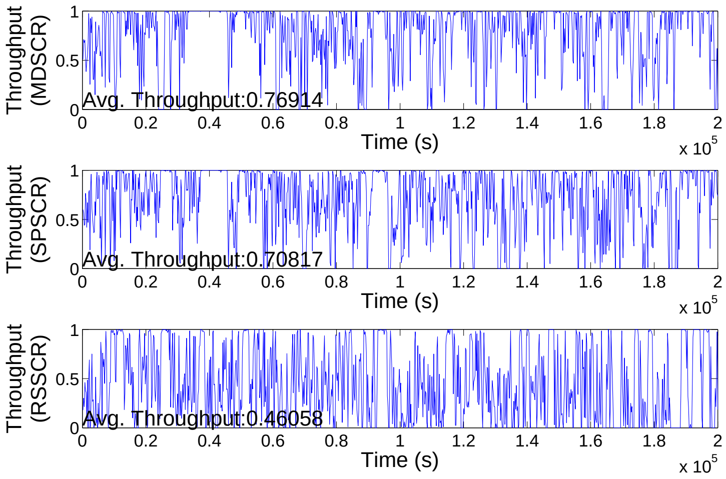

Figure III.14: Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 4 and Service Link Length Requirement is 3 Using CBR Traffic.

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.

We run simulations under different values of H $(1, ..., 5)$ and plot the average improvement ratios in Figure III.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.

Impact of Traffic Type

The performance of service composition and restoration 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 2×10^4 seconds. Figures III.15 and III.16 show the results of disruption index and throughput when service path length is 2 with no service link length requirement. 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 cannot distinguish the queueing loss from the packet loss caused by link failures, which is a common problem of TCP over wireless networks [37].

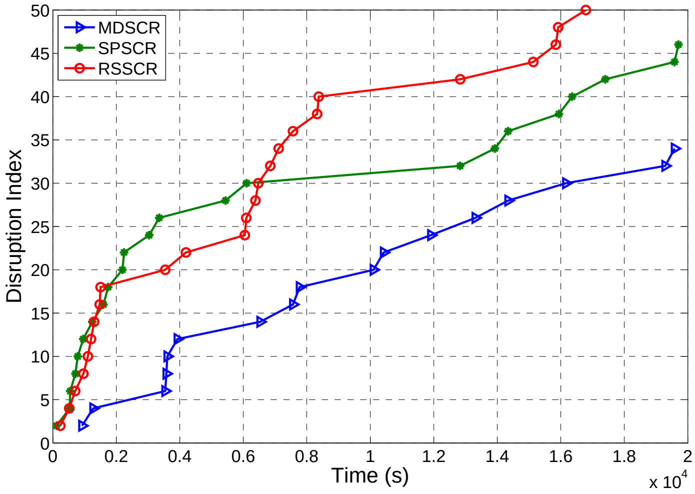

Figure III.15: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 Using TCP Traffic.

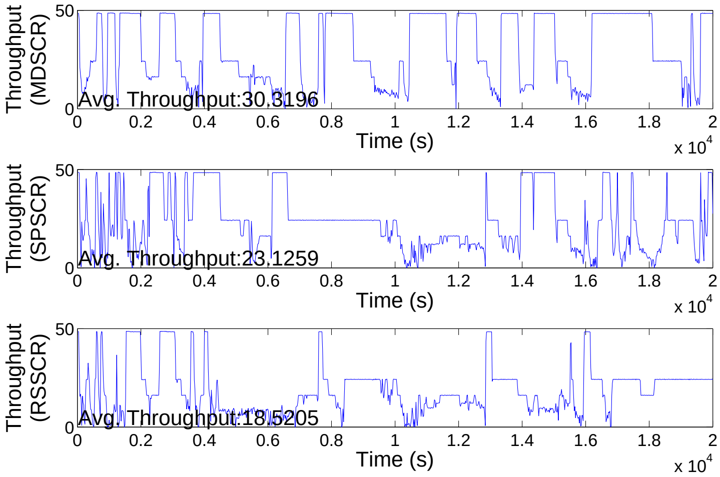

Figure III.16: Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 Using TCP Traffic.

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 III.17 and III.18.

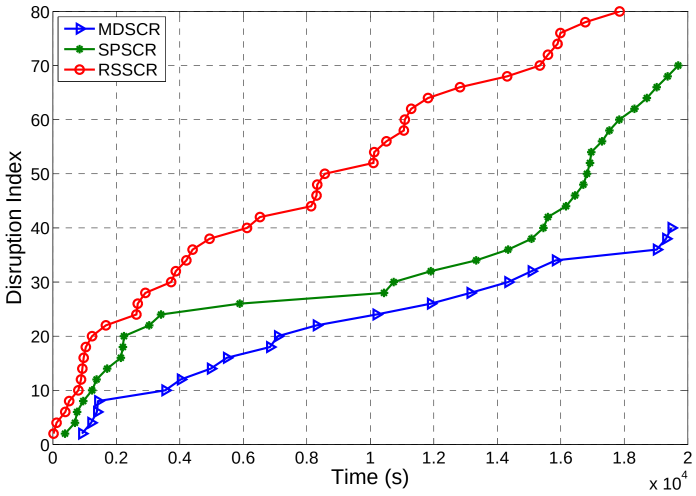

Figure III.17: Disruption Index for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using TCP Traffic.

Comparing these two figures with Figures III.15 and III.16 shows the same result with regard to the disruption index discussed in "Impact of Service Link Length Requirement H", *i.e.*, the disruption index increases with more restricted service link length requirement. 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.

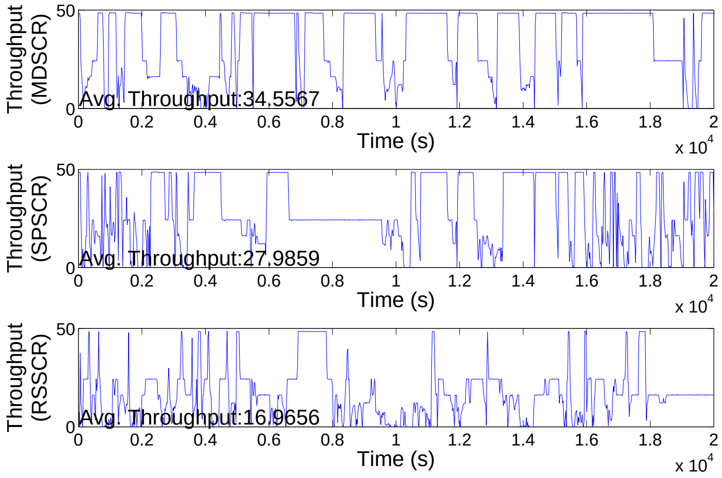

Figure III.18: Throughput for MDSCR, SPSCR, and RSSCR When Service Path Length is 2 and Service Link Length Requirement is 3 Using TCP Traffic.

Impact of Number of Component Replicas

The performance of service composition and restoration 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 $2,000$ seconds in Figure III.20.

Figure III.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.

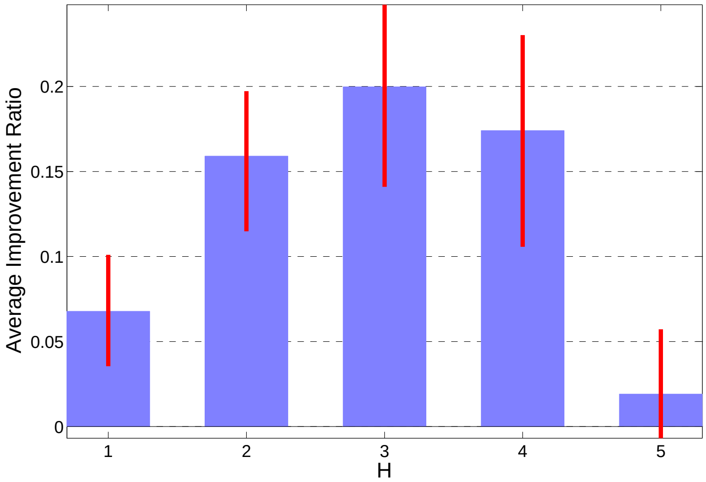

Figure III.19: Impact of Service Link Length Requirement H on Improvement Ratio.

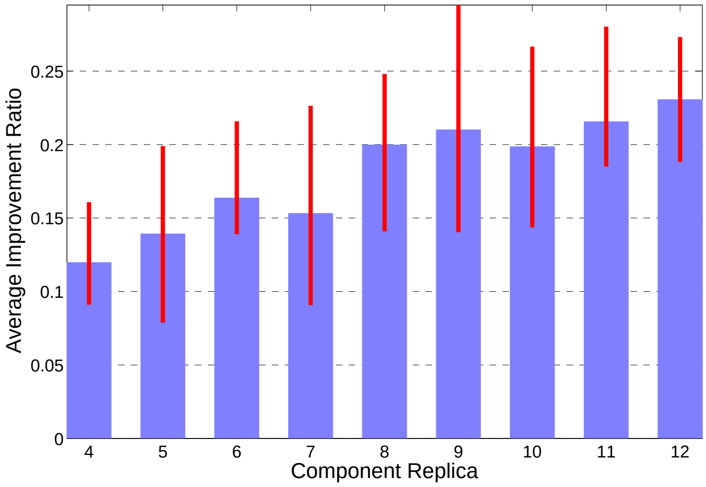

Figure III.20: Impact of Number of Component Replicas on Improvement Ratio.

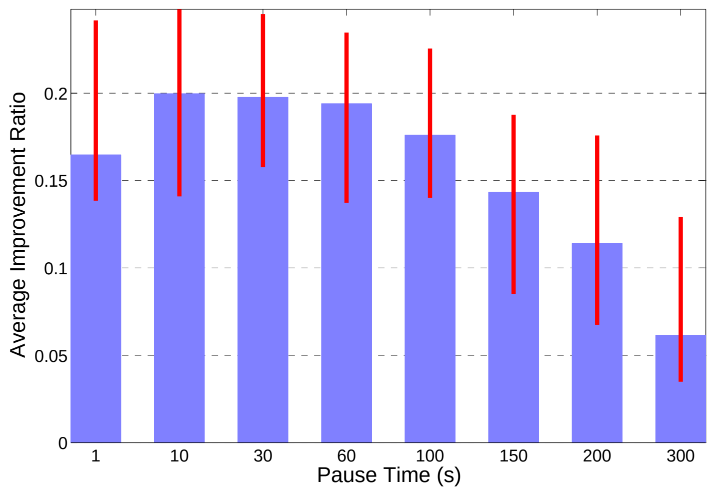

Figure III.21: Impact of Pause Time on Improvement Ratio.

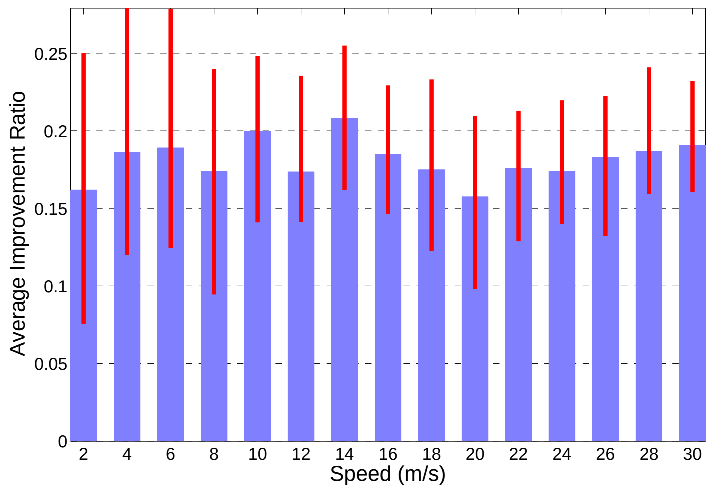

Figure III.22: Impact of Node Speed on Improvement Ratio.

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 1 s, 10 s, 30 s, 60 s, 100 s, 150 s, 200 s, 300 s and maximum node speeds of 2 m/s, 4 m/s, 6 m/s, ..., 30 m/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 $2,000$ seconds.

Figures III.21 and III.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 10 s to 100 s, which represents a medium-mobility environment. In this mobility environment, the service link lifetime prediction method provides the best prediction results.

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 III.23 compares the improvement ratios under linear, concave, and convex functions.

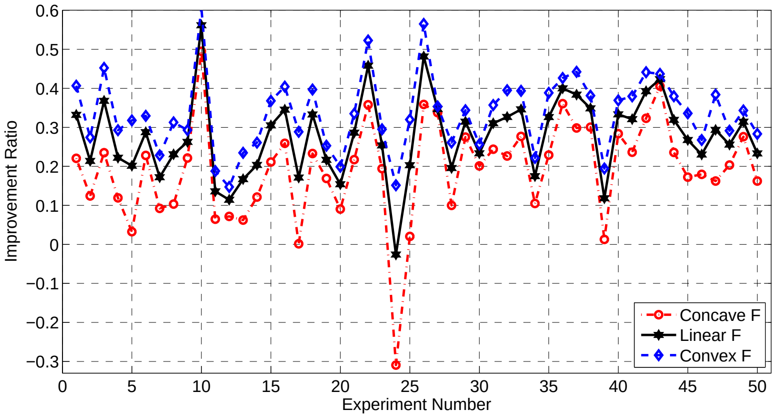

Figure III.23: Improvement Ratio Comparison for Concave, Linear, and Convex Penalty Function F.

Each experiment also runs over 50 different random network topologies for $2,000$ seconds.

Figure III.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 a convex function, local restoration (which tries to replace as few components/links as possible) incurs much less disruption penalty than global restoration due to the convex shape. Our MDSCR heuristic algorithm aggressively encourages local restoration and thus performs much better than the SPSCR algorithm. In the concave region, conversely, the benefits of local restoration are not significant, and the advantages of MDSCR are therefore less prominent.

CHAPTER IV

PROVIDING QOS SUPPORT FOR WIRELESS REMOTE HEALTHCARE SYSTEM

Introduction

The health care system has become a national challenge with a rapidly growing aging population and rising expenditure. According to the U.S. Census Bureau, the number of people over the age of 65 is expected to hit 70 million by 2030, having doubled since 2000. Health care expenditures in the United States are projected to rise to 15.9% of the GDP ($2.6 trillion) by 2010. This challenge calls for a major shift of health care from a traditional clinical setting to a patient/home-centered setting, which can reduce health care expenses through more efficient use of clinical resources, earlier detection of medical conditions and proactive management of wellness.

Recent advances in sensor technology and wireless networks have made it possible to deploy sensors on the human body and in the residential environment, allowing continuous remote health monitoring of human subjects. The application of wireless sensor networks to the medical environment, thus, provides a unique opportunity to shift health care outside a traditional clinical setting to a patient/home-centered setting. This will enable more efficient use of clinical resources and earlier detection of medical conditions, and as a result, significantly reduce healthcare expenses.

Towards this vision, wireless-sensor-based patient monitoring system becomes an active research area. Although a significant amount of efforts have been made, there remains a gap between the availability of the sensing technology and our ability to bring it into the practical use for home medical sensing. One of the critical issues is to assure the timely and robust delivery of the life-critical medical data in the resource-constrained wireless sensor networking environment. A remote healthcare system usually involves the usage of a variety of sensors. Different types of sensors (e.g., ECG (electrocardiogram) sensor, motion sensor) produce data with different characteristics which require different network assurance (e.g., delay, packet loss).

In this chapter, we investigate issue of providing Quality of Service (QoS) support for wireless remote healthcare systems. The proposed solution integrates three components: (1) QoS specification based on data-driven state machine, which is described using XML. The specification describes the QoS requirement of medical data under different patient conditions; (2) patient admission policy, which determines whether the sensor system has enough resource to support the required data delivery quality under all possible scenarios using Linear Programming (LP) techniques; and (3) differentiated scheduling and queue management, which enables data with higher quality requirement to enjoy a better treatment in the network. The proposed QoS support mechanism is implemented in *CareNet*, our two-tier wireless sensor system for remote healthcare. Extensive experiment results show that our system can provide low latency and low loss rate assurance to critical medical service traffic.

Related Work

Wireless-sensor-based patient monitoring system has become an active research area. To name a few, CodeBlue [24] has developed sensor devices that collect the heart rate, the oxygen saturation levels and ECG data. The data is transmitted to receiving devices such as laptops or PDAs, which can display the data in real time. Alarmnet [61] builds a wireless sensor network for smart health care, which integrates heterogeneous devices including wearable sensors and static environmental sensors. The Assisted Living Project [31] also integrates sensing, computing, wireless networking and middleware technologies to buld an assisted living environment for elderly people. MEDiSN [38] system consists of patient monitors, which collect the patients' vital signs and pass over the information through a wireless mesh infrastructure to the doctor station via a tree protocol. The work of [19] has developed a medical body networks using three types of lightweight and wearable sensors for data sensing, processing, and collecting respectively.

System Overview

We first introduce *CareNet*, our remote healthcare system. *CareNet* has three components: patient data collection at the *Home Healthcare Gateway*, data transmission from the *Home Healthcare Gateway* to the *Medical Record Database*, and data access at the *Accessing Client*, as shown in Figure IV.1.

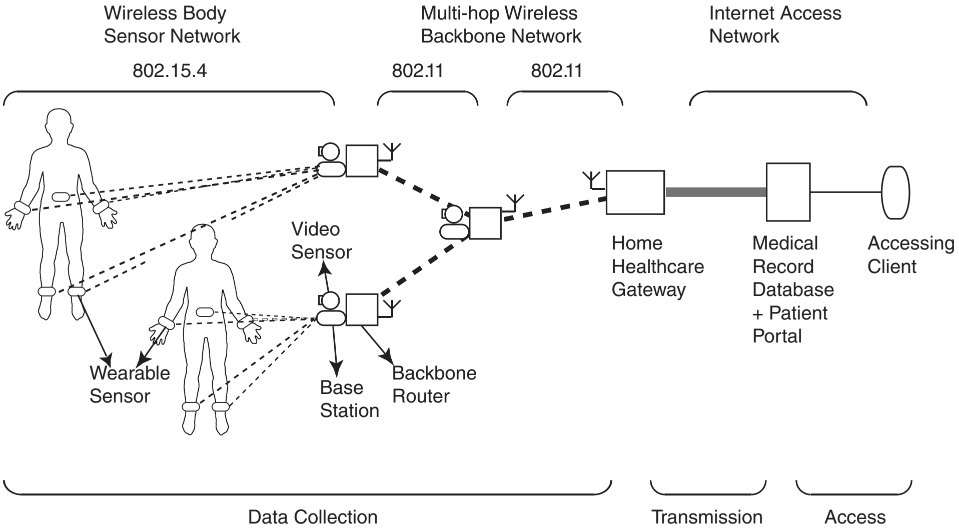

Figure IV.1: System Architecture.

As illustrated in Figure IV.1, a two-tier wireless network is used to provide data sensing and collection functions. At the lower tier, a body sensor network consisting of lightweight mobile *wearable sensors* provides data sensing and transmission functions. These sensors can communicate with the *base-station sensors* (which are attached to the backbone wireless network) directly using IEEE 802.15.4 wireless standard. Handoff service is provided for mobile wearable sensors when they roam between different base-stations. In particular, the base-stations can automatically receive the incoming data from the wearable sensors using synchronization messages. To ensure reliable packet delivery during the mobile sensor and base-station hand-offs, packets from the mobile sensors will be received by all base-stations within their transmission ranges. This means that the same data may be forwarded through more than one base-station (as well as backbone routers). To remove the duplicate data packets from the backbone network, each sensor data packet is marked with a timestamp in its packet header. Duplicate data packets that arrive late at the queue of a router will be dropped. The remaining duplicate and out-of-order packets will be dropped and sorted at the *Home Healthcare Gateway*.

At the upper tier of the network is a multi-hop IEEE 802.11-based wireless network, which provides a high-performance *backbone structure* for packet routing and forwarding. The backbone routers are connected to the base-station sensors that communicate with the mobile wearable sensors directly. Equipped with IEEE 802.11 wireless adaptors, the backbone routers communicate with each other and relay the sensing data to the *Home Healthcare Gateway*. Finally, the *Home Healthcare Gateway* serves as an interface between the patient's home and the caregiver's medical system, which processes all the sensing data and transmits them to the remote medical care system.

The backbone network of *CareNet* serves an important role of data delivery. Compared with sensor networks in which wireless communications are solely based on IEEE 802.15.4 standard, our backbone network design greatly improves the system reliability and scalability. The routing protocol among the backbone routers is implemented at the application level. This requires no modification of the operating system code, and thus makes it portable upon various operating systems. In particular, between each pair of neighboring backbone routers, a TCP tunneling is established to provide reliable single-hop wireless transmissions.

System Prototype

Our current *CareNet* system prototype provides physical activity monitoring and on-demand video monitoring services. The physical activity monitoring is able to provide continuous regular physical movement monitoring. Using a fall detection algorithm [18] that detects falls with the combination of speed and orientation changes, it is also able to provide emergent alarms when a fall is detected. The on-demand video monitoring can be used for movement data verification and analysis.

Hardware Devices

Table IV.1 summarizes all the hardware devices we use. We use Telos motes as the hardware devices of the body sensor network. For movement sensing and fall detection, these motes are equipped with accelerometers and gyroscopes. We use Stargate single board computers as the hardware devices of the wireless backbone network. The Stargate board is also connected with a web camera and serve as a video sensor.

Device	Description
Telos mote	MoteIV TMote Sky
Stargate Processor Board	SPB400CB Processor Board
Stargate Daughter Card	SDC400CA Daughter Card
WiFi Card	AmbiCom Wave2Net IEEE 802.11b Wireless CompactFlash Card
Webcam	Logitech QuickCam Pro 4000
Memory Stick	Kingston 512MB DataTraveler USB Flash Drive
Usb Hub	Belkin USB 2.0 Thumb Hub

Table IV.1: Hardware Devices.

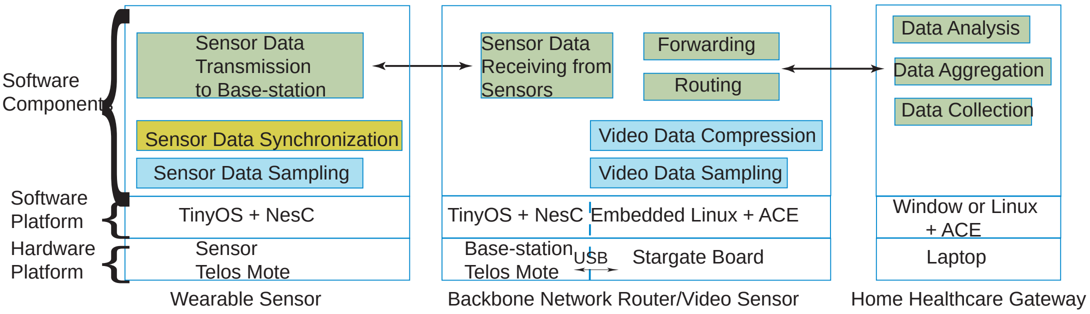

Figure IV.2: Software Architecture.

Software Design

CareNet is built upon a multi-layered software infrastructure based on the features and functions at each of the network tiers. The overall software architecture is shown in Figure IV.2.

We use the TinyOS [3] operating system and the NesC programming language to implement the movement data sensing at the wearable sensors and the data transmission between the mobile wearable sensors and the base-stations. We use Embedded Linux operating system and the ACE [1] programming environment to implement the network communication among the backbone network routers and between the backbone routers and the *Home Healthcare Gateway*. There are two major functions implemented at this layer: as backbone routing structure, software components are built to route and forward the video and sensor data to the *Home Healthcare Gateway*; as video sensors, video data sampling and compression functions are also implemented. We take advantage of ACE's strong communication and concurrency capabilities in our implementation. We use the Linux operating system and the ACE programming environment to develop the application software for the *Home Healthcare Gateway*.

QoS Support for Remote Healthcare System

In our remote healthcare system, different types of sensors (e.g., ECG sensor, motion sensor) produce data with different characteristics which require different network assurance (e.g., delay, packet loss).

To provide the QoS support for the sensor data delivery, our design integrates three components: (1) QoS specification, (2) QoS setup via patient admission control, and (3) run-time QoS support via differentiated scheduling and queue management.

QoS Specification Based on XML

The QoS requirements of the sensor data are determined by two factors: (1) sensor data type. For example, the physiological information (ECG, blood oxygen levels) may require a higher level of reliability than video sensor information; (2) Medical condition. For example, when a patient is having an unusual heart rate, the ECG data are highly emergent. These sensor data require higher reliability and lower delay.

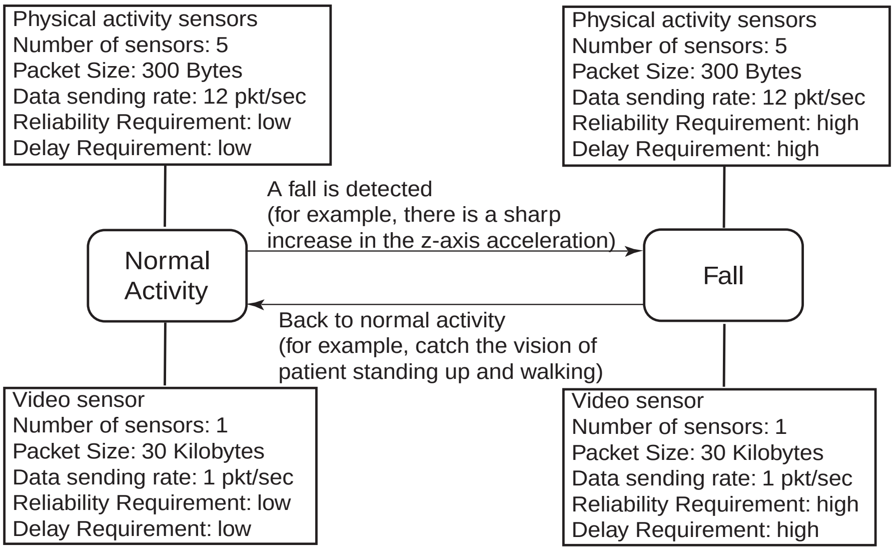
Physical activity sensors
Number of sensors: 5
Packet Size: 300 Bytes
Data sending rate: 12 pkt/sec
Reliability Requirement: low
Delay Requirement: low
Physical activity sensors
Number of sensors: 5
Packet Size: 300 Bytes
Data sending rate: 12 pkt/sec
Reliability Requirement: high
Delay Requirement: high
A fall is detected
(for example, there is a sharp
increase in the z-axis acceleration)
Normal Activity
Fall
Back to normal activity
(for example, catch the vision of
patient standing up and walking)
Video sensor
Number of sensors: 1
Packet Size: 30 Kilobytes
Data sending rate: 1 pkt/sec
Reliability Requirement: low
Delay Requirement: low
Video sensor
Number of sensors: 1
Packet Size: 30 Kilobytes
Data sending rate: 1 pkt/sec
Reliability Requirement: high
Delay Requirement: high

Figure IV.3: An State-machine with State-based QoS Requirements for Patient Physical Activity Monitoring.

We use a data-driven state-machine to specify the QoS requirement of sensor data under different patient conditions. Figure IV.3 shows an example of patient physical activity monitoring application. It has two states: *Normal Activity* and *Fall*. Transitions can occur from *Normal Activity* to *Fall* when

```
<state id="Normal_Activity">
  <sensor_type id="Physical_activity_sensor">
    <Number value=5>
    <Packet_Size value=300 Bytes/>
    <Sending_Rate value=12 unit="pkt/sec"/>
    <Reliability_Requirement value=low />
    <Delay_Requirement value=low />
  </sensor_type>
  <sensor_type id="Video_sensor">
    <Number value=1>
    <Packet_Size value=30 Kilobytes/>
    <Sending_Rate value=1 unit="pkt/sec"/>
    <Reliability_Requirement value=low />
    <Delay_Requirement value=low />
  </sensor_type>
  <transition condition="fall_detected">
    <target next="Fall"/>
  </transition>
</state>
```

```
<state id="Fall">
  <sensor_type id="Physical_activity_sensor">
    <Number value=5>
    <Packet_Size value=300 Bytes/>
    <Sending_Rate value=12 unit="pkt/sec"/>
    <Reliability_Requirement value=high />
    <Delay_Requirement value=high />
  </sensor_type>
  <sensor_type id="Video_sensor">
    <Number value=1>
    <Packet_Size value=30 Kilobytes/>
    <Sending_Rate value=1 unit="pkt/sec"/>
    <Reliability_Requirement value=high />
    <Delay_Requirement value=high />
  </sensor_type>
  <transition condition="back_to_normal">
    <target next="Normal_Activity"/>
  </transition>
</state>
```

Figure IV.4: XML Definition of the Example State-machine.

a fall is detected, and from *Fall* to *Normal Activity* when the patient is back to his/her normal activity. We use five physical activity sensors and one video sensor at each state. Since the *Fall* state is more emergent than the *Normal Activity* state, for both physical activity sensor data and video sensor data, the reliability the delay requirements of the former state is higher than those of the latter state. The exact values of QoS requirements of the sensors at different states as well as the transition conditions are specified by a particular patient and his/her family doctor.

We use the eXtensible Markup Language (XML) [4] to describe the state-machines. Figure IV.4 shows the XML definition of the example.

Patient Admission Policy

To ensure that the QoS requirements of all the sensor data transmissions are satisfied, we must first determine how many patients (i.e., number of sensors) can be supported by the current system resource and only admit the patients and their required sensors up to the system capacity. This is a non-trivial issue, as the sensors in use and their corresponding sending rates may be different for a patient under different medical conditions based on the QoS specification. In particular, we use d_p^s to denote the traffic demand (the total sending rate of all the sensors) of patient p at state $s^p \in S^p$, where S^p is the set of all the states of patient p. Our admission control policy should consider the worst-case resource utilization scenario under all possible states.

Specifically, a patient who wishes to use the current sensor network must first request an admission. This involves sending a connection request message to the *Home Healthcare Gateway* and informing the network about the types of the sensor data and their corresponding sending rates under regular and emergent states. All the previous request information of other accepted patients has been stored at the *Home Healthcare Gateway.* The *Home Healthcare Gateway* judges whether the sensor network has enough resource available to accept the connection, and then either accepts or rejects the connection request. If the network has enough resource, for each patient under a particular state, the actual traffic being routed should be no less than its traffic demand.

To determine whether the network has enough resource to support the required traffic demand, we can consider an alternative question – whether the traffic demand can be routed so that the possible throughput is larger than the demand. Formally, we define the minimum throughput *scaling* *factor* λ^s under a given state s, which is the minimum, over all the patients, of the actual traffic being routed divided by its traffic demand. Let us first consider the calculation of λ^s under a given state s. The maximum value of λ^s can be defined using the following routing optimization formulation, which is an integer linear programming (ILP) problem [6, 40].

$$\text{maximize} \quad \lambda^s \tag{IV.1}$$

subject to

$$\forall e \in E, \sum_{e' \in I(e)} \sum_{p \in P} x_p^s(e') \leq c \tag{IV.2}$$

$$\forall p \in P, \forall u \in V - \{s_p\},$$

$$\sum_{\substack{v \in V, \\ e=(v,u)}} x_p^s(e) - \sum_{\substack{v \in V \\ e=(u,v)}} x_p^s(e) = 0 \tag{IV.3}$$

$$\forall p \in P, \sum_{\substack{v \in V \\ e=(v,h)}} x_p^s(e) - \sum_{\substack{v \in V, \\ e=(h,v)}} x_p^s(e) = \lambda^s \cdot d_p^s \tag{IV.4}$$

$$\forall e \in E, \forall p \in P, x_p^s(e) = 0 \; or \; x_p^s(e) = \lambda^s \cdot d_p^s \tag{IV.5}$$

In this formulation, c denotes the channel capacity; $e \in E$ represents a wireless edge, $v \in V$ represents a backbone router; $I(e)$ denotes the set of edges which interfere with edge e; s_p is the backbone router that originally receives the sensor data from patient p; h denotes the *Home Healthcare Gateway*; $x_p^s(e)$ is the amount of patient p's traffic being routed on edge

e. Inequality (IV.2) is the wireless channel constraint. Equation (IV.3) and Equation (IV.4) are the flow conservation conditions. Equation (IV.5) means that a edge either does not transmit patient p's traffic, or transmits all of his/her traffic.

Now let's consider the value of λ under all possible states. In this case, we use the worst case network utilization to make an admission decision. Formally

$$\begin{aligned} &\lambda = \min_{s} \lambda^{s} \\ &\forall p_i \in P, \forall s_{p_i} \in S_{p_i}, s = \{s_{p1}, s_{p2}, ..., s_{p_i}, ...\} \end{aligned} \tag{IV.6}$$

λ reflects the minimum proportion of the traffic that can be routed for each patient over his/her traffic demand under all possible patient medical conditions. Therefore, as illustrated inEquation (IV.6), for any patient at any state, the value of the corresponding λ should be no less than 1 in order to admit the new patient.

Differentiated Scheduling and Queue Management

For the patients who are admitted into the system, their sensor data is associated with two types of priorities (services) – the reliability-based priority and the delay-based priority. Initial packet priority setting is straightforward: for two packets, if packet 1 requires a lower delay than packet 2, then its delay priority is higher than packet 2. The reliability-based priority is similarly set. The priorities of a packet will be used in scheduling and queue management at the backbone routers.

The behavior of an individual router is based on the priorities of the packets it receives. We use one normal packet queue at each router to store the packets received, and use two virtual priority queues, one for reliability priority and one for delay priority, to store the location and priority values of each received packet.

The priorities of the packets that stored in the packet queue are dynamically adjusted: the reliability priority decreases with time and the delay priority increases with time. This is to ensure that the oldest packet has more opportunity to be scheduled, however, it is also prone to be dropped if it has not be scheduled.

Our scheduling and queue management policies are based on the comparison of the priorities of the packets, which need to achieve the goal that the highest delay priority packet has the least delay and the highest reliability priority packet has the least likelihood of being dropped due to a queue reaching its maximum capacity. These two policies are described as follows.

• *Differentiated Scheduling.* When the network adaptor is ready to schedule a packet, the packet with the highest delay priority is scheduled and transmitted to the next hop.

• *Queue Management.* When the queue is full at the backbone router, the packet with the lowest reliability priority will be dropped.

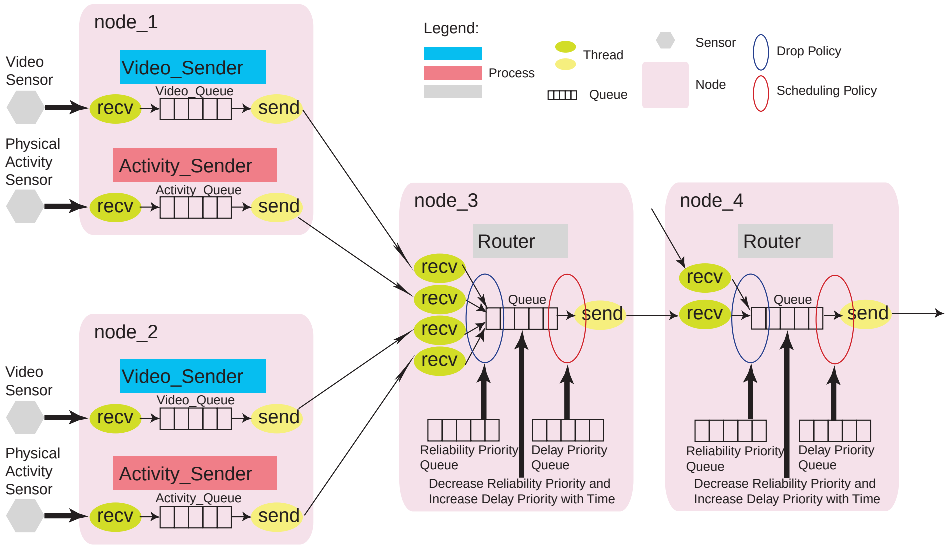

Figure IV.5: Backbone Router Using Priority Queue and A Single Sending Thread.

A backbone router implements the following two functionalities at the application level: (1) Routing and forwarding functionality, which means select the next hop router and forward sensor data to it. This is implemented based on the optimization-based routing protocol; and (2) Queuing and scheduling functionality, the design details of which are shown in Figure IV.5.

The backbone router is running a "Router" process. We use several receiving threads, each of which is used by a predecessor router, and a single sending thread, which is used by a successor router; these threads are spawned by the "Router" process at the backbone router. The number of the receiving threads is the number of TCP connections requested by other backbone routers it accepted. There is a packet queue shared by all of these threads. This queue uses the reliability priority queue and the delay priority queue to determine which packet to be dropped and which packet to be scheduled.

If a backbone router receives the sensor data directly from a patient, it also performs video and/or physical activity data sender functionality. As a video data sender, a backbone router is running a "Video_Sender" process, which spawns two threads, one for receiving video packet from the video sensor, the other for sending video packet to the next-hop backbone router. There is a video packet queue shared by these two threads. As a physical activity data sender, a backbone router performs the similar functionality.

System Prototype Experiment and QoS Evaluation

This section describes our system prototype experiment and evaluates the performance and QoS support capability of our remote healthcare system.

System Prototype Experiment

Figure IV.6 shows the screen shot for an experiment using our system prototype. In the experiment, five physical activity sensor motes are mounted on a patient, two on the wrists, two on the ankles, and one on the waist. Each sensor mote is capable of recording accelerations in three dimensions (x, y, and z axis) as well as rotations in two dimensions (x and y axis). We received

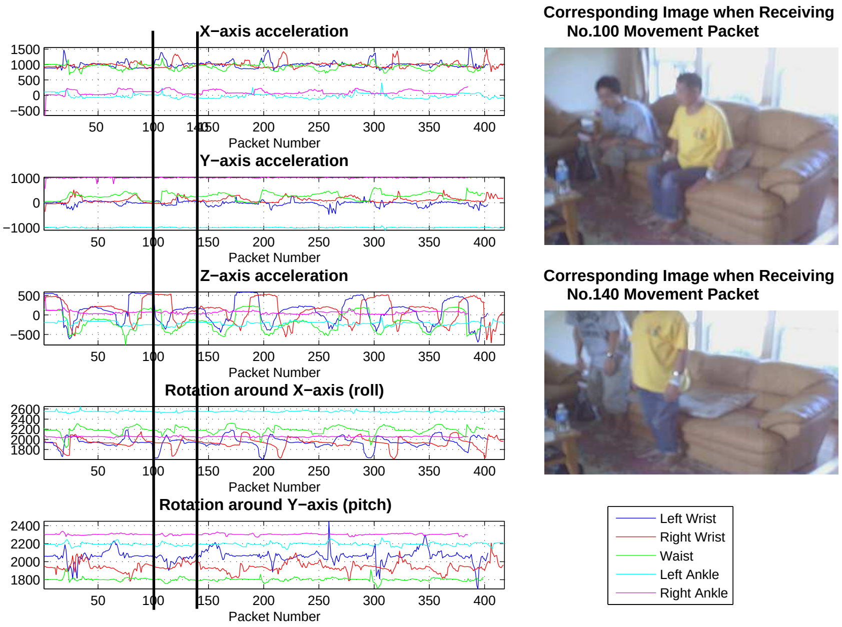

Figure IV.6: Sit-to-stand and stand-to-sit experiment.

more than 400 movement packets from each of the five sensors, and for each packet we recorded the X-, Y-, Z-axis accelerations and the rotations around X-, Y-axis. We also show the corresponding video images when receiving movement packets 100 and 140. Detailed description of our software design and experiment results can be found in our earlier work [33].

System Performance and QoS Experiment

Using the priority setting algorithm and the queuing and scheduling methods, and the design and implementation of the backbone routers, we conduct a set of system performance and QoS experiments with and without QoS

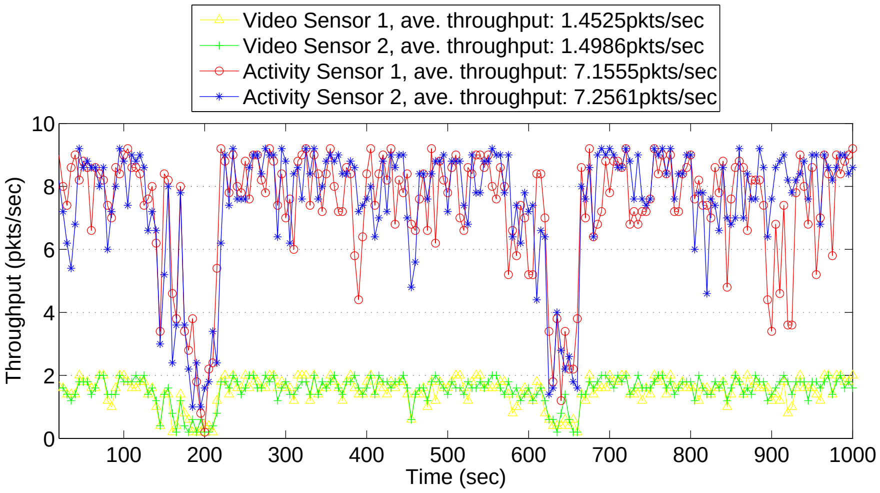

Figure IV.7: Throughput (scenario 1).

support. We compare the throughput, delay, and drop rate of the sensor data packets from two patients; patient 1 is under the *Normal Activity* state and patient 2 is under the *Fall* state. Each patient is equipped with five physical activity sensors and is monitored by one video sensor.

In this experiment, we suppose that for both the physical activity sensors and video sensors, the reliability requirement is 0.65 and delay requirement is 90 *milliseconds* at the *Normal Activity* state; and 0.85 and 45 *milliseconds* at the *Fall* state.

Without QoS Support (Scenario 1)

For the first set of experiments, we do not provide the QoS support. All of these sensors are with the same reliability-based priority and delay-based

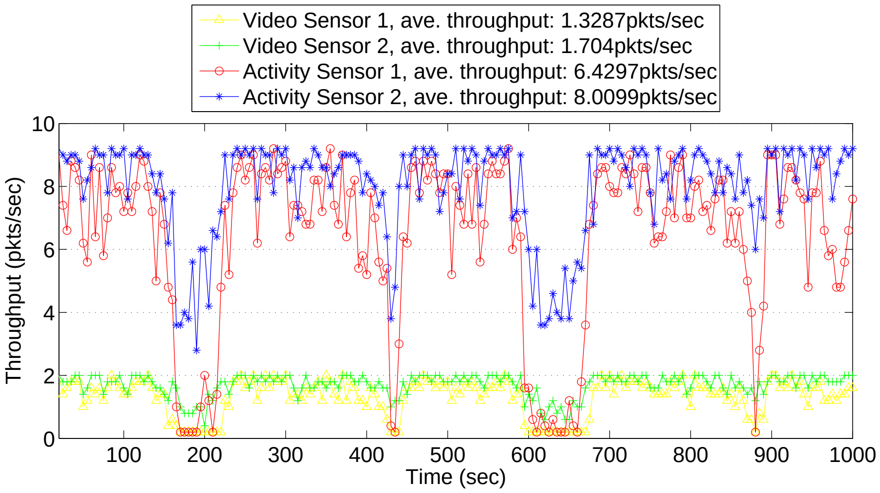

Figure IV.8: Throughput (scenario 2).

priority. The results of the throughput, drop rate, and delay are shown in Figure IV.7, Figure IV.9, and Figure IV.11.

Table IV compares the reliability and delay of these sensors under the same priority settings; they satisfy the QoS requirements defined before under the *Normal Activity* state; however, they do not satisfy the QoS requirements under the *Fall* state.

With QoS Support (Scenario 2)

For the second set of experiments, since patient 2 is at the *Fall* state, we increase the reliability and delay priorities of video sensor 2 and physical activity sensor set 2. The results of the throughput, drop rate, delay and are shown in Figure IV.8, Figure IV.10, and Figure IV.12. From these figures,

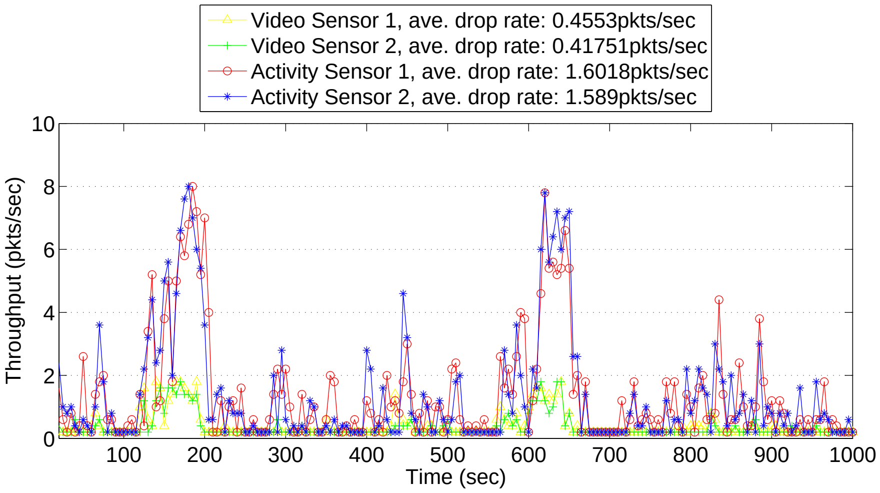

Figure IV.9: Drop Rate (scenario 1).

	Video Sensor 1	Physical Activity Sensor Set 1	Video Sensor 2	Physical Activity Sensor Set 2
Reliability	0.7613	0.8171	0.7821	0.8203
Delay (msec)	58.34	61.59	57.24	61.06

Table IV.2: Reliability and Delay with the same Priority Settings.

we can see that the sensors that have higher reliability and delay priorities achieve higher throughput, lower delay, and lower drop rate.

Table IV compares the reliability and delay of these sensors under different priority settings; they satisfy the QoS requirements defined before under both the *Normal Activity* and *Fall* states.

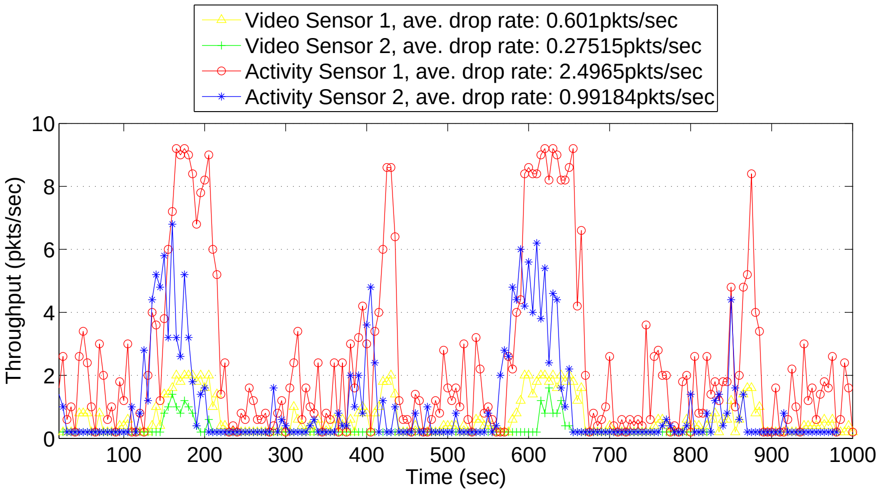

Figure IV.10: Drop Rate (scenario 2).

	Video Sensor 1	Physical Activity Sensor Set 1	Video Sensor 2	Physical Activity Sensor Set 2
Reliability	0.6886	0.7203	0.8610	0.8898
Delay (msec)	89.07	70.55	37.66	42.75

Table IV.3: Reliability and Delay of with different Priority Settings.

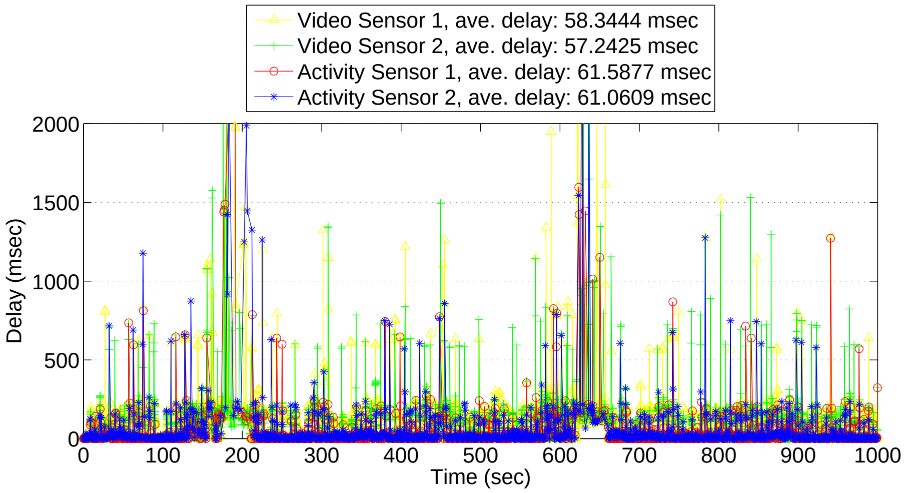

Figure IV.11: Delay (scenario 1).

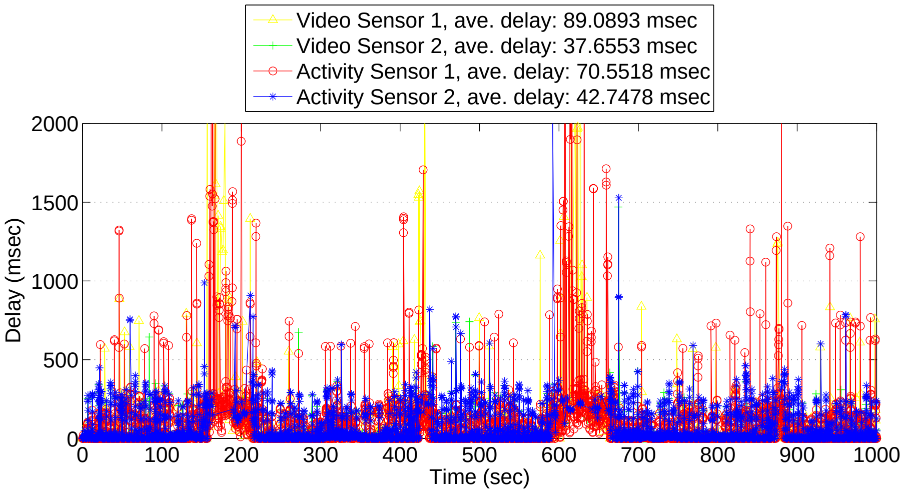

Figure IV.12: Delay (scenario 2).

CHAPTER V

CONCLUDING REMARKS

Wireless networks are more prone to failures than their wireline counterparts. The unique characteristics of wireless networks introduce fundamental challenges to the design of restoration related wireless networks that can satisfy the performance and reliability requirements. Network and service restoration schemes posed for wireline networks (such as the Internet) are poorly suited for highly dynamic and unstable wireless networks. This dissertation investigates the network and service restoration as well as QoS support issues in the design of reliable multi-hop wireless networks. Specifically, as shown in Table V.1, it investigates the following three problems for wireless stationary and mobile networks, respectively.

Network Restoration for Multi-hop Wireless Stationary Networks	Linear programming approach is applied to minimize the performance degradation caused by node failures and jamming attacks
Service Restoration for Multi-hop Wireless Mobile Networks	Dynamic programming approach is applied to minimize the service disruption caused by node mobility
QoS Failure Prevention for Wireless Remote Healthcare System	XML-based service description, patient admission policy, differentiated scheduling and queue management is applied to provide QoS support

Table V.1: Research Contributions.

It first investigates the network restoration problem in multi-hop wireless stationary networks under node failures and jamming attacks. The proposed defense strategy dynamically adjusts the channel assignment and traffic routes to bypass the failed nodes and the jamming area. The goal is to minimize the performance degradation caused by node failures and jamming attacks. To achieve this goal, an optimization-based approach is applied, which formulates network restoration strategies as linear programming problems and gives an upper bound on the achievable network throughput. After we solve the LP problems, we have a set of flows assigned to edges that have been assigned to different channels. And based on the LP solutions, we provide a greedy scheduling algorithm using dynamic channel assignment, which schedules both the network traffic and the jamming traffic. We further provide a greedy static edge channel assignment algorithm, where a channel is assigned to an edge at the beginning and will remain fixed over all time slots. In particular, we consider two strategies, namely *global restoration* and *local restoration*, which can support a range of tradeoffs between the restoration latency and network throughput after restoration. To quantitatively evaluate the impact of network failures during and after restoration, we define two performance degradation indices, *transient disruption index (TDI)* and *throughput degradation index (THI).* Network performance of these optimal network restoration strategies is evaluated via comprehensive simulation study under different failure scenarios.

It second investigates the service composition and restoration problem in multi-hop wireless mobile networks under frequent mobility-caused wireless link failures. To address this issue, we propose a service composition and restoration framework designed to achieve minimum service disruptions. 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 restoration (MDSCR) as a dynamic programming problem and analyze the properties of its optimal solution for wireless mobile networks with known mobility plan. Based on the derived analytical insights, we present our MDSCR heuristic algorithm for wireless mobile 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 conduct comprehensive simulation study to compare and analyze the results of throughput and disruption index of our MDSCR algorithm with the traditional methods (*e.g.*, the shortest path routing and service composition) under the impacts of service path length, service link length requirement, traffic type, service component redundancy, system dynamics, and disruption penalty function.

It also investigates the QoS support scheme for health monitoring services that integrates XML-based service description, patient admission policy, differentiated scheduling and queue management. The proposed solution is implemented in *CareNet*, our two-tier wireless sensor system for remote healthcare. Extensive experimental results show that our system can provide low latency and low loss rate assurance to critical medical service traffic.

BIBLIOGRAPHY

[1] *http://www.cs.wustl.edu/ schmidt/ACE.html.*

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

[3] *http://www.tinyos.net/tinyos-2.x/doc/.*

[4] *http://www.w3schools.com/xml/.*

[5] The lp_solve Mixed Integer Linear Programming (MILP) solver. http://lpsolve.sourceforge.net/5.5/.

[6] M. Alicherry, R. Bhatia, and L. Li. Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks. In *Proc. of the 11th Annual International Conference on Mobile Computing and Networking*, 2005.

[7] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar. Providing Quality of Service Over a Shared Wireless Link. *IEEE Communications Magazine*, 39(2), 2001.

[8] D. Applegate, L. Breslau, and E. Cohen. Coping with Network Failures: Routing Strategies for Optimal Demand Oblivious Restoration. *ACM SIGMETRICS Performance Evaluation Review*, 32(1), 2004.

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

[10] B. Awerbuch, A. Richa, and C. Scheideler. A Jamming-Resistant MAC Protocol for Single-Hop Wireless Networks. In *Proc. of Principles of Distributed Computing*, 2008.

[11] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP Performance over Wireless Networld. In *Proc. of International Conference on Mobile Computing and Networking*, 1995.

[12] 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*, 80(7), 2007.

[13] A. Banerjea. Fault Recovery for Guaranteed Performance Communications Connections. *IEEE/ACM Transactions on Networking*, 7(5), 1999.

[14] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa. On the Performance of IEEE 802.11 under Jamming. In *Proc. of IEEE INFOCOM*, 2008.

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

[16] J. Broth, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc NeWork Routing Protocols. In *Proc. of International Conference on Mobile Computing and Networking*, 1998.

[17] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. Service Composition for Mobile Environments. *Journal of Mobile Networks and Applications*, 10(4), 2005.

[18] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. Wearable Sensors for Reliable Fall Detection. In *Proc. of IEEE Engineering in Medicine and Biology Society*, 2005.

[19] F. Dabiri, H. Noshadi, and H. Hagopian. Lightweight Medical BodyNets. In *Proc. of BodyNets*, 2007.

[20] Liang Dai, Yuan Xue, Bin Chang, and Yi Cui. Throughput Optimization Routing under Uncertain Demand for Wireless Mesh Networks. In *Proc. of IEEE MASS*, 2007.

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

[22] C. Dovrolis and P. Ramanathan. Resource Aggregation for Fault Tolerance in Integrated Services Networks. *ACM SIGCOMM Computer Communication Review*, 28(2), 1998.

[23] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi. An Empirically Based Path Loss Model for Wireless Channels in Suburban Environments. *IEEE Journal on Selected Areas in Communications*, 17(7), 1999.

[24] T. Gao, T. Massey, L. Selavo, M. Welsh, and M. Sarrafzadeh. Participatory User Centered Design Techniques for a Large Scale Ad-Hoc Health Information System. In *Proc. of HealthNet*, 2007.

[25] Y. Gao, D.-M. Chiu, and J. C. S. Lui.

[26] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[27] M. Grunewald, T. Lukovszki, C. Schindelhauer, and K. Volbert. Distributed Maintenance of Resource Efficient Wireless Network Topologies. *Lecture Notes in Computer Science*, 2400, 2002.

[28] X. Gu and K. Nahrstedt. A Scalable QoS-Aware Service Aggregation Model for Peer-to-Peer Computing Grids. In *Proc. of IEEE HPDC*, 2002.

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

[30] P. Gupta, , and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.

[31] J. Hou, Q. Wang, L. Ball, S. Birge, M. Caccamo, C. Cheah, E. Gilbert, C. Gunter, E. Gunter, C. Lee, K. Karahalios, M. Nam, N. Nitya, C. Rohit, L. Sha, W. Shin, Y. Yu, and Z. Zeng. PAS: A Wireless-Enabled, Sensor-Integrated Personal Assistance System for Independent and Assisted Living. In *Proc. of Joint Workshop on HCMDSS and MDPnP Interoperability*, 2007.

[32] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of Interference on Multi-hop Wireless Network Performance. In *Proc. of International Conference on Mobile Computing and Networking*, 2003.

[33] S. Jiang, Y. Cao, S. Iyengar, P. Kuryloski, R. Jafari, Y. Xue, R. Bajcsy, and S. Wicker. CareNet: An Integrated Wireless Sensor Networking Environment for Remote Healthcare. In *Proc. of 3rd International Conference on Body Area Networks*, 2008.

[34] S. Jiang, D. He, and J. Rao. A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks. In *Proc. of IEEE Infocom*, 2001.

[35] 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.

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

[37] Shrikrishna Karandikar, Shivkumar Kalyanaraman, Prasad Bagal, and Bob Packer. TCP rate control. *Journal of ACM SIGCOMM Computer Communication Review*, 30(1), 2000.

[38] J. Ko, R. Musaloiu-E., J. H. Lim, Y. Chen, A. Terzis, T. Gao, W. Destler, and L. Selavo. Demo Abstract: MEDISN: Medical Emergency Detection in Sensor Networks. In *Proc. of ACM Conference on Embedded Networked Sensor Systems*, 2008.

[39] M. Kodialam and T. V. Lakshman. Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration. In *Proc. of IEEE INFOCOM*, 2000.

[40] Murali Kodialam and Thyaga Nandagopal. Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proc. of ACM Mobicom*, 2005.

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

[42] 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*, 2(1), 2004.

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

[44] V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic Aspects of Capacity in Wireless Networks. In *Proc. of ACM SIGMETRICS*, pages 133–144, 2005.

[45] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne. Multiple Routing Configurations for Fast IP Network Recovery. *IEEE/ACM Transactions on Networking*, 2008.

[46] B. Li and K.H. Wang. NonStop: Continuous Multimedia Streaming in Wireless Ad Hoc Networks with Node Mobility. *IEEE Journal on Selected Areas in Communications*, 21(10), 2003.

[47] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris. Capacity of Ad Hoc Wireless Networks. In *Proc. of International Conference on Mobile Computing and Networking*, 2001.

[48] X. Lin and S. Rasool. A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks. In *Proc. of IEEE Infocom*, 2007.

[49] 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.

[50] 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.

[51] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic Bandwidth Management and Adaptive Applications for a Variable Bandwidth Wireless Environment. *IEEE Journal on Selected Areas in Communications*, 19(10), 2001.

[52] K. Murakami and H. S. Kim. Optimal Capacity and Flow Assignment for Self-healing ATM Networks Based on Line and End-to-end Restoration. *IEEE/ACM Transactions on Networking*, 6(2), 1998.

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

[54] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A General Model of Wireless Interference. In *Proc. of International Conference on Mobile Computing and Networking*, 2007.

[55] B. Raman and R. H. Katz. An Architecture for Highly Available Wide-Area Service Composition. *Computer Communications Journal*, 26(15), 2003.

[56] R. Ramanathan and J. Redi. A brief overview of ad hoc networks: challenges and directions. *IEEE Communications Magazine*, 40(5), 2002.

[57] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. PATHS: Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols. In *Proc. of ACM MobiHoc*, 2003.

[58] B.H. Sathyaraj and R.C. Doss. Route Maintenance Using Mobility Prediction for Mobile Ad Hoc Networks. In *Proc. of IEEE Mobile Ad-hoc and Sensor Systems*, 2005.

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

[60] A. P. Snow, U. Varshney, and A. D. Malloy. Reliability and Survivability of Wireless and Mobile Networks. *IEEE Computer Magazine*, 33(7), 2000.

[61] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood. Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges. In *Proc. of HCMDSS Workshop*, 2005.

[62] 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*, 80(5), 2007.

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

[64] J. Tang, G. Xue, and W. Zhang. Reliable Ad Hoc Routing Based on Mobility Prediction. *Journal of Combinatorial Optimization*, 11(1), 2006.

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

[66] X. Wang and H. V. Poor. Adaptive Joint Multiuser Detection and Channel Estimation in Multipath Fading CDMA Channels. *Wireless Networks*, 4(6), 1998.

[67] Z. Wang and J. Crowcroft. Analysis of Shortest-path Routing Algorithms in a Dynamic Network Environment. *Journal of ACM SIGCOMM Computer Communication Review*, 22(2), 1992.

[68] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang. Distributed Channel Assignment and Routing in Multiradio Multi-channel Multihop Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 24(11), 2006.

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

[70] Y. Xiong and L. G. Mason. Restoration Strategies and Spare Capacity Tequirements in Self-healing ATM Networks. *IEEE/ACM Transactions on Networking*, 7(1), 1999.

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

[72] W. Xu, W. Trappe, and Y. Zhang. Channel Surfing: Defending Wireless Sensor Networks from Interference. In *Proc. of Information Processing in Sensor Networks*, 2007.

[73] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proc. of ACM Mobihoc*, 2005.

[74] 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.

[75] 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.

[76] T. Yu, Y. Zhang, and K.J. Lin. Efficient Algorithms for Web Services Selection with End-to-end QoS Constraints. *Journal of ACM Transactions on the Web (TWEB)*, 1(1), 2007.

[77] 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.

[78] H. Zhai, X. Chen, and Y. Fang. How Well Can the IEEE 802.11 Wireless LAN Support Quality of Service. *IEEE Transactions on Wireless Communications*, 4(6), 2005.

[79] J. Zhao and R. Govindan. Understanding Packet Delivery Performance In Dense Wireless Sensor Networks. In *Proc. of Embedded Networked Sensor Systems*, 2003.