ABSTRACT

Title of dissertation: TOPOLOGY CONTROL ALGORITHMS FOR RULE-BASED ROUTING

Kiran Kumar Somasundaram,
Doctor of Philosophy, 2010

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

In this dissertation, we introduce a new topology control problem for rule-based link-state routing in autonomous networks. In this context, topology control is a mechanism to reduce the broadcast storm problem associated with link-state broadcasts. We focus on a class of topology control mechanisms called local-pruning mechanisms. Topology control by local pruning is an interesting multi-agent graph optimization problem, where every agent/router/station has access to only its local neighborhood information. Every agent selects a subset of its incident link-state information for broadcast. This constitutes the pruned link-state information (pruned graph) for routing. The objective for every agent is to select a minimal subset of the local link-state information while guaranteeing that the pruned graph preserves desired paths for routing.

In topology control for rule-based link-state routing, the pruned link-state information must preserve desired paths that satisfy the rules of routing. The non-triviality in these problems arises from the fact that the pruning agents have access

to only their local link-state information. Consequently, rules of routing must have some property, which allows specifying the global properties of the routes from the local properties of the graph. In this dissertation, we illustrate that rules described as algebraic path problem in idempotent semirings have these necessary properties.

The primary contribution of this dissertation is identifying a policy for pruning, which depends only on the local neighborhood, but guarantees that required global routing paths are preserved in the pruned graph. We show that for this local policy to ensure loop-free pruning, it is sufficient to have what is called an inflatory arc composition property. To prove the sufficiency, we prove a version of Bellman's optimality principle that extends to path-sets and minimal elements of partially ordered sets.

As a motivating example, we present a stable path topology control mechanism, which ensures that the stable paths for routing are preserved after pruning. We show, using other examples, that the generic pruning works for many other rules of routing that are suitably described using idempotent semirings.

TOPOLOGY CONTROL ALGORITHMS FOR RULE-BASED ROUTING

by

Kiran Kumar Somasundaram

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Armand Makowski
Professor Richard J. La
Professor Steve Marcus
Professor S. Raghavan

Acknowledgments

It is my pleasure to record my gratitude towards my advisor Prof. John S. Baras for his support and encouragement during my graduate studies at the University of Maryland. His diverse interests and expertise gave me the freedom to work on problems of my interest. I thank him for urging me to discover underlying principles that connect disparate areas of research, which I believe has bought clarity to my thinking and research.

I am also grateful to my dissertation committee members – Prof. A. Makowski, Prof. R. La, Prof. S. Marcus and Prof. S. Raghavan – for agreeing to serve in my committee. I must also thank Prof. S. Marcus for referring to related work on algebraic dynamic programming, which helped improve the dissertation manuscript.

I want to thank the administrative staff of the ISR and the ECE department, and in particular, Kim Edwards for helping me in several official and administrative matters. I am grateful to all my colleagues and friends who have offered help in various ways. I like to specially thank, my good friend, Kaustubh Jain for the several constructive conversations that we had concerning this dissertation topic. I also like to thank Ranjit Kumaresan for introducing me to shortest path problems, which is pivotal to this dissertation.

My profound sense of gratitude is to my family: my mother, my father and my brother. Their constant support and confidence in me made graduate studies really enjoyable. I am greatly indebted to my beloved Gowri, for without her love, support and encouragement, this dissertation might not have been possible.

My research work was supported by the Army Research Office award number W911NF-08-1-0238. This support is acknowledged with gratitude.

Table of Contents

List of Figures

List of Abbreviations

Sets

$\mathbb{R}$	Set of reals
$\mathbb{Z}$	Set of integers
$\mathbb{R}_+$	Set of non-negative reals
$\mathbb{Z}_+$	Set of non-negative integers
$\hat{\mathbb{R}}_+$	The set $\mathbb{R}_+ \cup \{\infty\}$
$\hat{\mathbb{Z}}_+$	The set $\mathbb{Z}_+ \cup \{\infty\}$

Graphs

$G(V, E)$	Undirected graph with vertex set V and edge set E
$G(V, A)$	Directed graph with vertex set V and arc set A
$G' \subseteq G$	G' is a subgraph of G
$p \in G'$	p is a path in graph G'
$P_{ij}^{G'}$	Set of paths between vertices $i, j \in V'$
$2^{P_{ij}^{G'}}$	Set of all sub-sets of $P_{ij}^{G'}$
$p_{(k)}$	$k \geq 1$, k^{th} vertex in the sequence of vertices constituting path p
	$k \leq 1$, k^{th} vertex in the sequence of vertices of path p reversed
$\Omega_i^{G'}$	In an undirected graph G', the set of incident edges to $i \in V'$; in a directed graph G', the set of incident in-arc for $i \in V'$
$d_{hc}(i, j)$	Minimal hop-count distance between $i, j \in V$
N_h^k	k-hop neighborhood of $h \in V$
∂N_h^k	Boundary of N_h^k
G_h^{local}	Local view of $h \in V$
γ_p^h	Gateway for a path p in G_h^{local}
Ω_h^{pruned}	Pruned incident edge set of h
$G^{broadcast}$	Broadcast view
G^{global}	Global view, which is $G_h^{local} \cup G^{broadcast}$
w_p	Scalar weight of the path p for scalar arc weights
	Vector weight of the path p for vector arc weights
w_P	For $P \subset P_{ij}^{G'}$, $w_P = \{w_p,\ p \in P\}$
$x_{ij}^{G'}$	Optimal metric from $i \in V$ to $j \in V$ in G'
	For vector weight, non-dominated solution
$P_{ij}^{G'*}$	Optimal path-set from $i \in V$ to $j \in V$ in G'
$\mathcal{P}_{ij}^{G'*}$	Efficient path-set from $i \in V$ to $j \in V$ in G'
$x_{ij}^{h-local}$	Optimal metric from $i \in V$ to $j \in V$ in G_h^{local}
	For vector weight, non-dominated solution
$P_{ij}^{h-local*}$	Optimal path-set from $i \in V$ to $j \in V$ in G_h^{local}
$\mathcal{P}_{ij}^{h-local*}$	Efficient path-set from $i \in V$ to $j \in V$ in G_h^{local}

Rules

π_h^R	A global property at $h \in V$ for a rule R
Π_h^R	Subgraphs that have the property π_h^R
$\pi_h^{\Omega-R}$	A local property at $h \in V$ for a rule R
$\Pi_h^{\Omega-R}$	Subsets of arcs that have the property $\pi_h^{\Omega-R}$
$\otimes_l$	Componentwise arc composition operator
$\otimes$	Vector arc composition operator
$\oplus$	Path selection operator
Min	Non-dominance function

Others

TC	Topology Control
SPTC	Stable Path Topology Control
NDC	Neighbor Discovery Component
STIDC	Selector of Topology Information to Disseminate Component
TDC	Topology Dissemination Component

Chapter 1

Introduction

1.1 Motivation for Topology Control in Link State Routing

Wireless multi-hop networks, such as Wireless Sensor Networks (WSNs) and Mobile Ad Hoc Networks (MANETs) [45] are, typically, constructed on the fly, with no pre-existing infrastructure. For these ad hoc networks, mobility of the nodes and their ad-hoc association makes the routing control mechanism critical to the functioning of the network [45]. A successful routing protocol for these networks is one that provides a mechanism to deliver data packets to any destination of the network even under dynamic topologies [60].

Routing protocols in mobile multi-hop networks are broadly classified as reactive, proactive and hybrid [45]. In reactive protocols, a source requests for a route to a particular destination if it has data to send. On the other hand, in proactive routing protocols, every source maintains at-least one route to every destination of interest, by periodic updates. Hybrid protocols adopt mechanisms from both reactive and proactive protocols. Many routing protocols, proposed for MANETs, use network broadcasting as means to establish routes [30, 6, 12, 24]. Network broadcasting, referred to as broadcasting in this dissertation, is a process by which a packet sent from one station reaches all the stations in the network.

In this dissertation, we study the broadcasting associated with proactive link-state protocols. Link state protocols, such as OSPF [39] and OLSR [11], broadcast link-state information necessary for routing: every station broadcasts its local link-state, and consequently, every station is aware of the global link-state of the entire network. Subsequently, unicast traffic is routed by hop-by-hop routing to any destination station [39, 11]. In this dissertation, we investigate the former aspect of link-state routing protocols: broadcasting of link-state information. The motivation to study this arises from a problem associated with link-state broadcast in mobile networks: in mobile multi-hop networks, link-states are very dynamic, and consequently, a large number of routing control packets, corresponding to every link-state change, is broadcast in the network. This control overhead throttles the limited resources of a multi-hop wireless network. This problem is referred to as the *broadcast storm problem* [60].

A class of algorithms called *Local Pruning* or *Neighbor Knowledge* Topology Control (TC) algorithms [60, 61] reduces the broadcast storm by controlled flooding. Controlled flooding, as opposed to naive flooding, constructs an overlay-flooding network to broadcast a reduced topology information. In these algorithms, controlled flooding becomes feasible due to some local neighborhood information that the stations learn in the neighbor discovery process: in the neighbor discovery phase for many link-state routing mechanisms, the stations discover only their immediate neighboring stations; however, in local pruning algorithms, the stations discover not only their neighbors, but also the neighbors of their neighbors and other associated neighborhood metrics. This additional information is then used to reduce the broadcast storm by two means:

1. Topology Compression - Reducing the link-state information that is broadcast by selecting only a subset of the topology information that is “sufficient” for routing.

2. Efficient Broadcast - Reducing redundant broadcasts by selecting a subset of stations (flooding overlay network) that ensures network-wide broadcast.

In this dissertation, our focus is on the former: we develop algorithms for topology compression, i.e., pruning to preserve a sufficient topology for routing. In particular, we develop algorithms that preserve paths that satisfy Quality of Service (QoS) requirements.

1.2 Survey of Topology Control Algorithms

In [31], we introduced a component-based architecture for link-state routing protocols. Fig. 1.1 illustrates the primary components that are necessary for topology broadcast. Link-state routing protocols have a neighbor discovery component that enables each station to discover its local neighborhood information. This local neighborhood information is fed into two components, Selector of Topology Information to Disseminate Component (STIDC) and Topology Dissemination Component (TDC). The functionality of the STIDC is to perform topology compression (Section 1.1) and that of the TDC is to perform efficient broadcast (Section 1.1). The functionality of the three components are explained in greater detail in Chapters 2 and 4.

In this dissertation, since we investigate only the topology compression aspect of link-state broadcasting, we will assume that there exists a TDC that is capable of disseminating the compressed topology. We separate the concerns between the STIDC and the TDC by making the functionalities of the two components orthogonal/independent. In other words, the algorithms that we develop for topology compression will be independent of the underlying broadcast mechanism.

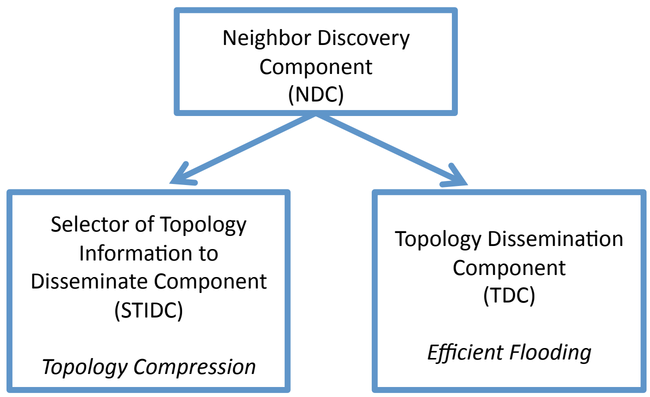


Figure 1.1: Components of Link State Broadcast Mechanism

Before we present a survey of different TC algorithms, we introduce the topology selection and the broadcast mechanism of OLSR [12, 11], in detail, because it forms the foundation for the local pruning TC algorithms that we introduce in this dissertation.

1.2.1 Topology Control Mechanism of OLSR

In Optimized Link State Routing (OLSR) protocol [12, 11], every host station, in the network, discovers its local neighborhood by periodic HELLO messages [10]. Since every host locally broadcasts to its neighbors the set of neighbors that it can hear, every host discovers its one-hop neighbors and all the neighbors for each of these one-hop neighbors (called two-hop neighbors). Fig. 1.2 shows the neighborhood that the host h discovers from the HELLO messages. The neighbor discovery protocol [10] is designed to discover only symmetric neighbors (that can hear each other), and consequently, all the links discovered are undirected. In the example of Fig. 1.2, $\{i_1, i_2, \cdots, i_5\}$ are the one-hop neighbors, and $\{j_1, j_2, \cdots, j_7\}$ are the two-hop neighbors of h. Note that the neighbor discovery protocol at h does not expose the links between two-hop neighbors, i.e., h cannot discover the links between j_m and j_n, $1 \leq m,\ n \leq 7$. This local graph discovered at the host h is called the *local view* of h. The notion of a local view is defined more formally in Chapters 2 and 4.

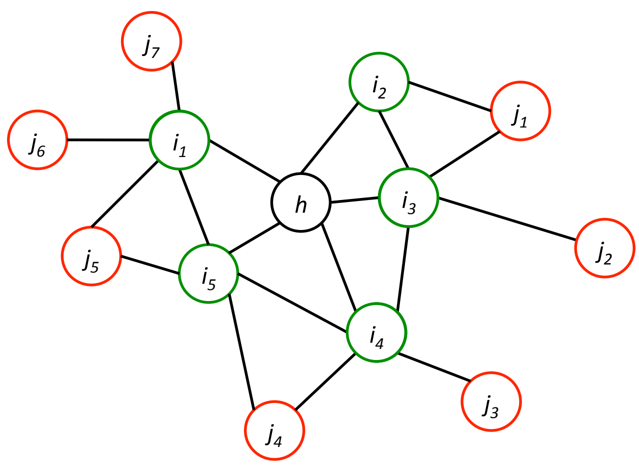


Figure 1.2: Local view of the host h

The original version of OLSR [12] treats the topology-pruning problem as Connected Dominating Set (CDS) construction given only the local view [61]. The graph-theoretic concept of CDS is explained in Appendix A.3. Every host solves a local set-cover problem in the local view to find the minimum set of one-hop neighbors that cover all two-hop neighbors. For instance, in the example local view shown in Fig. 1.2, the host h selects from the set of one-hop neighbors, $\{i_1, i_2, \ldots, i_5\}$, a minimal subset that cover all two-hop neighbors, $\{j_1, j_2, \ldots, j_7\}$: all the two-hop neighbors must be reachable from this selected subset of one-hop neighbors in the local view. This distinguished subset, of one-hop neighbors, is called Multi-Point Relay (MPR) set in OLSR. The pruning problem to compute the MPR set is shown to be NP-hard and a greedy heuristic is used to obtain the MPR set [46]. For this example of Fig. 1.2, the MPR set is $\{i_1, i_3, i_4\}$. Then the host broadcasts links $\{(h, i_1), (h, i_3), (h, i_4)\}$ across the entire network. Similarly, every station in the network selects and broadcasts a subset of its incident links. The selection of a subset of the incident links corresponds to the topology compression functionality of the STIDC. It is shown in [29] that this reduced topology preserves the shortest hop-count paths for routing.

The MPR set serves another functionality: OLSR uses the multipoint relay stations to form an overlay-flooding network to achieve network-wide broadcast of this compressed topology. Since the MPRs form a CDS [61], flooding via the MPRs guarantees network-wide broadcast [29]. This reduced flooding corresponds to the efficient broadcasting functionality of the TDC. Interestingly, in OLSR both STIDC and TDC functionalities are served by the MPR sets! However, in general, the two functionalities could be achieved by different mechanisms.

1.2.2 Other Topology Control Algorithms

There are several local pruning TC algorithms proposed in the literature. We present a brief summary of these algorithms.

Scalable Broadcast Algorithm (SBA) proposed in [43], uses local neighborhood information, in particular, the connectivity of its neighbors to schedule local re-broadcasts. The dominant pruning algorithm proposed in [1] identifies a dominant set of broadcast relays using a greedy set-cover algorithm. The multi-point relaying algorithm (explained in Subsection 1.2.1) proposed in [46] is a more optimized version of dominant pruning. The ad hoc broadcast protocol proposed in [44] is a more efficient broadcast mechanism that uses the MPR construction. For a summary of the different local pruning algorithms proposed for MANET routing, see [60]. In [61], the authors show most of the pruning algorithms construct a CDS using a local dominance/priority function. This work encompasses a large class pruning algorithms that includes [43, 1, 46, 44, 28, 20, 8, 54, 27, 59].

Appendix A.3 illustrates why a CDS is used for constructing an overlay-flooding network in multi-hop wireless networks. A CDS is a covering property of the vertices and does not translate trivially to a property of paths. Consequently, the CDS constructions in literature do not offer any guarantees on the QoS of the path preserved (those preserved subsequent to pruning).

1.3 Degrees of Local Information

In pure link-state mechanisms, such as OSPF [39], the link-state broadcast makes visible the entire weighted communication graph. This raw information creates a complete *global view* of the network that is "more than" sufficient for routing. In this dissertation, as introduced in Section 1.2, we investigate protocols that have local neighborhood information (made visible by the NDC). This additional information provides an opportunity to reduce the amount of link-state broadcast. The recipe in these local pruning algorithms is illustrated in Fig. 1.3. Note that the gains in topology compression are essentially due to the local link-state discovery using the NDC.

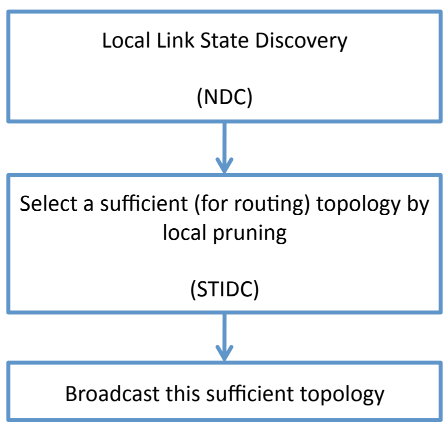


Figure 1.3: Recipe for Local Pruning Algorithms

At a superficial level, the above recipe suggests that these optimized link-state routing protocols break-down the link-state discovery mechanism into two stages: local discovery by NDC and global discovery by TDC. When viewed at this superficial level, it might appear that there is no reduction in the overhead because the cost of performing neighbor discovery might balance or even outweigh the savings from reduced link-state broadcast. However, in several protocols such as OLSR [11], the neighbor discovery mechanism [10] offers the local link-state information for free. Any modification to these protocols to discover more local link-state information will always incur an additional cost and must be considered in designing the protocol.

There is also another advantage in discovering the local link-state. In mobile multi-hop wireless networks, the local link-state is likely to be very dynamic, e.g., the link stability metrics can change drastically in a fading wireless channel. Usually such highly dynamic links are not preferred for routing and consequently, they can be discarded from broadcast. Consequently, the link-state information broadcast rate is reduced, i.e., observing the local link-state enables to create a more stable routing graph. This observation is used in the Stable Path TC algorithm described in Chapter 2. However, there is always a trade-off between the cost in learning the local link-state information and savings in the link-state broadcast. This is will characterized formally in Chapter 2.

1.4 Contributions of the Dissertation

In this dissertation, we formulate the TC problem as a multi-agent graph pruning problem. The agents are the stations/routers of the network, and these agents have access to only their local neighborhood information. The agents, using this constrained view of the network, select a minimal local topology to construct a pruned graph for routing. However, there is a constraint on the pruned graph that it must preserve the globally desirable paths for routing described by a rule for routing.

We identify a class of rules that can be described as an algebraic path problem in idempotent semirings – allows specifying global properties of paths from local properties of the graph. We show that this class of rules satisfy, what we call, the "distribution of order" property that enables local pruning; in essence, this distribution of order property is Bellman's optimality principle extended to path-sets and minimal elements of partially ordered sets. This becomes necessary when you have rules defined over multiple metrics – vector metrics.

The primary contribution of this dissertation is identifying a policy for pruning, which depends only on the local neighborhood, but guarantees that the pruned graph preserves desirable paths described by generic rules of routing. We show that for this local policy to ensure loop-free pruning, it is sufficient to have what is called an inflatory arc composition property.

Thus, the two primary contributions of this dissertation are the following:

1. Extension of algebraic routing to vector metrics, where notions of optimum and optimality are replaced by that of non-dominance and efficiency, respectively. We present a version of Bellman's optimality principle extended to multiple vectors, minimal elements and efficient path-sets.

2. Identification of sufficient local pruning conditions for rule described using idempotent semirings – strict inflatory arc composition – that guarantee loop-free pruning.

1.5 Organization of the Disseration

This dissertation is organized into three primary parts (three chapters):

1. We introduce the stable path topology control problem. This serves as a motivating example for the generic topology control algorithms that will follow in the subsequent chapters. In this part, we develop the machinery that is necessary to formally specify the topology control problem. The results in this part make extensive use of Bellman's optimality principle: Bellman's optimality principle yields a tool by the global QoS of a path can be described by the QoS of a local path (visible in the local neighborhood). In this part, we develop sufficient conditions of local pruning that guarantee to preserve global QoS stable paths for link-state routing. We also formally characterize the benefit and cost in having additional local neighborhood information. Finally, using simulations, we present quantitative comparisons between our topology control algorithm and those in prior art.

2. We introduce rule-based routing. We illustrate that rules of compositions that yields path weights and rules of path selections can be effectively described using idempotent semiring algebras. This part is intended to serve as a tutorial to algebraic routing. The essence of this part is to introduce the notion of "distribution of order" and to formally define it for path-sets.

3. We develop the generic topology control (link-state selection) pruning conditions for rule-based routing. We extend the pruning conditions of the stable

path topology control problem, of the first part, to a generic rule-based routing framework. The local policies that we develop illustrate the need for the distribution of order property, which is a tool to translate local properties to global properties of directed labeled graphs. In doing so, we show a manifestation of the Bellman's optimality principle that extends to path-sets and minimal elements of partially ordered sets. We show a sufficient condition, called strict inflatory arc composition, which guarantees loop-free pruning.

Chapter 2

Stable Path Topology Control

2.1 Overview

In this chapter, we introduce the stable path topology control problem. This serves as a motivating example for the generic topology control algorithms that will follow in the subsequent chapters.

In this chapter, we develop the machinery that is necessary to formulate the topology control problem. Although, the notions are tailored for the stable path topology control, they are in a form from which they can be easily extended to the generic rule-based routing. The results in this chapter make extensive use of Bellman's optimality principle: Bellman's optimality principle yields a tool by the global QoS of a path can be described by the QoS of a local path (visible in the local neighborhood). The main results in this chapter are in developing sufficient conditions of local pruning that guarantee to preserve global stable paths for link-state routing.

We also formally characterize the benefit and cost in having additional local neighborhood information. Finally, using simulations, we present quantitative comparisons between our topology control algorithm and those in prior art.

This chapter is organized as follows. We first introduce the notion of stable path routing in Section 2.2. Next, in Section 2.3, we introduce the mathematical notations, definitions and terminology necessary to describe the topology control problem. In Section 2.4, we introduce the notion of local pruning and policies for local pruning; we establish a sufficient condition for loop-free local pruning. Finally, in Section 2.5, we present an approximation algorithm to solve the stable path topology control problem and substantiate its performance with simulation results.

2.2 Stable Path Topology Control

One important metric for routing in MANETs is *path longevity* or *path stability* [9]. In this dissertation, we refer to it as path stability. Although path stability has been studied for many reactive distance vector schemes [9, 49], there is little work that addresses topology control for stable paths in link state routing.

In this chapter, we introduce a new topology control algorithm: *Stable Path Topology Control* (SPTC), is a mechanism to prune the initial topology (to reduce the broadcast storm) while guaranteeing that the stable paths for (unicast-)routing from every host to any target station are preserved in the pruned topology. Topology control for stable paths has a two-fold advantage:

1. These long-lived paths are *cheaper to maintain* because they are less likely to change.

2. It offers the higher layer traffic long-lived paths and consequently yields *improved traffic carrying performance.*

Note that our goal is not to engineer new metrics, but to develop TC algorithms that can use the metrics that have been proposed in literature. We will, next, introduce some popular stability metrics.

2.2.1 Stability Metrics

Majority of routing protocols proposed for wireless multi-hop networks, both reactive and proactive, are mechanisms that use *hop-count* as the metric for routing [45]: traffic is routed along the minimum hop-count path from source to destination. However, wireless links in a multi-hop network are vulnerable to frequent breakage due to mobility and channel erasures [55, 21, 3]. Hence, schemes based merely on hop-count, which are inherently insensitive to the dynamic stability of the paths, have shown poor performance [15]. This limitation has inspired a number of protocols that use *link stability* as a metric for routing.

Perhaps, the earliest MANET protocol to use link stability metric for routing is the Associativity Based Routing (ABR) scheme [58], which uses an *associativity* threshold used to predict the stability of a neighboring station. It assumes neighbors that remain associated beyond this threshold are less likely to move away and hence form stable links. Signal Stability based Adaptive routing (SSA) [18] is another link stability based routing protocol that uses signal strength and location information from neighboring stations to estimate the stability of the links. Route-lifetime Assessment Based Routing (RABR) [2] is an extension to SSA that uses thresholding of link ages to choose routes. Mobility prediction was suggested in [55] to improve unicast and multicast routing protocols for MANETs. This scheme uses

GPS location information to estimate the residual lifetime for links.

In [21], the authors present a simulation study of the empirical distribution of link lifetimes for various mobility models [7]. From these empirical distributions, they also derive a method to compute the residual lifetime distribution for these different mobility models. The study reveals that there are strict thresholds beyond which the residual lifetimes exhibit a positive correlation with the link age. Another statistical characterization of link lifetimes is presented in [9]. These simulation results show that longer lifetime paths tend to have longer length (in hop-count), suggesting that there is a tradeoff between path stability and path length (delay). The Stability and Hop-count based Algorithm for Route Computing (SHARC) [53] identifies this tradeoff and combines a link stability metric and the hop count metric to find short paths (in terms of hop-count) that also have good stability (lexicographic ordering).

Another simulation study, presented in [48], shows that path life is inversely related to the maximum velocity and the hop-count, and is directly related to the transmission range. The authors observe that under high mobility patterns, the path durations can be approximated using exponential distributions. In [25], *Han* et. al. use Palm calculus to show that under certain conditions, the path durations converge to exponential distributions as the number of hop-count increases.

The wireless mesh networking community has also been actively developing several stability metrics for routing. Since the backbone routers of a wireless mesh network are stationary, routing using link stability metrics, rather than mere hop-count, is more feasible compared with the MANET case, where the network topology is more dynamic [33]. To the best of our knowledge, the first metric proposed for wireless mesh networks is the Expected Transmission Count (ETX) metric in [14, 13]. The ETX metric for a link is the expected number of attempts for a packet to successfully reach the other end of the link, in an Automatic Repeat reQuest (ARQ) scheme. The authors of [14] design the ETX metric for 802.11 MAC with ARQ. Thus, the ETX metric accounts for link stability both in the forward and the reverse direction of the link.

The ETX metric of a link is calculated using the forward and reverse delivery ratios of the link. The forward delivery ratio, df, is the measured probability that a data packet successfully arrives at the recipient, and the reverse delivery ratio, dr, is the probability that the ACK packet is successfully received. The probability that a transmission is successfully received and acknowledged is $df \times dr$. A sender will retransmit a packet that is not successfully acknowledged. Since each attempt to transmit a packet can be considered to be Bernoulli trial, the expected number of transmissions is given by

$$ETX = \frac{1}{df \times dr}$$

The ETX metric of a path is the sum of the link ETX metrics along the path; the sum gives the expected number of transmissions to deliver a packet successfully from the source vertex to the terminal vertex of the path. Although, it is not included in the RFC [12], the ETX metric has been incorporated in popular OLSR implementations [62, 47].

In [42], the authors argue that the ETX metric, being additive, suffers from route oscillations. Instead, they propose, a multiplicative metric, Minimum Loss (ML) metric that computes the loss probability of a path. Another problem with the ETX metric computation is that the data and control packets are typically larger than the probe packets used to compute the metric, and consequently, the computed loss probabilities need not be equal to the loss probabilities of the data and control packets. This problem is identified and a new metric called Expected Transmission Time (ETT), which computes the expected transmission time instead of the count, is proposed in [17]. ETT adapts the ETX for different PHY transmission rates and packet sizes. They also propose the Weighted Cumulative ETT (WCETT) metric that modifies ETT to also consider intra-flow interference. This metric is composed of both end-to-end delay and channel diversity; a tunable parameter is used to combine both components.

In wireless networks, the link stability is usually highly dynamic, and consequently, several of the metrics proposed, if used crudely, can cause significant control overhead or route oscillations [33]. In [33], the authors propose two metrics: modified ETX (mETX) and Effective Number of Transmissions (ENT) that consider the variance of the link-stability while computing the metrics. Another metric that considers link-quality variation is iAWARE [56]. This metric uses the signal to noise ratio and signal to interference and noise ratio to continuously reproduce neighboring interference variations onto routing metrics. A number of other link stability metrics have also been proposed for MANETs; they include [58, 49, 38, 9, 63, 55, 53, 3, 18, 2].

2.2.2 Limitations of Existing Topology Control Mechanisms for Topology Compression

Algorithms that make use of link stability metrics, in most cases, are modifications to reactive distance vector protocols such as Dynamic Source Routing (DSR) [30] and Ad Hoc On-demand Distance Vector (AODV) [6]; these include Link Quality Source Routing (LQSR) [17], Multi-Radio LQSR (ML-LQSR) [17], SrcRR [13] and others. There are few proactive link-state routing protocols that incorporate these link stability metrics for topology control for controlled flooding. Most of these are variants of OLSR's [12] pruning methods (Subsection 1.2.1). In [4, 40, 36], the authors propose modifications to OLSR's MPR selection algorithm to incorporate link stability metrics: the MPR selection is posed as a weighted set-cover algorithm [32]. The implementations of the OLSR protocols in [62, 47] provide provisions to compute the ETX link metric. They also provide options to use the ETX metric in calculating the MPR set (Subsection 1.2.1). Let $ETX(u, v)$ be the symmetric ETX metric for the link (u, v). In the topology control algorithm implemented in [62], the host h computes the ETX metric of the best two-hop path to reach a two-hop neighbor j by

$$\min_{l} ETX(h, i_l) + ETX(i_l, j), \tag{2.1}$$

where i_l's are the one-hop neighbors of h. The host then selects a minimal set of its one-hop neighbors – MPR set – such that all the two-hop neighbors are reachable from h via the best two-hop path, using these selected MPRs. In essence, this is another set-cover problem where all the two-hop neighbors are covered by a subset of one-hop neighbors, using the computed ETX weights (Equation (2.1)).

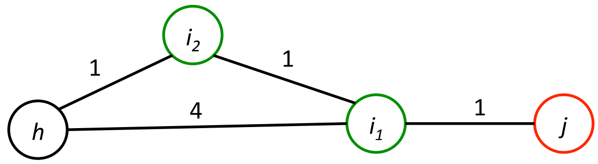


Figure 2.1: Example Local View with ETX metric for each link indicated

However, these set-cover methods offer no proof guarantees for the stability of the pruned paths, i.e., the stable paths for routing need not be preserved by these pruning methods. To illustrate this, consider a weighted graph shown in Figure 2.1. The symmetric ETX metrics for the links are indicated in the figure. The host h has two one-hop neighbors i_1 and i_2 and one two-hop neighbor j. In this example, the link (h, i_1) is unstable ($ETX = 4$), while all other links are stable. For this topology, the set-cover method of the implementations in [47, 62] has only one feasible (two-hop) path (h, i_1, j) to reach j, which has an ETX cost 5. Note that in the implementation, all feasible paths from h, the host, to reach j, the two-hop neighbor, is of the form (h, i, j), where i is an one-hop neighbor (Equation (2.1)). However, if we relax the artificial constraint that two-hop neighbors need to be reached from the host using *strictly* one one-hop neighbor, there exists an alternative better path (h, i_2, i_1, j) of ETX cost 3. Clearly, the set-cover pruning methods (of OLSR and its variants) will not preserve this stable path.

Note that this example is not a mere pedagogical example. In wireless radio networks, any one-hop neighbor that covers several two-hop neighbors is, typically, far from the host. Consequently, the link between the host and this one-hop neighbor is unstable. However, the set-cover formulation of [47, 62] will always choose this unstable neighbor as its MPR.

In the forthcoming sections, we will formulate and solve a distributed pruning (topology control) problem that can provably preserve all the stable paths to every destination station in the pruned topology. Our TC algorithm is not specific to the ETX metric, which has been discussed in this subsection, but can be applied to all the stability metrics discussed in Subsection 2.2.1.

2.3 Notations and Definitions

In this section, we will introduce some terminology and protocols that are relevant for the TC problem. See Appendix A.1 for definitions from elementary graph theory. In this chapter, we will deal with only edge-labeled undirected graphs.

2.3.1 Graphs and Neighborhoods

Let $G(V, E)$ denote the communication graph, where V is the vertex set of stations and E is the undirected edge set (communication adjacency between the vertices). For $(u, v) \in E$, there is an associated symmetric link stability metric $a_{uv} = a_{vu} \geq 0$. Thus, G is an undirected edge-weighted graph.

A subgraph of G, denoted by $G' \subseteq G$, is a graph $G'(V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$ (restricted to $V' \times V'$). The set of paths in any subgraph G' from $i \in V'$ to $j \in V'$ is denoted by $P_{ij}^{G'}$. For any path $p = (v_1, v_2, \cdots, v_n)$, $p_{(k)} = v_k$, for $1 \leq k \leq n$, i.e., $p_{(k)}$ denotes the k^{th} vertex of the sequence of vertices in p. For

$-n \leq k \leq -1$, $p_{(k)}$ denotes the k^{th} vertex of the reversed path, i.e., $p_{(k)} = v_{n-k+1}$, $-n \leq k \leq -1$. For any vertex $i \in V'$, the set of edges incident to i in G' is denoted by $\Omega_i^{G'}$.

We introduce the notion of hop-based neighborhoods, which is defined for the graph G. The hop-count $hc(p)$ of a path p is the number of edges in p. The minimal hop-count distance between a pair of vertices (i, j) in G is $d_{hc}(i, j) = \min_{p \in P_{ij}^G} hc(p)$. If j is not reachable from i in G, i.e., $P_{ij}^G = \emptyset$, then $d_{hc}(i, j) = \infty$. We define the k-hop neighborhood for a host/vertex $h \in V$ by

$$N_h^k = \{j \in V : d_{hc}(h, j) \leq k\}.$$

Here, k is called the *size of the neighborhood*. The boundary set for the neighborhood N_h^k is given by

$$\partial N_h^k = N_h^k \backslash N_h^{k-1},$$

where $N_h^0 = \{h\}$, and $N_h^k = \emptyset, k < 0$. Let $N_h^{k^-} = N_h^k \backslash \{h\}$ denote the *exclusive* neighborhood, which is the neighborhood excluding the host h.

2.3.2 Local and Global Views

In [52], we extended the notion of local and global views introduced in [61] to encompass edge-weighted graphs. We summarize these extensions in the subsections.

Link state routing protocols for MANETs, such as [12, 11, 47], have a local neighbor discovery phase/protocol [10], where every host/node discovers its local neighborhood, using periodic HELLO messages. This mechanism is called heartbeat neighbor discovery.

In OLSR [12], the HELLO message of a host consists of all its symmetric neighbors: symmetric neighbors are those that can hear each other. Apart from this adjacency information, the HELLO message also contains link quality information to each of these symmetric neighbors [11]. For instance, in the implementation of [47], the link quality information is the ETX link metric (Subsection 2.2.1) from the host to its symmetric neighbor. The HELLO message from every host is broadcast locally to all its neighbors. Subsequently, every host not only discovers its communication adjacencies and link qualities to all its neighbors, but also the communication adjacencies and link qualities of the neighbor to its neighbors. Formally, every host $h \in V$, discovers (h, i) and a_{hi}, $i \in \partial N_h^1$, and also (i, j) and a_{ij}, $i \in \partial N_h^1$, $j \in \partial N_i^1$.

We generalize this functionality of the [10] protocol, to formalize the neighbor discovery phase. We assume that the HELLO message from each host contains both the communication adjacency and the link quality information for all its $(k-1)$-hop neighbors ($k \geq 2$). Consequently, every host $h \in V$ discovers the communication adjacency and the link quality information in its k-hop neighborhood: a special labeled subgraph G_h^{local}, which is a subgraph of G, that contains only the vertices in N_h^k and all the edges between them, except those between any two vertices of the boundary set, i.e., the vertex set is N_h^k, the edge set is $\{(u, v) \in E : \ u, v \in N_h^k \text{ and } \{u, v\} \not\subseteq \partial N_h^k\}$, and the labels a_{uv} for every edge in this edge set. We call this labeled subgraph, G_h^{local}, the *local view* for the host h. Fig. 2.2 illustrates the notion of local view using an example.

We next introduce the notion of gateways for paths restricted to a local view.

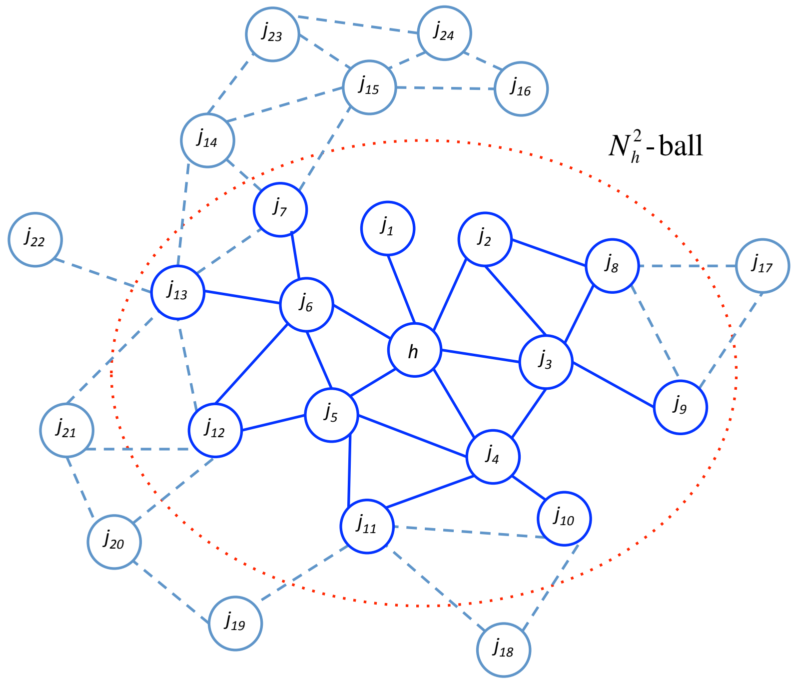


Figure 2.2: Local View G_h^{local}: Illustrates the local view of h for a neighborhood size $k = 2$. The N_h^2 neighborhood ball is shown by a dotted ellipse. Host h discovers all the nodes inside this ball. It also discovers the communication adjacencies and the link quality metrics for all the edges indicated by solid lines; it correspond to all the edges contained in the N_h^2 neighborhood ball, other than the edges between the boundary nodes, i.e., the edges between the nodes in ∂N_h^2. The edges that h cannot discover in the neighbor discovery phase is indicated by dashed lines. The boundary set of G_h^{local} is $\partial N_h^k = \{j_7, j_8, j_9, j_{10}, j_{11}, j_{12}, j_{13}\}$.

For any path $p = (h = u_1, u_2, \ldots, u_n = j) \in P^G_{hj}$, the *gateway* of p in G^{local}_h, denoted by γ^h_p, is the first vertex of p that is in the boundary set ∂N^k_h, i.e., $\gamma^h_p = u_t$ if and only if $u_t \in \partial N^k_h$ and $u_s \notin \partial N^k_h$, $1 \leq s < t$. If the path p never intersects ∂N^k_h, i.e., $u_s \notin \partial N^k_h$, $1 \leq s \leq n$, then γ^h_p is not defined. Fig. 2.3 illustrates the notion of a gateway.

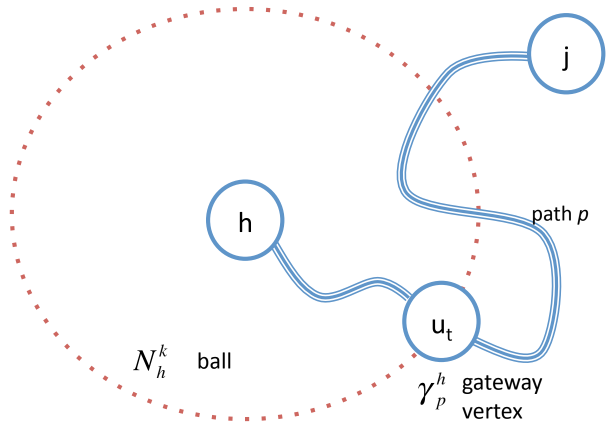


Figure 2.3: Gateway for the path p in the local view G^{local}_h. The N^k_h ball is indicated as a dotted ellipse. The gateway node γ^h_p is the node u_t.

In local pruning algorithms, the host $h \in V$, which has discovered its local view G^{local}_h, chooses a subset of its incident edges, which we call the *pruned incident edge set* of h. The set of edges incident to h in G^{local}_h is denoted by $\Omega^{G^{local}_h}_h$. However, since all the edges from the host to its one-hop neighbors are contained in G^{local}_h, $\Omega^{G^{local}_h}_h = \Omega^G_h$. The pruned incident edge set is denoted by $\Omega^{pruned}_h \in 2^{\Omega^G_h}$, where $2^{\Omega^G_h}$ is the power set of Ω^G_h, which is the set of all subsets of Ω^G_h. The local pruning for compressed link-state selection, essentially, boils to computing this Ω^{pruned}_h, given

G_h^{local}. In naive link-state flooding, the Ω_h^G is broadcast network-wide. However, in compressed link-state the pruned link-state $\Omega_h^{pruned} \subseteq \Omega_h^G$ is flooded, and this reduces the topology broadcast storm; the objective is to minimize $|\Omega_h^{pruned}|$ while guaranteeing that the pruned graph satisfies some global properties for routing.

Every host h broadcasts, using the Topology Dissemination Component (TDC), its pruned incident edge set, Ω_h^{pruned}, and the link quality metrics for these edges. The corresponding broadcast edge set is given by $E^{broadcast} = \cup_{h \in V} \Omega_h^{pruned}$, and this induces a labeled subgraph $G^{broadcast}$, which we call the *broadcast view*. The broadcast view is common to all the hosts $h \in V$. Fig. 2.4 illustrates the notion of a broadcast view.

Once the broadcast view is available at the host $h \in V$, it has the final link-state information to compute the paths for routing. This link-state information is a combination of the local view and the broadcast view, and we call this, the *global view*: At every host station $h \in V$, the *global view* G_h^{global} is the graph union $G_h^{local} \cup G^{broadcast}$ along with the corresponding edge weights, where G_h^{global} and $G^{broadcast}$ are exposed by the NDC and TDC respectively. Note that the global view, like the local view, is specific to a host h; every host h sees a different global view. Fig. 2.5 shows the global view corresponding the local and broadcast views of Figure 2.2 and 2.4 respectively.

Note that the edge-labeled graph G_h^{global} is the only information that h has to compute optimal routes to different destinations, which subsequently yields the best next-hop to forward the packets to any destination node. We will define the notion of optimality for stable routing in the following subsection.

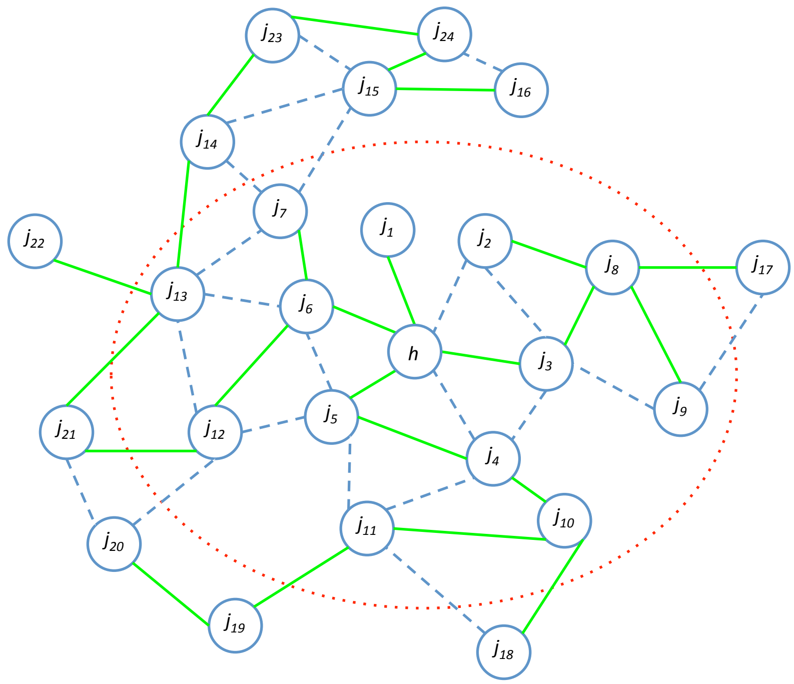


Figure 2.4: Broadcast View $G^{broadcast}$. The edges indicated by solid lines correspond to $E^{broadcast}$, while the edges indicated by dashed lines are those that are pruned during the topology compression.

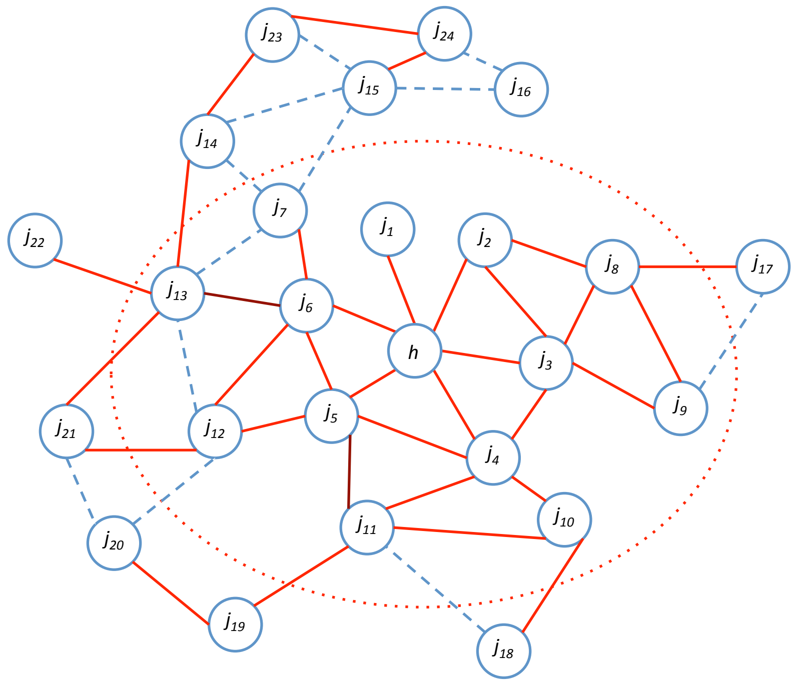


Figure 2.5: Global view of h corresponding to the local view of Fig. 2.2 and the broadcast view of Fig. 2.4. The edges indicated by the solid lines are visible in the global view, while those indicated by dashed lines are neither broadcast nor visible in the local view. The global view is the graph union of the local and broadcast views; it also includes the weights/labels for the edges in the global view, which are the link quality metrics.

2.3.3 Path Stability

For the stability metrics discussed in Subsection 2.2.1, the stability of path p, denoted by w_p (weight of the path), is computed by composing the link stability metrics $a_{uv},\ (u,v) \in p$. Most of the metrics from Subsection 2.2.1 follow either additive or multiplicative compositions. Since a multiplicative composition can be transformed to an additive composition, i.e., using logarithms, we only consider additive compositions for path stability:

$$w_p = \sum_{(u,v)\in p} a_{uv}.$$

The optimal value of path stability between a vertex pair (i,j) in G' is

$$\begin{aligned} x_{ij}^{G'} &= \min_{p \in P_{ij}^{G'}} w_p \\ &= \min_{p \in P_{ij}^{G'}} \sum_{(u,v)\in p} a_{uv}, \end{aligned} \tag{2.2}$$

and the corresponding optimal path set, which is the set of paths that achieve the optimal value of the path stability from i to j in the subgraph G', is

$$P_{ij}^{G'*} = \{p \in P_{ij}^{G'} :\ w_p = x_{ij}^{G'}\}.$$

In essence, computing the optimally stable paths corresponds to computing the shortest paths in G'. For an introduction to the shortest path problem, see Appendix A.2. From hereon, we will call these shortest paths, optimal paths. Also in this context, since shortest paths and stable paths are identical, we abbreviate both by *sp*. For brevity of notation, the optimal path stability and the corresponding optimal path sets restricted to the local view, G_h^{local}, are denoted by $x_{ij}^{G_h^{local}} = x_{ij}^{h-local}$ and $P_{ij}^{G_h^{local}*} = P_{ij}^{h-local*}$.

Finally, note that this additive path stability metric follows the *Bellman's optimality principle*:

Lemma 2.3.1 *For any optimally stable path* $p = (i = u_1, u_2, \ldots, u_n = j) \in P_{ij}^{G'^*}$*, any sub-path* $(u_k, u_{k+1}, \ldots, u_l) \in {P_{u_k u_l}^{G'}}^*$ *for* $1 \leq k < l \leq n$*.*

Proof Suppose that the sub-path $(u_k, u_{k+1}, \ldots, u_l) \notin {P_{u_k u_l}^{G'}}^*$. Consider an alternative better path (of lower weight) $(u_k = v_1, v_2, \cdots, , v_m = u_l) \in {P_{u_k u_l}^{G'}}^*$, i.e., $w_{(v_1, v_2, \cdots, , v_m)} < w_{(u_k, u_{k+1}, \ldots, u_l)}$. Then the path

$p' = (i = u_1, u_2, \cdots, u_k = v_1, v_2, v_3, \cdots, v_m = u_l, u_{l+1}, \cdots, u_n)$ has a lower weight than p, i.e., $w_{p'} < w_p$. This is a contradiction.

The above lemma yields the following results w.r.t. h-locally optimal paths:

Lemma 2.3.2 *For a globally optimal* $p = (h = u_1, u_2, \ldots, u_n = j) \in P_{hj}^{G^*}$*, let* $p^{h-local} = (u_1, u_2, \ldots, u_l = \gamma_p^h)$ *be the sub-path from* h *to the first occurrence* γ_p^h*, i.e.,* $l = \min\{1 \leq s \leq n : \; u_s = \gamma_p^h\}$ *(shown in Fig. 2.6). Then* $p^{h-local} \in P_{h\gamma_p^h}^{h-local*}$*, i.e.,* $p^{h-local}$ *is a* h*-locally optimal path.*

Proof From Lemma 2.3.1, $w_{p^{h-local}} = x_{h\gamma_p^h}^G$. Since $P_{h\gamma_p^h}^{h-local*} \subseteq {P_{h\gamma_p^h}^G}^*$, $x_{h\gamma_p^h}^{h-local} \geq x_{h\gamma_p^h}^G$. Since $p^{h-local} \in P_{h\gamma_p^h}^{h-local} \supseteq P_{h\gamma_p^h}^{h-local*}$, $w_{p^{h-local}} \geq x_{h\gamma_p^h}^{h-local}$. Thus, $w_{p^{h-local}} = x_{h\gamma_p^h}^{h-local}$.

Lemma 2.3.3 *Let* $p = (h = u_1, u_2, \ldots, u_n = j) \in P_{hj}^{G^*}$ *be a globally optimal path in* G*, with* $p^{h-local} = (u_1, u_2, \ldots, u_l = \gamma_p^h)$ *(the sub-path from* h *to the first occurrence* γ_p^h*, see Fig. 2.6). Consider a* h*-locally optimal path* $p^{h-local'} = (h = v_1, v_2, \cdots, v_m = \gamma_p^h) \in P_{h\gamma_p^h}^{h-local*}$*. Then the path obtained by replacing* $p^{h-local}$ *with* $p^{h-local'}$ *in* p *is also globally optimal, i.e.,* $q = (h = v_1, v_2, \cdots, v_m = u_l = \gamma_p^h, u_{l+1}, \cdots, u_n) \in P_{hj}^{G^*}$*.*

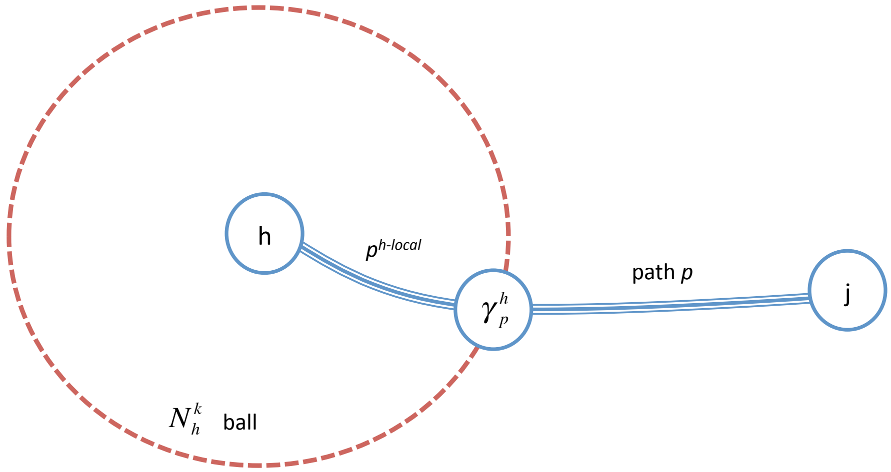


Figure 2.6: Illustrates the relation between a globally optimal path p and its sub-path $p^{h-local}$. $p^{h-local}$ is the sub-path from h to the first occurrence of γ_p^h.

Proof From Lemma 2.3.2, $w_{p^{h-local}} = x^G_{h\gamma_p^h} = x^{h-local}_{h\gamma_p^h} = w_{p^{h-local'}}$.

$$\begin{aligned} w_q &= w_{p^{h-local'}} + w_{(u_l,u_{l+1},\cdots,u_n)} \\ &= w_{p^{h-local}} + w_{(u_l,u_{l+1},\cdots,u_n)} \\ &= w_p. \end{aligned}$$

Note that the principle illustrated in Lemma 2.3.1 is the fundamental principle based on which the theoretical results of the this chapter are developed. This principle is expressed, in greater generality, in an algebraic framework in Chapter 3 and is used for the generalized topology control problem in Chapter 4. In essence, Lemma 2.3.1 is pivotal to this dissertation.

2.4 Local Pruning

Topology control by local pruning [60] can be viewed as a multi-agent graph optimization problem: The objective of each agent, $h \in V$, is to make use of the local neighborhood information, G_h^{local} (Subsection 2.3.2), to select a subset of the topology information, Ω_h^{pruned}, that is then broadcast to the network. This subset is chosen so that the resulting pruned graph preserves some global properties of the original graph. The non-triviality in these problems is in translating the global properties/requirements to the local properties/requirements.

Although there have been many local pruning algorithms [40, 11, 47, 60], there are few algorithms that guarantee global QoS given only local neighborhood information. Notable of them are the algorithms in [37, 61], which provide constructions of global Connected Dominating Sets (CDS) that satisfy some global constraints using only local neighborhood information. However, as illustrated in Appendix A.3, the CDS constructions do not guarantee path QoS. To the best of our knowledge, there are no constructions that guarantee global QoS of paths given only local neighborhood information. In this section, we present graph-pruning procedures that preserve sp paths for link-state routing.

2.4.1 Expressing Global Constraints

The fundamental pruning problem for each host, $h \in V$, is to construct a minimal pruned edge set, Ω_h^{pruned}, such that G_h^{global} preserves desired properties of G. This is an interesting multi-agent optimization problem where the objective function (finding a minimal pruned edge set) for each agent (host) depends only on local neighborhood information, G_h^{local}. However, the agents (hosts) together must satisfy a global constraint, i.e., the global view must preserve desired properties of G. Before we consider the optimization problem (of finding the minimal pruned edge set), we will mathematically express the global constraint for stable path topology control. This is non-trivial because the global constraint involves the global view, while the hosts have access to strictly their local view. We will introduce further notation to express this problem.

For sp routing, we want G_h^{global} to preserve stable routing paths (of G) from h to every other vertex $j \in V$. This can be expressed as a shortest path property $\pi_h^S(G)$ for $h \in V$:

$$\pi_h^S(G): \ \exists p \in P_{hj}^{G*}, \ j \in V. \tag{2.3}$$

Let $\Pi_h^S(G)$ denote all the subgraphs of G for which $\pi_h^S(G)$ holds, i.e., those that contain at-least one sp tree rooted at h. Then the desired stable path preserving global constraint can be expressed as

$$G_h^{global} \subseteq \Pi_h^S(G). \tag{2.4}$$

Although we have mathematically expressed the global constraint for stable path pruning, this constraint cannot be directly imposed on the local view; we need local constraints that will guarantee that the global constraints are satisfied. Remember, local pruning conditions should impose constraints on $\Omega_h^{pruned} \subseteq \Omega_h^G$, given G_h^{local} (Subsection 2.3.2). Next, we will formulate such local constraints that imply the desired global constraint (Equation (2.4)).

2.4.2 Local Pruning Conditions

The sp problem has an interesting structure that arises from Bellman's optimality principle (Lemma 2.3.1). Lemma 2.3.1 states that for a global shortest path, any sub-path is also a shortest path. This suggests that locally suboptimal paths can be discarded/pruned-away in computing the sp solutions. This can be represented formally by constraints on $\Omega_h^{pruned} \subseteq \Omega_h^G$.

Given the local view G_h^{local}, we can compute all the locally optimal paths (including multiplicities) from h to every boundary node, $j \in \partial N_h^k$, using any sp algorithm [32]; these paths are denoted by sets $P_{hj}^{h-local*}$, $j \in \partial N_h^k$. Note for any path $p \in P_{hj}^{h-local*}$, p begins with the edge (h, i_k) where $p_{(2)} = i_k$ is an one-hop neighbor of h, i.e., $i_k \in \partial N_h^1$. Consider the set of such edges that lie on any of the h-locally optimal paths to any of the boundary nodes, $j \in \partial N_h^k$. We denote them by the set

$$\Omega_h^{non-dominated} = \cup_{j \in \partial N_h^k} \cup_{p \in P_{hj}^{h-local*}} \{(h, p_{(2)})\}.$$

For example, consider the local view of h shown in Fig. 2.7, with a neighborhood size $k = 2$. The optimal paths to $j_1 \in \partial N_h^2$ are (h, i_1, i_2, j_1) and (h, i_3, j_1), and that to $j_2 \in \partial N_h^2$ are (h, i_3, j_2) and (h, i_4, j_2). The set $\Omega_h^{non-dominated} = \{(h, i_1), (h, i_3), (h, i_4)\}$ consists of all the edges which are on all these optimal paths. It does not contain the edge (h, i_2) because none of the h-locally optimal paths begin with the edge (h, i_2).

We have, now, introduced the required notations to define the local pruning conditions of interest. Given the local view, G_h^{local}, the local property $\pi_h^{\Omega-weak-S}(G_h^{local})$

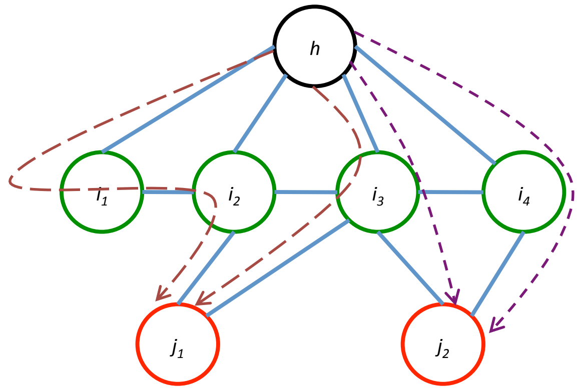


Figure 2.7: h-locally optimal paths in G_h^{local}. The boundary vertices are j_1 and j_2, and the h-locally optimal path sets are $P_{hj_1}^{h-local*} = \{(h, i_1, i_2, j_1), (h, i_3, j_1)\}$, and $P_{hj_2}^{h-local*} = \{(h, i_3, j_2), (h, i_4, j_2)\}$.

is said to hold for any $\Omega \in 2^{\Omega_h^G}$ if

$$\Omega_h^{non-dominated} \subseteq \Omega,$$

and let $\Pi_h^{\Omega-weak-S}(G_h^{local})$ denote the set of all such Ω's. The local pruning condition, which we are interested in, is given by

$$\Omega_h^{pruned} \in \Pi_h^{\Omega-weak-S}(G_h^{local}). \tag{2.5}$$

We will prove that if $\Omega_h^{pruned} \in \Pi_h^{\Omega-weak-S}(G_h^{local})$, then $G_h^{global} \in \Pi_h^S(G)$. Before we present the proof, we present the intuition behind this local condition. As we alluded to, in the start of this subsection, the primary structure for this condition is Bellman's optimality principle.

First, we show an analogy between the procedure by which $\Omega_h^{non-dominated}$ is constructed and the trajectory pruning procedure of online Viterbi decoding [26], which is a manifestation of Bellman's optimality principle at work. Viterbi decoder is an online maximum likelihood decoder for a convolutional encoder [26]: a convolutional encoder generates an output string of bits for an input string of bits (message from source); this corresponds to a trajectory, a sequence of states, in the encoder. The objective of the Viterbi decoder, at the receiver, is to detect this input string, or equivalently, the state trajectory, from the output string that is corrupted by noise. The underlying principle behind Viterbi decoder is finite state space dynamic programming, which is an algorithm that makes use of Bellman's optimality principle.

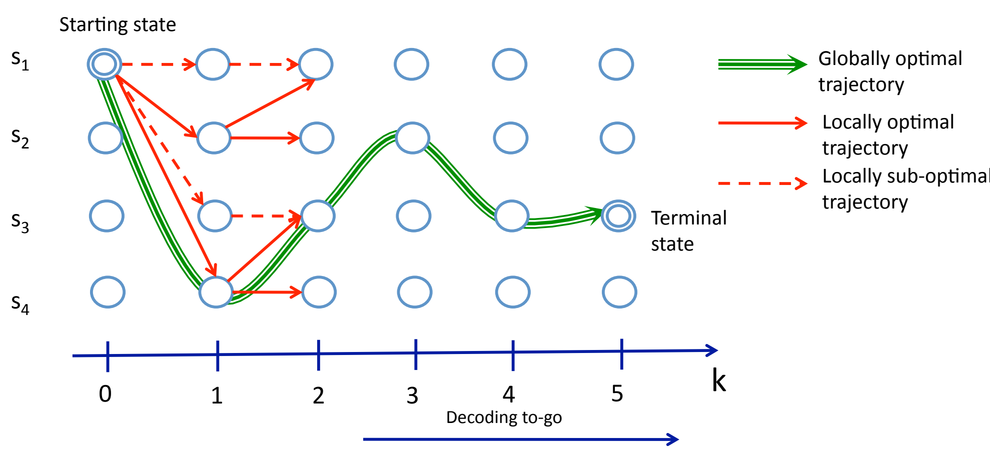


Figure 2.8: Viterbi decoding example: The figure illustrates the online trajectory pruning procedure in Viterbi decoding. The start state, s_1, and terminal state, s_3, are indicated by double-lined circles. The figure shows all locally optimal trajectories with solid lines. As examples, it shows two locally suboptimal trajectories (s_1, s_1, s_1) and (s_1, s_3, s_3).

To better illustrate the algorithm, we present an example in Fig. 2.8. The figure shows, what is called, a trellis diagram [26]. In this example, the convolutional encoder can be in one of the 4 states $\{s_1, s_2, s_3, s_4\}$. The start state is s_1, which is known at the receiver. The trajectory corresponding to the maximum likelihood solution is the trajectory/path $(s_1, s_4, s_3, s_2, s_3, s_3)$. The figure shows the online decoding stage $k = 2$. At $k = 2$, the decoder has an apriori information of only the first two bits of the corrupted output string, and consequently, can not make any decisions about the future, $k > 2$. However, within $k = 2$, the decoder prunes away some locally suboptimal trajectories. The figure shows all the locally optimal trajectories till $k = 2$ for each of the states, i.e., those sub-trajectories have the maximum likelihood from s_1 to each of s_1, s_2, s_3 and s_4 in two stages. Solid lines show these locally optimal sub-trajectories. As examples, we show, with dashed lines, two locally suboptimal trajectories (s_1, s_1, s_1) and (s_1, s_3, s_3), i.e.,

$$\begin{aligned} \text{likelihood}(s_1, s_1, s_1) &< \text{likelihood}(s_1, s_2, s_1), \\ \text{likelihood}(s_1, s_3, s_3) &< \text{likelihood}(s_1, s_4, s_3). \end{aligned}$$

These locally suboptimal trajectories can be pruned away, i.e., they need not be stored to compute the desired globally optimal trajectory. However, the locally optimal trajectories too every state (s_1, s_2, s_3, s_4) at $k = 2$ need to be stored for the global computation because any of these trajectories can be potentially a sub-trajectory of the globally optimal trajectory. The example of trajectory pruning in Viterbi decoding illustrates the necessity to preserve all the locally optimal paths, when we have only local information. This is the intuition for considering the set of paths $\cup_{j \in \partial N_h^k} P_{hj}^{h-local*}$, i.e. h-locally optimal paths to every boundary node.

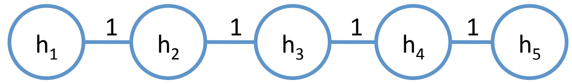


Figure 2.9: Example linear graph illustrating the local pruning condition

Although the trajectory pruning procedure in Viterbi decoding explains the need for preserving all locally optimal paths, it does not explain the local condition on Ω_h^{pruned} (Equation (2.5)). This local condition is a condition on the edges of Ω_h^G. To understand the intuition behind this local condition, consider a line graph shown in Fig. 2.9, where all edge-weights are 1. Let the size of the neighborhood be $k = 2$. In this example, only the hosts h_2 and h_3 are potentially responsible of selecting the edge (h_2, h_3) (Subsection 2.3.2). Consider the pruning policy at h_2. For h_2, $\partial N_{h_2}^k = \{h_4\}$, and (h_2, h_3, h_4) is the only path from h_2 to h_4. And for h_3, $\partial N_{h_3}^k = \{h_1, h_5\}$, and (h_3, h_2, h_1) is the only path from h_3 to h_1. If $(h_2, h_3) \notin \Omega_{h_2}^{pruned}$ and $(h_3, h_2) \notin \Omega_{h_3}^{pruned}$, then $(h_2, h_3) \notin G^{broadcast}$. Since $(h_2, h_3) \notin G_{h_5}^{local}$ and $(h_2, h_3) \notin G^{broadcast}$, $(h_2, h_3) \notin G_{h_5}^{global}$. Consequently, $G_{h_5}^{global}$ does not contain the globally optimal paths from h_5 to h_1 and h_2. This leads us to the following theorem.

Theorem 2.4.1 *If $\Omega_h^{pruned} \in \Pi_h^{\Omega-weak-S}(G_h^{local})$, then $G_h^{global} \in \Pi_h^S(G)$.*

Proof If $\Omega_h^{pruned} \in \Pi_h^{\Omega-weak-S}(G_h^{local})$, we will show that for any $j \in V$, all globally optimal paths from h to j are contained in G_h^{global}. This implies $G_h^{global} \in \Pi_h^S(G)$. Consider $p = (h = u_1, u_2, \cdots, u_n = j) \in P_{hj}^{G*}$. There are two cases:

Case 1: p is completely contained in G_h^{local}. Then p is contained in G_h^{global} because $G_h^{global} = G_h^{local} \cup G_broadcast.$

Case 2: p is not completely contained in G_h^{local}. We will prove by construction that p is preserved in G_h^{global}. For some $1 < l < n$, the sub-path $(u_1, u_2, \cdots, u_l)$ is completely contained in G_h^{local}, and consequently, it is contained in G_h^{global} (Case 1). We will show that the remnant path $(u_{l+1}, u_{l+2}, \cdots, u_n)$ is contained in $G^{broadcast}$. For any $l < m \leq n$, consider the local view at u_m, $G_{u_m}^{local}$. Consider the upstream of sub-path p from u_m to $h = u_1$, $q = (u_m, u_{m-1}, \cdots, u_1)$. By Lemma 2.3.1, q is globally optimal, i.e., $q \in {P_{u_m u_1}^{G}}^*$. Consider the sub-path of q from u_m to the first occurrence of $\gamma_q^{u_m}$, $(u_m, u_{m-1}, \cdots, \gamma_q^{u_m})$. By Lemma 2.3.2, this sub-path is u_m-locally optimal, $(u_m, u_{m-1}, \cdots, \gamma_q^{u_m}) \in {P_{u_m \gamma_q^{u_m}}}^*$. Then, $(u_m, u_{m-1}) \in \Omega_{u_m}^{non-dominated}$. Since $\Omega_{u_m}^{pruned} \in \Pi_{u_m}^{\Omega-weak-S}(G_h^{local})$, $(u_m, u_{m-1}) \in \Omega_{u_m}^{pruned}$.

This argument holds for $l < m \leq n$, and consequently, the sub-path $(u_{l+1}, u_{l+2}, \cdots, u_n)$ is preserved.

The proof shows that entire set of the optimal paths from h to any destination $j \in V$, P_{hj}^{G*}, is preserved in G_h^{global} if the local pruning condition of Equation (2.5) holds. This is more than sufficient for hop-by-hop routing [51]: for hop-by-hop routing every host, h, must be able to compute a best next-hop for every destination, $j \in V$. So we need not preserve the entire set P_{hj}^{G*}; preserving a subset of P_{hj}^{G*} would suffice. The local pruning condition of Equation (2.5) requires that all locally optimal paths to be preserved, and consequently, gives lesser freedom to prune. It is for this reason that we call it a "weak pruning condition."

We now present a stronger local pruning condition. This stronger condition requires to preserve at-least one locally optimal path to every boundary node, rather than all the locally optimal paths. It is expressed by another local property denoted by $\pi_h^{\Omega-S}$ at $h \in V$: given the local view, G_h^{local}, the property $\pi_h^{\Omega-S}(G_h^{local})$ is said to hold for an edge set $\Omega \in 2^{\Omega_h^G}$ if for all $j \in \partial N_h^k$ there exists a path $p \in P_{hj}^{h-local*}$ such that $(h, p_{(2)}) \in \Omega$. Let $\Pi_h^{\Omega-S}(G_h^{local})$ denote the set of all such Ω's for which $\pi_h^{\Omega-S}(G_h^{local})$ holds. i.e.,

$$\Pi_h^{\Omega-S}(G_h^{local}) = \{\Omega \in 2^{\Omega_h^G} \ : \ (\cap_{j \in \partial N_h^k}\{p \in P_{hj}^{h-local*} \ : \ (h, p_{(2)}) \in \Omega\}) \neq \emptyset\}.$$

The strong local pruning condition is given by

$$\Omega_h^{pruned} \in \Pi_h^{\Omega-S}(G_h^{local}). \tag{2.6}$$

Note that $\Pi_h^{\Omega-weak-S}(G_h^{local}) \subseteq \Pi_h^{\Omega-S}(G_h^{local})$, and consequently, the strong pruning conditions gives greater freedom to prune. To illustrate this stronger pruning condition, we revisit the example of Fig. 2.7. Remember, the locally optimal paths from h to j_1 are (h, i_1, i_2, j_1) and (h, i_3, j_1) and that from h to j_2 are (h, i_3, j_2) and (h, i_4, j_2). Clearly, $\Omega_h^{non-dominated} = \{(h, i_1), (h, i_3), (h, i_4)\} \in \Pi_h^{\Omega-S}(G_h^{local})$ because it intersects with all the locally optimal paths. Interestingly, smaller subsets $\{(h, i_1), (h, i_3)\}, \{(h, i_1), (h, i_4)\}$, $\{(h, i_3))\}, \{(h, i_3), (h, i_4)\} \in \Pi_h^{\Omega-S}(G_h^{local})$ because they intersect with at-least one h-locally optimal path to j_1 and j_2. As the example illustrates, the stronger pruning condition provides a greater degree of pruning.

However, this local condition is not sufficient to ensure $G_h^{global} \in \Pi_h^S(G)$, since it does not guarantee *loop-freedom*. This is a well-known problem in distributed routing protocols [45]: Loops typically occur in distributed graph algorithms when tie-breaking mechanisms are not employed. Using an example, we illustrate that a similar problem is likely to occur in stable path distributed pruning without tie-breaking.

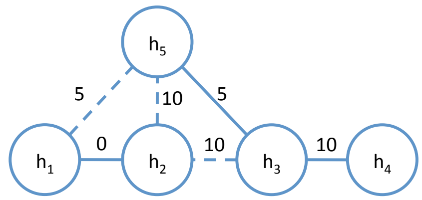


Figure 2.10: Loopy pruning: Shows an edge-weighted graph, with an instance of the strong pruning policy, $\Omega_h^{pruned} \in \Pi_h^{\Omega-S}(G_h^{local})$, satisfied at every $h \in V$. The edges chosen for broadcast, $E^{broadcast}$, are indicated by solid lines and those that are pruned away are indicated by dashed lines.

Consider an example edge-weighted graph shown in Fig. 2.10, where the edge-weights correspond to some link stability metric (Section 2.2.1). Consider any stable path pruning policy at stations $h_1, h_2, \ldots, h_5$. Let the size of the neighborhood be $k = 2$. The neighborhood boundary sets are $\partial N_{h_1}^2 = \{h_3\}$, $\partial N_{h_2}^2 = \{h_4\}$, $\partial N_{h_3}^2 = \{h_1\}$, $\partial N_{h_4}^2 = \{h_2, h_5\}$ and $\partial N_{h_5}^2 = \{h_4\}$. If the pruning mechanisms at these stations satisfy the strong local pruning condition (Equation (2.6)), then h_4 choses (h_4, h_3) and h_5 choses (h_5, h_3). However, the pruning mechanisms at h_1, h_2 and h_3 have multiple optimal paths to choose from. For h_1 to reach h_3, there are two optimal paths, (h_1, h_2, h_3) and (h_1, h_5, h_3). For h_2 to reach h_4, there are two optimal paths, (h_2, h_3, h_4) and $(h_2, h_1, h_5, h_3, h_4)$. For h_3 to reach h_1, there are two optimal paths, (h_3, h_2, h_1) and (h_3, h_5, h_1). The Fig. 2.10 illustrates one pruning policy that satisfies the necessary conditions: h_1 chooses (h_1, h_2) for path (h_1, h_2, h_3), h_2 chooses (h_2, h_1) for path $(h_2, h_1, h_5, h_3, h_4)$, and h_3 chooses (h_3, h_5) for path (h_3, h_5, h_1). The pruned graph $G^{broadcast}$, shown in the Figure 2.10, is then disconnected! Clearly, the distributed pruning does not preserve the stable optimal paths in the different global views.

2.4.3 Positivity Assumption and Sufficiency for Strong Pruning

The example of Fig. 2.10 suggests a sufficient condition, which we call the *positivity condition*: all the edge weights are strictly positive, $a_{uv} > 0,\ (u, v) \in E$. We will show that under the positivity assumptions, the strong pruning conditions (Equation (2.6)) become sufficient to ensure that optimal paths from a host to every destination vertex are preserved in its global view.

Theorem 2.4.2 *Under the positivity assumption,*$a_{uv} > 0$, $(u, v) \in E$, *for all* $h \in V$, *if* $\Omega_h^{pruned} \in \Pi_h^{\Omega-S}(G_h^{local})$, *then* $G_h^{global} \in \Pi_h^S(G)$.

Proof We need to show that G_h^{global} has at-least one optimal path to any vertex $j \in V$. We will prove by construction that one optimal path is preserved under pruning. Similar to the proof of Theorem 2.4.1, there are two cases.

Case 1: $p \in P_{hj}^G*$ is contained in G_h^{local}, i.e., $p \in P_{hj}^{h-local}$. Then p is contained in G_h^{global} because $G_h^{global} = G_h^{local} \cup G^{broadcast}$.

Case 2: No $p \in P_{hj}^G*$ is completely contained in G_h^{local}. We will prove by construction that a reverse path $p \in P_{jh}^G*$ is contained in G_h^{global}. Consider the pruning at $j = u_1$. Let $p \in {P_{u_1h}^G}^*$ be any globally optimal path from u_1 to h.

Consider the sub-path of p from u_1 to the first occurrence $\gamma_p^{u_1}$, $p^{u_1-local}$. Denote the remnant of the path from $\gamma_p^{u_1}$ by $p^{u_1-not-local}$. Since the strong pruning condition, $\Omega_{u_1}^{pruned} \in \Pi_{u_1}^{\Omega-S}(G_h^{local})$, holds at u_1, there exists a path $q \in P_{u_1\gamma_p^{u_1}}{}^*$ such that $(u_1, q_{(2)}) \in \Omega_{u_1}^{pruned}$. Let $q_{(2)} = u_2$, which we call the pruned next-hop of u_1 to reach h. This is step is illustrated in Fig. 2.11.

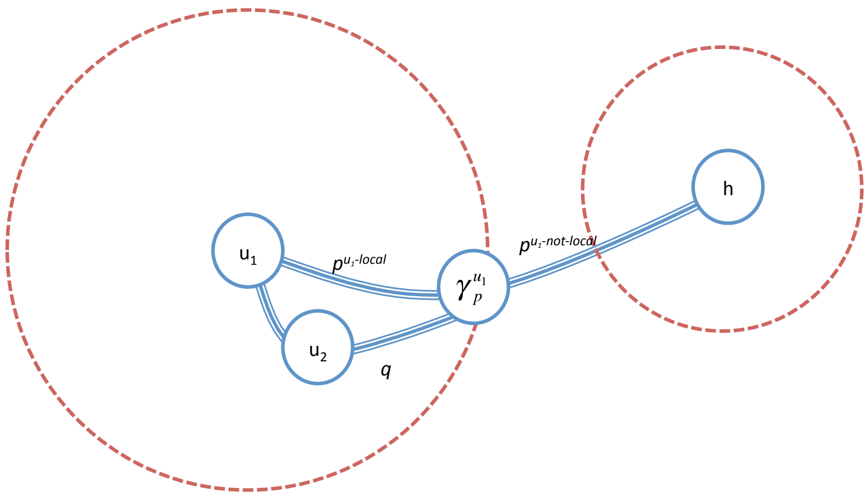


Figure 2.11: Path construction procedure: shows a globally optimal path p from u_1 to h. $p = p^{u_1-local}.p^{u_1-not-local}$ is a concatenation of $p^{u_1-local}$ from u_1 to $\gamma_p^{u_1}$ and $p^{u_1-not-local}$, the remnant path. q is u_1-locally optimal path from u_1 to $\gamma_p^{u_1}$ and $u_2 = q_{(2)}$.

By Lemma 2.3.3, the concatenation of the q and $p^{u_1-not-local}$ is globally optimal, i.e., $q.p^{u_1-not-local} \in P_{u_1h}^{G}{}^*$. Then $w_{q.p^{u_1-not-local}} = x_{u_1h}^G = a_{u_1u_2} + x_{u_2h}^G$. Since

$a_{u_1u_2} > 0$, we have $x^G_{u_1h} > x^G_{u_2h}$.

Repeating the above construction, we get a sequence of vertices $u_1, u_2, u_3, \cdots, u_r$, where u_{s+1} is the pruned next-hop of u_s to reach h. Then $x^G_{u_1h} > x^G_{u_2h} > x^G_{u_3h} > \cdots > x^G_{u_rh}$. This implies that there can be no loops in pruning, i.e., $u_r \neq u_s, 1 \leq s < r$ (if $u_r = u_s$, then $x^G_{u_rh} = x^G_{u_sh}$, which is a contradiction). Since there are no loops (each u_r is unique) and V is finite, there exists a u_l in the construction such that $u_l \in \partial N^k_h$. There exists a h-locally optimal path from h to u_l in G^{local}_h, say $(h = u_n, u_{n-1}, \cdots, u_l) \in P^{h-local*}_{hj}$. We have proved by construction that $(h = u_n, u_{n-1}, \cdots, u_1 = j)$ is preserved in G^{global}_h.

2.4.4 Degrees of local neighborhood

As alluded to in Section 1.3, in Chapter 1, we study the degree of pruning possible with increasing degrees of local neighborhood information. In the theory developed so-far, for local pruning, we have considered the neighborhood size, k, to be fixed. In this subsection, we present results for the effectiveness of local pruning with increasing k. To present these results, we parameterize the local view with the neighborhood size; we denote by $G^{local}_h(k)$, the local view with a neighborhood size k. Similarly, we denote the h-locally optimal paths and solutions by $P^{h-local*}_{hj}(k)$ and $x^{h-local}_{hj}(k)$ as functions of k.

We illustrated in Subsection 2.4.2 that the strong pruning condition allows for better pruning. This was a consequence of the relation $\Pi^{\Omega-weak-S}_h(G^{local}_h) \subseteq \Pi^{\Omega-S}_h(G^{local}_h)$; we also showed that $\Pi^{\Omega-S}_h(G^{local}_h)$ contained smaller subsets of Ω^G_h.

This suggests that the inclusion relation induces an order, which we call the degree of pruning δ. For any two pruning policies, on Ω_h^G, π_i and π_j, ordered by an inclusion relation, $\Pi_i \subseteq \Pi_j$, where Π_i and Π_j are the subsets of Ω_h^G that satisfy π_i and π_j respectively, the degree of pruning is related by $\delta(\pi_i) \leq \delta(\pi_j)$. In other words, there is greater degree of pruning for a policy that is stronger.

Remember, $\Omega \in \Pi_h^{\Omega-S}(G_h^{local}(k))$ implies that for every $j \in \partial N_h^k$, there exists a $p \in P_{hj}^{h-local*}(k)$ such that $(h, p_{(2)}) \in \Omega$. The monotonicity of the degree of pruning with the neighborhood size is characterized by the following results.

Lemma 2.4.3 $\Pi_h^{\Omega-S}(G_h^{local}(k)) \subseteq \Pi_h^{\Omega-S}(G_h^{local}(k+1))$, $k \geq 1$.

Proof Let $\Omega \in \Pi_h^{\Omega-S}(G_h^{local}(k))$. To show that $\Pi_h^{\Omega-S}(G_h^{local}(k)) \subseteq \Pi_h^{\Omega-S}(G_h^{local}(k+1))$, we need to show that $\Omega \in \Pi_h^{\Omega-S}(G_h^{local}(k+1))$, i.e., for every $j \in \partial N_h^{k+1}$, there exists a $p \in P_{hj}^{h-local*}(k+1)$ such that $(h, p_{(2)}) \in \Omega$.

Consider a $p \in P_{hj}^{h-local*}(k+1)$, $j \in \partial N_h^{k+1}$. The path can be decomposed into $p = p'.p''$, where $p' = (h = u_1, u_2, \cdots, u_l)$ is contained in $G_h^{local}(k)$, and p'' is the remnant path. By Lemma 2.3.1, $p' \in P_{hu_l}^{h-local*}(k)$. Since $\Omega \in \Pi_h^{\Omega-S}(G_h^{local}(k))$, there exists a $q' \in P_{hu_l}^{h-local*}(k)$ such that $(h, q'_{(2)}) \in \Omega$. Consequently for every $p \in P_{hj}^{h-local*}(k+1)$, $j \in \partial N_h^{k+1}$, there exists an alternative (optimal) path $q'.p''$ such that $(h, q'_{(2)}) \in \Omega$. This implies $\Omega \in \Pi_h^{\Omega-S}(G_h^{local}(k+1))$

Theorem 2.4.4 $\Pi_h^{\Omega-S}(G_h^{local}(k)) \subseteq \Pi_h^{\Omega-S}(G_h^{local}(k'))$, $1 \geq k \geq k'$.

Proof Proof follows directly from Lemma 2.4.3.

Corollary 2.4.5 $\delta(\pi_h^{\Omega-S}(G_h^{local})(k)) \leq \delta(\pi_h^{\Omega-S}(G_h^{local})(k'))$, $1 \geq k \geq k'$.

Note that the greater degree of pruning with increased degree of local neighborhood information comes at the cost of more complex neighbor discovery protocols.

2.4.5 Salient Features of Stable Path Topology Control

Note all the results in the section are a consequence of Lemma 2.3.1 and its derivatives. There are two underlying principles that are made use of in deriving the results:

1. Distribution of sub-optimality: Both in the weak pruning and strong pruning conditions, the means by which the local description is translated to a global description is based pruning of suboptimal paths. Thus sub-optimality, in a loose sense, distributes across paths: a locally suboptimal cannot be globally optimal.

2. Positivity and Downstream: The positivity assumption ensures that the pruning mechanism constructs path by going downstream, i.e., subsequent next hops of pruning always progress towards the host.

In this dissertation, we formalize the above two observations. In the later chapters, we will introduce generic rule-based routing schemes, which are described using semiring algebras. We will show that the above two observations translate to

1. Distribution of order, and

2. Inflatory arc composition

in the generalized path problem framework. In essence, these two principles provide the framework on which topology control algorithms for generic rule-based routing are developed in this dissertation.

2.4.6 Optimal Pruning as a Local Set-Cover Problem

The topology control problem for stable path routing boils down to selecting a minimal pruned edge-set such that local pruning conditions are satisfied:

$$\begin{aligned} \min_{\Omega_h^{pruned} \subseteq \Omega_h^G} \quad & |\Omega_h^{pruned}| \\ \text{subject to} \quad & \Omega_h^{pruned} \in \Pi_h^{\Omega-S}(G_h^{local}). \end{aligned} \tag{2.7}$$

Listing out all feasible subsets $\Omega_h^{pruned} \subseteq \Pi_h^{\Omega-S}(G_h^{local})$, in general, is computationally intractable. We will show that this problem can be reduced to a set-cover problem. To formulate this set-cover problem, we introduce further notation. Let $\zeta_h : \partial N_h^1 \to 2^{\partial N_h^k}$ denote the covering function: for $i \in \partial N_h^1$ and

$$\zeta_h(i) = \{j \in \partial N_h^k : \ \exists p \in P_{hj}^{h-local*} \text{ such that } i = p_{(2)}\},$$

i.e., $\zeta_h(i),\ i \in \partial N_h^1$ is the set of boundary nodes which are reachable via locally optimal paths starting with the edge (h, i). The corresponding inverse function $\zeta_h^{-1} : \partial N_h^k \to 2^{\partial N_h^1}$ is, for $j \in \partial N_h^k$,

$$\zeta_h^{-1}(j) = \{i \in \partial N_h^1 : \ j \in \zeta_h(i)\}.$$

This function ζ_h can be computed efficiently using any shortest path procedures [32] (see Section 2.5). Then the set-cover problem is

$$\min_{\Delta \in 2^{\partial N_h^1}} \quad |\Delta| \tag{2.8}$$
$$\text{subject to} \quad \cup_{i \in \Delta} \zeta_h(i) = \partial N_h^k.$$

Theorem 2.4.6 *For any minimizer Δ^* of the problem in Equation (2.8), $\{(h,i) : i \in \Delta^*\}$ solves the minimal pruning problem of Equation (2.7).*

Proof Since $\cup_{i\in\Delta}\zeta_h(i) = \partial N_h^k$, $\Omega_h^{pruned}) = \{(h,i) :\ i \in \Delta^*\} \in \Pi_h^{\Omega-S}(G_h^{local})$.

2.5 Stable Path Topology Control Algorithm

In this section, we present the Stable Path Topology Control (SPTC) algorithm that solves the set-cover problem (to an approximation) in Equation (2.8) introduced in Subsection 2.4.6. Finally, we demonstrate the performance of the SPTC algorithm by using the ETX metric (see Subsection 2.2.1).

2.5.1 Computing ζ_h

Algorithm 1 computes the covering function ζ_h used in the set-cover formulation (Equation (2.8)). The local view G_h^{local} is input to the algorithm and it outputs ζ_h. Given G_h^{local}, the function `computeAllPairSPFloydWarshall`, used in Algorithm 1, computes the all pair shortest paths (h-locally optimal paths) in the exclusive neighborhood N_h^{k-}, using the well-know *Floyd-Warshall* algorithm [32]. It returns a matrix $\mathbb{SP}_{N_h^{k-}}$ that yields the shortest (h-locally optimal restricted to N_h^{k-}) path metrics. The next step of Algorithm 1 is called vertex expansion at h.

The locally optimal paths to every boundary node, $j \in \partial N_h^k$ from h is computed and the one-hop neighbor ,$i \in \partial N_h^1$ on these optimal paths is stored in $\zeta_h^{-1}(j)$. Note that each element $\zeta_h^{-1}(j)$ is a set of one-hop neighbors.

Algorithm 1 Compute covering function ζ_h at $h \in V$

INPUT: G_h^{local}

//Compute all-pair-shortest paths in exclusive neighborhood

$SP_{N_h^k-} \leftarrow$ `computeAllPairSPFloydWarshall`(G_h^{local});

//Vertex expansion

for all $j \in \partial N_h^k$ **do**

$\zeta_h^{-1}(j) \leftarrow arg\min_{i \in \partial N_h^1} a_{h,i} + SP_{N_h^k-}(i,j)$;

end for

OUTPUT: ζ_h

2.5.2 Greedy Approximation Algorithm to Solve Set-Cover Problem

Given ζ_h, Algorithm 2 is a greedy algorithm that approximately solves Equation (2.8). The algorithm first extracts the essential cover elements and appends it to the set $R_{h-greedy}$. A one-hop neighbor i is an essential cover for a boundary node j if there are no other one-hop neighbors which lie on the locally optimal paths to j, i.e., $\zeta_h(i) = \{j\}$. In the next step, the Algorithm 2 performs a recursive greedy step, where it selects that one-hop neighbor i, which covers the most uncovered boundary nodes. The recursive procedure terminates when all the boundary nodes are covered.

Algorithm 2 Greedy Set-Cover Algorithm at h

INPUT: $\zeta_h, G_h, \partial N_h^1, \partial N_h^k$

INIT: $R_{h-greedy} \leftarrow \emptyset$, $U \leftarrow \partial N_h^k$;

// Find and append essential cover elements

for all $\{j \in \partial N_h^k : \ |\zeta_h^{-1}(j)| = 1 \}$ **do**

  $R_{h-greedy} \leftarrow R_{h-greedy} \cup \zeta_h^{-1}(j)$;

  $U \leftarrow U \backslash \{j\}$;

end for

// Greedy selection

while $U \neq \emptyset$ **do**

  $i^* \leftarrow \arg\max_{i \in \partial N_h^1} |\{j \in U : \ j \in \zeta_h(i)\}|$

  $R_{h-greedy} \leftarrow R_{h-greedy} \cup \{i^*\}$

  $U \leftarrow U \backslash \{j \in U : \ i^* \in \zeta_h(j)\}$

end while

Output: $R_{h-greedy}$

Algorithm 2 is adapted from a set-cover approximation algorithm illustrated in Chapter 11 of [32]. Let $d_h^* = \max_{i \in \partial N_h^1} |\zeta_h(i)|$. Then the following Lemma gives the approximation bounds for the greedy solution $R_{h-greedy}$:

Lemma 2.5.1 *Let the optimal solution to Equation (2.8) be Δ_h^* and $R_{h-greedy}$ be the output of Algorithm 2 at host h, then $|R_{h-greedy}| \leq H(d_h^*)|\Delta_h^*|$, where $H(N) = \sum_{n=1}^{N} \frac{1}{n}$.*

This lemma is proved in Chapter 11 of [32]. We call the Algorithms 1 and 2 together as the SPTC algorithm.

2.5.3 Simulation Setup

All simulations were carried out in OPNET Modeler 14.5 [41]. For the simulations, the mobile node model *manet station advanced* was chosen. The parameters given in Table 2.1 were used in the simulations.

We modified the default code for the OLSR model, which is an OLSR version 1 [12] implementation. We made suitable modifications to the neighbor discovery mechanism, as per [47], to compute the ETX metric online. We also modified the MPR selection algorithms to implement the SPTC algorithm. For the simulations, we used $k = 2$, size of the neighborhood.

To study the performance of the SPTC algorithm, we compare it with the OLSR implementation in [47], which uses the ETX metric to select MPRs using set-cover methods (illustrated in Subsection 2.2.2). We implemented both the OLSR and the SPTC algorithm to use the ETX metric. We call these two implementations, OLSR-ETX and SPTC-ETX respectively.

In the simulations, we compared both the data traffic carrying and Topology Control (TC) overhead performance of SPTC-ETX and OLSR-ETX. For the data

Group	Parameter	Value
MAC and PHY	Protocol	802.11b
	Transmission Rate	11 Mbps
	Transmit Power	5 mW
	Receiver Sensitivity	-95 dBm
	Error Correction Capabilities	None
Routing and TC	Protocol	OLSR-ETX or SPTC-ETX
	HELLO message interval	2 s
	Neighbor hold time	32 s
	TC message interval	5 s
ETX Computation	ETX Memory Length	32 s
	ETX Memory Interval	2 s
	ETX Hello Timeout Expiry	2.5 s
Traffic	Type	UDP CBR
	Packet length	1024 bits

Table 2.1: Parameters for simulation

traffic carrying performance, we studied the *carried load* for various *offered loads*. We set up a *UDP* traffic generator that sends *Constant Bit Rate* (CBR) traffic between pairs of stations. We then swept across this CBR rate to study the traffic performance with OLSR-ETX and SPTC-ETX.

In link state mechanisms such as OLSR, the TC broadcast mechanism is proactive, and consequently not all TC messages broadcast correspond to topology changes. To study the overhead due to topology changes, we measured the rate of reactive TC messages and the total number of actual topology changes. Reactive TC messages are those that are generated due to changes in the selected topology. This is a good estimate of the actual topology overhead for the pruned network. We will compare this topology control overhead for SPTC-ETX against that of OLSR-ETX.

2.5.4 Scenario Illustrating CDS Limitations

Before we present the results for complicated topologies, we will study the performance of SPTC-ETX and OLSR-ETX for a simple topology shown in Figure 2.12. This scenario corresponds to the example topology in Subsection 2.2.2 that illustrates the fundamental limitation of CDS constructions. The topology is set up such that there is a long-distance unstable wireless link between *manet_0* and *manet_1*. All other links are short and hence more stable compared to *(manet_0,manet_1)*.

Consider OLSR-ETX's MPR selection process at node *manet_0*. Since the link *(manet_0,manet_1)* is unstable, it goes ON and OFF frequently. Whenever

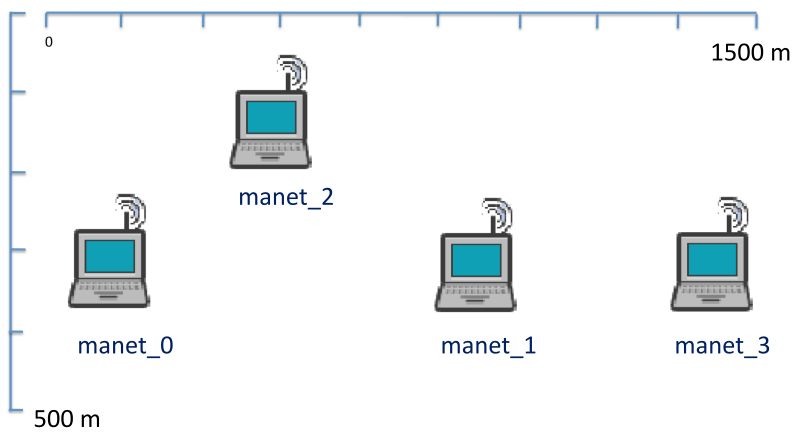


Figure 2.12: 4 node topology to illustrate the limitation of CDS constructions

this link is ON, $manet\_3 \in \partial N^2_{manet\_0}$. The OLSR-ETX's set-cover construction, selects the unstable link *(manet_0,manet_1)* as $\Omega^{pruned}_{manet\_0}$ because it the only edge in the two-hop path to reach *manet_3*. When the link *(manet_0,manet_1)* is OFF, $manet\_1 \in \partial N^2_{manet\_0}$, and consequently, OLSR-ETX chooses *(manet_0,manet_2)* as $\Omega^{pruned}_{manet\_0}$. Thus as the unstable link *(manet_0,manet_1)* goes ON and OFF, $\Omega^{pruned}_{manet\_0}$ oscillates between *(manet_0,manet_1)* and *(manet_0,manet_2)*. This is illustrated in Figure 2.13, which shows a realization of the topology selection process at *manet_0* obtained by OPNET simulation.

Fundamental limitation of the set-cover OLSR construction is that it is not designed to exploit the local path diversity. This limitation is overcome by the SPTC-ETX that provides a more stable $\Omega^{pruned}_{manet\_0}$. From simulations we observed that SPTC-ETX almost always chooses $\Omega^{pruned}_{manet\_0} = \{(manet\_0, manet\_2)\}$ and $\Omega^{pruned}_{manet\_2} =$

{*(manet_0,manet_1)*}, thus preserving the stable path *(manet_0,manet_2,manet_1)*. For a simulation period of 1 *hour*, we observed 96 topology changes for OLSR-ETX and 6 for SPTC-ETX.

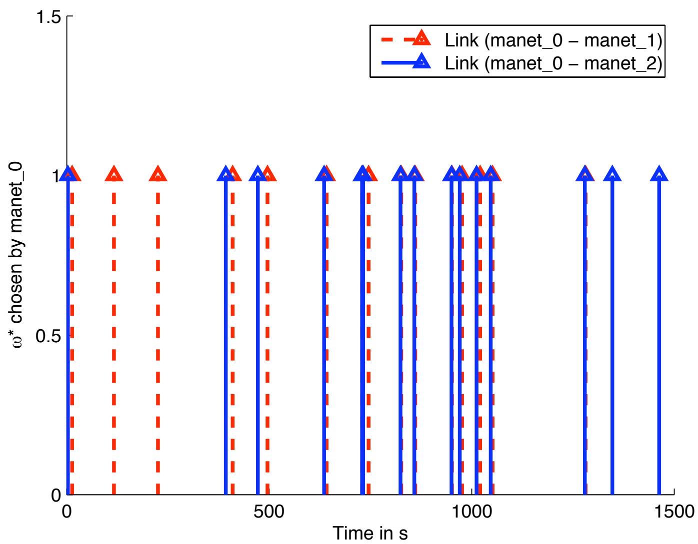


Figure 2.13: Topology selection process at manet_0 for OLSR-ETX

2.5.5 Static Grid Topology

The next topology that we consider is a 100-node static grid topology shown in Figure 2.14. The network consists of many stable and unstable links. This topology suffers from the same problem explained in Subsection 2.5.4. The one-hop neighbors that are far off (in physical distance) typically cover more two-hop neighbors. However, by the nature of radio propagation, these links are unstable.

The UDP CBR traffic is sent between 5 different random source-destination pairs. The comparison of the traffic-carrying performance of SPTC-ETX and OLSR-

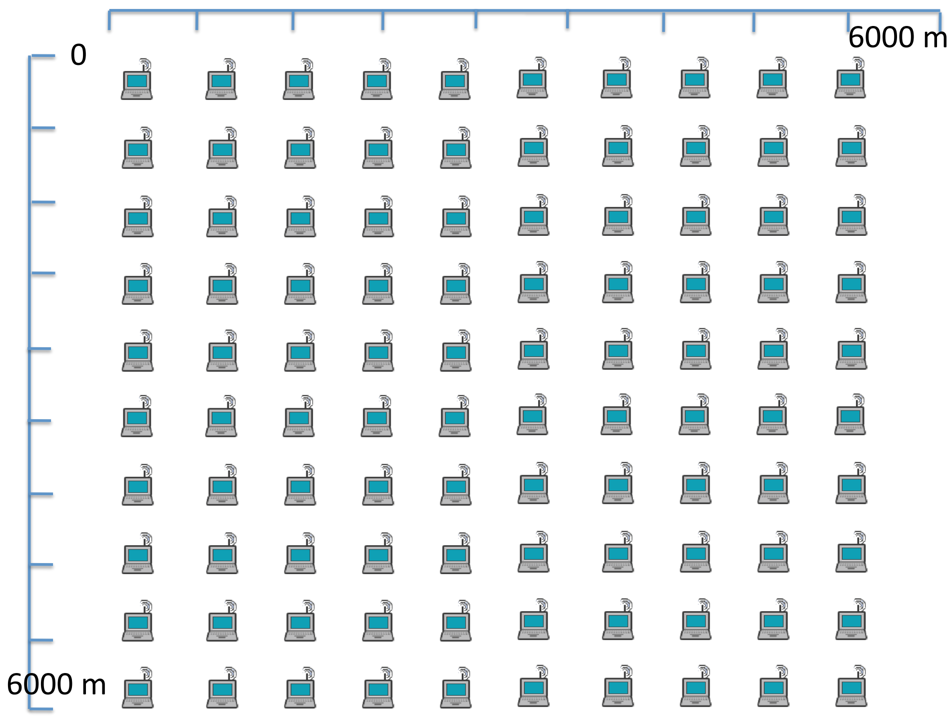


Figure 2.14: 100-node grid network

ETX is shown in Fig. 2.15. The simulation results indicate that SPTC-ETX has a saturation capacity of $86kbps$, while that of OLSR-ETX is $75kbps$.

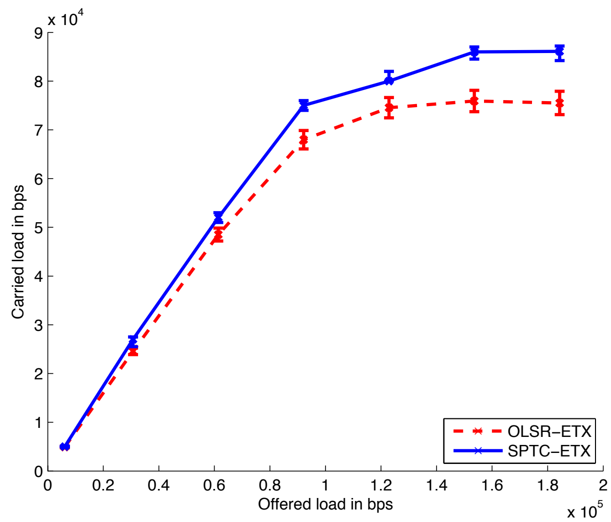


Figure 2.15: Carried load vs. Offered load for 100-node grid shown with 95% confidence intervals

The average number of total topology changes (for many runs of the simulation) was 11254 and 8280 for OLSR-ETX and SPTC-ETX respectively. The corresponding rate of reactive TC messages was $930bps$ and $681bps$ respectively. This implies that the pruned subnetwork of SPTC-ETX is stable/long-lived compared to that of OLSR-ETX.

2.5.6 Random Waypoint Mobility Scenario

Random waypoint mobility pattern is a commonly used to study protocol performances in a mobile environment [7]. The mobility parameters that we used for the random waypoint mobility pattern are shown in Table 2.2. All statistics were

Parameter	Value
No of stations	25
Simulation Area	3000m $\times$ 3000m
Speed	(5,20] m/s
Pause time	0 s

Table 2.2: Random waypoint mobility parameters

collected once the simulations reached stochastic stationarity.

Again, UDP CBR traffic is sent between two different random source-destination pairs. The sample mean of the carried load as a function of the offered load is shown in Fig. 2.16. We observe that SPTC-ETX is capable of carrying 13% more load than OLSR-ETX. This is because in OLSR, we observed that significantly more traffic is routed through unstable links.

The average number of topology changes was 80124 and 60874 in one hour of simulation time for OLSR-ETX and SPTC-ETX respectively. The corresponding rate of reactive TC messages was $8.3kpbs$ and $6kpbs$ respectively.

2.5.7 Battlefield Scenario

Finally, we consider a battlefield scenario, introduced in [5], with an initial topology as shown in Sub-figure 2.17a. It comprises of 3 platoons of stations: Platoon A consists of nodes 0 to 9, Platoon B consists of nodes 10 to 19, and Platoon C consists of nodes 20 to 29. The three platoons move in the trajectories shown in

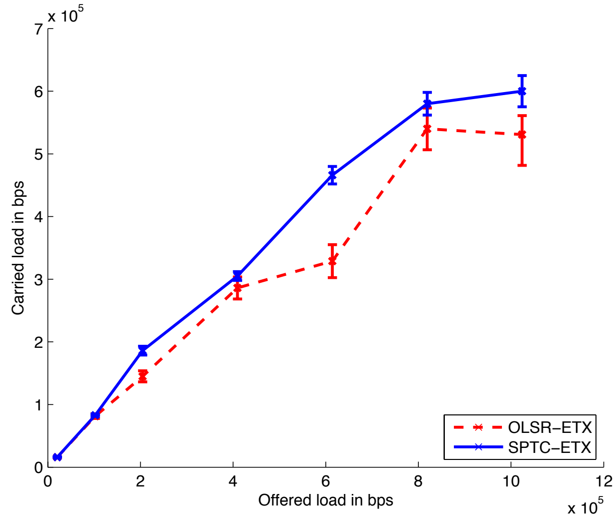


Figure 2.16: Carried load vs. Offered load of Random Waypoint shown with 95 % confidence intervals

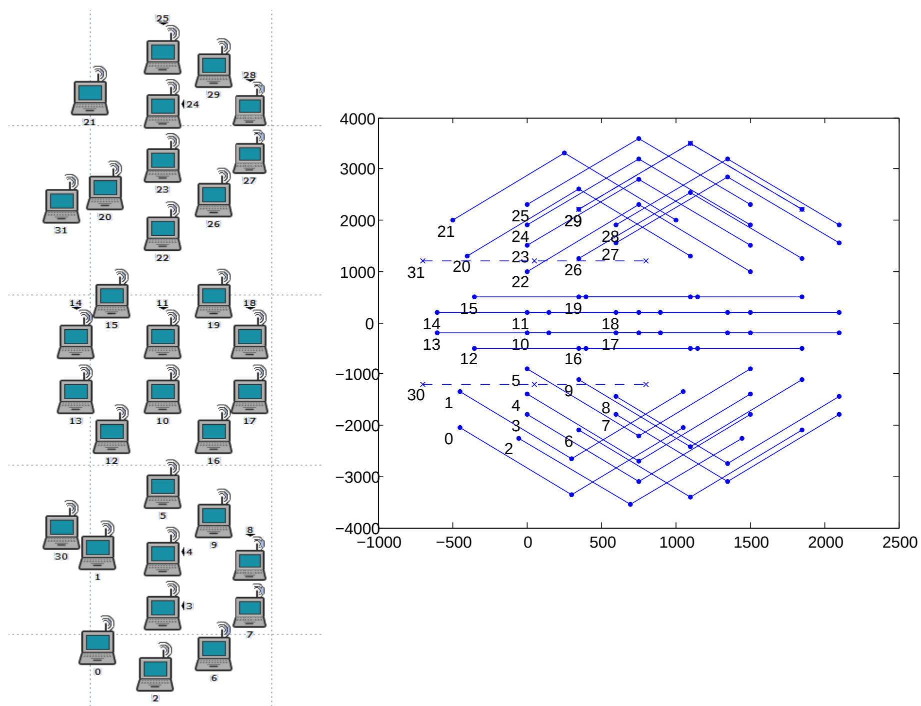
Sub-figure 2.17b: Platoon B moves forward along the x direction, and Platoons A and C move forward and away from platoon B at speed of 1.5 m/s in the y direction. Then the platoons move together back to the initial formation. To ensure better connectivity among the platoons, two supporting nodes 30 (to support connections between Platoon A and B) and 31 (to support connections between Platoon B and C) move alongside the platoons (in the x direction). The simulations were carried out with the parameters shown in Table 2.1. This yields a radio range of approximately 900m. Hence within each platoon, all the nodes are at most two-hops from each other. When the platoons are close together, the inter-platoon communication is stable without using the supporting nodes 30 and 31. However, when the platoons move away from each other, the direct inter-platoon connections become unstable and the supporting stations become necessary for delivering high traffic. Again for

Type	Source-Destination	Offered Load ($kbps$)
Intra-Platoon	(1,3),(2,9),(4,6),(7,5),(20,29), (14,17),(16,11),(17,18),(19,12), (21,22),(23,27),(23,28)	12
Inter-Platoon	(1,18) (20,11),(20,0) (10,1),(21,10)	2.4 6 12

Table 2.3: Traffic connections for Battlefield scenario

SPTC, we chose the *current age* as link stability metric (this is only a heuristic).

UDP traffic was sent between 17 source-destinations pairs. Table 2.3 shows the base traffic for the scenario. For the traffic analysis, we focus on the connection $(20, 0)$ (from Platoon C to A) because this is a long connection and would be potentially sensitive to path stability. We scale the base traffic (offered load) of all connections (in Table 2.3) by the same factor and obtain the carried load vs. offered load performance for connection $(20, 0)$ shown in Fig. 2.18. Again, we observe that SPTC carries significantly more load than OLSR for this connection. This is because when the platoons are maximally apart, we observe that for connection $(20, 0)$, SPTC-ETX routes significantly more traffic (about 1.5 times more) through the supporting nodes 30 and 31 when compared to OLSR-ETX's routing mechanism. We observe that the carried load for the other connections is also higher. Thus the overall network throughput is improved. For example, when the offered load to the network (all connections) was $2Mbps$, SPTC-ETX was able to carry $923kbps$, while OLSR-ETX is able to carry only $890kpbs$. Figure 2.18 compares the traffic carrying performance for the long connection $(20, 0$ for SPTC-ETX and OLSR-ETX.



(a) Initial topology

(b) Node trajectories

Figure 2.17: Battlefield Scenario

In the TC study, we observed that the average number of topology changes was 12266 and 5360 changes in one hour of simulation time for OLSR-ETX and SPTC-ETX respectively. The corresponding rate of reactive TC messages was $884bps$ and $338bps$ respectively.

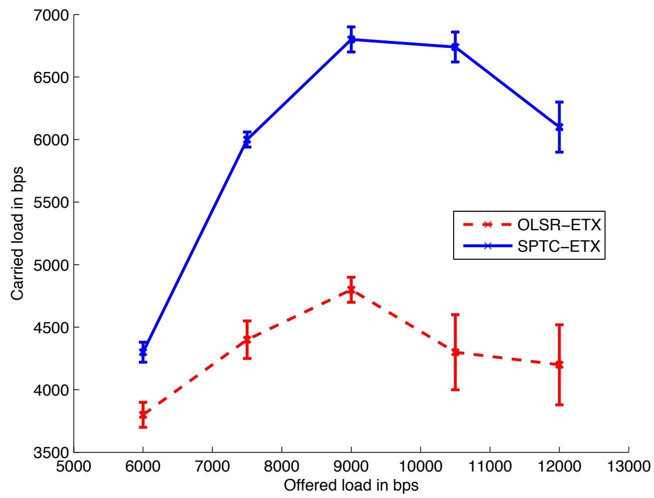


Figure 2.18: Carried load vs. Offered load for the longest connection of Battlefield Scenario shown with 95 % confidence intervals

2.5.8 Summary of Simulation Results

The comparison between the results of SPTC-ETX and OLSR-ETX illustrate the two-fold advantage of stable path topology control:

1. Long-lived paths preserved by SPTC-ETX are cheaper to maintain because they are less likely to change. The simulations show that there are less frequent topology changes for SPTC-ETX.

2. Long-lived paths offers the higher layer traffic more stable sessions. This is reflected in the higher saturation carried load of SPTC-ETX.

2.6 Summary

In this chapter, we introduced the SPTC problem. We developed a framework that allows us to describe global path via local properties of undirected graphs. We use Bellman's optimality principle as a tool to link local properties to global properties of edge-weighted undirected graphs. We showed weak and strong local pruning conditions and their relations to dynamic programming. For the strong local pruning condition we show that strict positivity is a sufficient condition to ensure loop-freedom in pruning.

We also present an approximation algorithm to solve the SPTC problem: SPTC algorithm. Using the ETX as an example metric, we establish using simulations that the SPTC-ETX outperforms the OLSR-ETX.

Chapter 3

Algebraic Path Problems

3.1 Overview

In this chapter, we introduce rule-based routing. In the context of rule-based routing, we illustrate that composition rules that yield path weights and rules of path selections can be effectively described using idempotent semiring algebras. This chapter is intended to serve as a tutorial to algebraic routing, which will be used in Chapter 4.

In particular, we illustrate that Bellman's optimality principle, which was a key tool in Chapter 2, can be expressed algebraically using semiring distribution axiom for idempotent semirings. It captures the notion of "distribution of order."

This chapter is organized as follows. In Section 3.2, we introduce rule-based routing. In Section 3.3, we introduce notation and terminology to describe the rules of routing; we introduce the notions for labeled directed graphs and composition operators for rules of routing. Finally, in Section 3.4, we illustrate the rules of routing can be described in idempotent semiring algebras.

3.2 Rule-Based Routing

In Chapter 2, we introduced the notion of stable path routing. In essence, stable path routing was shortest path routing where the link metrics were stability metrics. We rewrite Eqn. 2.2:

$$
\begin{aligned}
x_{ij}^{G'} &= \min_{p \in P_{ij}^{G'}} w_p \\
&= \min_{p \in P_{ij}^{G'}} \sum_{(u,v) \in p} a_{uv},
\end{aligned}
$$

The above equation computes the optimal paths by using two operations: it sums the link stability metrics along a path to compute the weight of a path, $\sum_{(u,v)\in p} a_{uv}$, and then picks up the path with minimum weight, $\min_{p\in P_{ij}^{G'}} w_p$. These two operators together form, what we call, the sp rule.

A rule of routing is a rule, described by a combination of operations, that selects a path or a set of paths that have the desired properties for routing. There are different types of traffic, and each type has a different rule-based on which the routing paths are selected. The sp rule is only such rule to select shortest paths w.r.t. a metric of interest; in Chapter 2, the metric of interest was path stability. However, there are other metrics of interest in an autonomous network. Different types of traffic, such as voice, which is delay sensitive, or TCP traffic, which is rate and congestion sensitive, have different requirements for path-selection. The requirements are expressed as rules on the metrics of interest. Some of the commonly used rules for routing are shown in Table 3.1.

For the rules of routing in Table 3.1, each link in the network has some attributes or metrics, for instance, delay or trustworthiness. The link attributes of a

Name	Rule	Routing application
Shortest Path (sp)	The weight of a path is the sum of the link weights; the lightest path is chosen for routing	stable path routing, min hop-count routing
Widest Path (bw)	The width of a path is the minimum of the link widths; the widest path is chosen for routing	max-bandwidth path routing, secure routing
Reliable Path (rel)	The reliability of a path is the product of the link reliabilities; the most reliable path, the one with maximum reliability, is chosen for routing	reliable path routing, trusted routing in the context of packet dropping
Pareto bi-objective shortest path (bi-obj sp)	The bi-weight of the path is the sum of the link vector-weights in $\hat{\mathbb{Z}}_+^2$; the paths with the minimal weights (Min^{com}) in the Pareto order are chosen for routing	delay-trusted routing

Table 3.1: Common rules used in network routing

path are composed by some rule and the path or the path-set which has the most desirable attributes (described by an order relation or rule) in chosen. In other words, the rules of routing in Table 3.1 have two key components:

1. Arc Composition: The link attributes or metrics are composed along the path.

2. Order: There is a notion of order for composed metrics that lets the router to choose desirable paths or path-sets.

We will formalize this abstraction in the forthcoming sections. We will also show that this abstraction need not to be restricted to single rule over a single metric, but can be generalized to multiple rules over multiple metrics, e.g., bi-obj sp.

3.3 Graphs and Metrics for Rule-Based Routing

In Chapter 2, we dealt with only undirected graphs because the metric of interest, link stability a_{uv}, is a symmetric metric, i.e., $a_{uv} = a_{vu},\ (u, v) \in E$. However, a general metric need not be symmetric. A good example of such a metric is trustworthiness in an autonomous system [57]. The trust that $u \in V$ has on $v \in V$ need not be identical to the trust that v has on u. Consequently, we need a directed labeled graph abstraction, to handle generic metrics. The labels of interest for rule-based routing are vector metrics, which live in ordered sets. For a brief introduction to order theory, see Appendix B.1.

Let $G(V, A)$ denote a directed graph, where V is the vertex set and $A \subseteq V \times V$ is the directed arc set. Associated with each arc $(u, v) \in A$ is a label of m metrics, denoted by the vector $\underline{a}_{uv}$. Each component $\underline{a}_{uv}(l) \in S_l,\ 1 \le l \le m$, where S_l is a totally ordered set. We call S_l, the *constituent metric* set and $S = \times_l S_l$, the *product metric* set. There is no natural on S; we will introduce non-dominance operators that induce an order on S.

Let $G'(V', A')$ be a subgraph of G. Similar to the undirected graphs case, $P_{ij}^{G'}$ denotes the set of paths from $i \in V'$ to $j \in V'$ in G'. Additionally in this chapter, we introduce deal with subsets of $P_{ij}^{G'}$. Let $2^{P_{ij}^{G'}}$ denote the power set, i.e., the set of all subsets, of $P_{ij}^{G'}$. Any subset of $2^{P_{ij}^{G'}}$ is denoted in the calligraphic alphabet, e.g., $\mathcal{P} \subset 2^{P_{ij}^{G'}}$. For a summary of notation used in graph theory, see Appendix A.1.

We next define the composition operators that are necessary to specify the rules of routing. As mentioned in Section 3.2, we need to specify rules to compose the attributes or metrics along a path, and rules to select desirable paths from these composed attributes or metrics.

3.3.1 Composing Arc Metrics

For every path $p = (i = u_1, u_2, u_3, \ldots, u_{n-1}, u_n = j) \in P_{ij}^{G'}$, we obtain the path metric by composing the arc metrics along the path. We obtain a m-dimensional path metric w_p by the componentwise composition:

$$w_p(l) = \underline{a}_{u_1 u_2}(l) \otimes_l \underline{a}_{u_2 u_3}(l) \otimes_l \cdots \otimes_l \underline{a}_{u_{n-1} u_n}(l), \quad 1 \leq l \leq m, \tag{3.1}$$

where $\otimes_l$ is the rule for arc composition of the l^{th} component. For instance, for the metric set could be $S_l = \{\perp, \top\}$ the corresponding arc composition rule $\otimes_l$ could be Boolean disjunction $\vee$ or conjunction $\wedge$. Note that for $S_l = \{\perp, \top\}$, a possible order is $\perp < \top$. For the sp example, the metric set is $\hat{\mathbb{R}}_+$, and $\otimes_l$ is standard addition and the order is the natural order of the reals. Thus depending on the metric, different rules of composition can be defined. For all the constituent metrics we assume that the composed constituent metric lives in a totally ordered set. We also assume that constituent composition, $\otimes_l,\ 1 \le l \le m$, yields a weight component, $w_p(l),\ 1 \le l \le m$, that lives in the totally ordered constituent metric set, S_l, i.e., $w_p(l) \in S_l$. The componentwise compositions of Equation 3.1 can be compactly represented in vector notation:

$$w_p = \underline{a}_{u_1 u_2} \otimes \underline{a}_{u_2 u_3} \otimes \cdots \otimes \underline{a}_{u_{n-1} u_n}. \tag{3.2}$$

$w_p \in S$ is the vector-valued weight of path p.

Let ①$_l$ and ⓪$_l$ denote the bottom and top elements of S_l respectively.

$$a \in S_l,\ ①_l \le a \le ⓪_l.$$

We will assume that totally ordered sets have bottom and top elements. Note that any totally ordered set can be lifted to have bottom or top element [16]. The vector weight of an empty path is ①=[①$_1$, ①$_2$, , . . . , ①$_l$]^T.

3.3.2 Rules for Path Selection

Although, the constituent metric sets $S_l,\ 1 \le l \le m$ are totally ordered, there is no natural order on S, and consequently, we cannot define a path/path-selection rule. To resolve this, we resort to methods from multi-criteria optimization. See B.1.1 for a summary of definitions and notations from multi-criteria optimization.

Given the vector-weights for all the paths between a pair of vertices, $w_p,\ p \in P_{ij}^{G'}$, the path weights can be compared using a non-dominance operator Min. The

Min operator could be one of the Min^{com}, Min^{lex} or Min^{MO}. It could be any rule that satisfies the axioms of a partially ordered set (Appendix B.1). The minimal elements of the set $\{w_p : \ p \in P_{ij}^{G'}\}$ are called non-dominated weights and those paths that achieve these minimal elements are called efficient paths.

The composition of path weights, to find the non-dominated weights corresponding to the efficient paths, i.e., *path composition* or *path selection*, is expressed by another operator $\oplus(= \mathrm{Min})$

$$x_{ij}^{G'} = \oplus_{p \in P_{ij}^{G'}} \{w_p\},$$

$$x_{ij}^{G'} = \mathrm{Min}_{p \in P_{ij}^{G'}} \{w_p\}. \tag{3.3}$$

Note that the non-dominated solution $x_{ij}^{G'}$ is a subset of S, and the non-dominated solution is achieved by a subset of paths of $P_{ij}^{G'}$, called efficient path-set. There could be more than one such efficient path sets. Let the set of all such efficient path-sets be

$$\mathcal{P}_{ij}^{G'*} = \{P \in 2^{P_{ij}^{G'}} : \ w_P = \mathrm{Min}(w_{P_{ij}^{G'}})\}, \tag{3.4}$$

where $w_P, \ P \in 2^{P_{ij}^{G'}}$, is the set $\{w_p, \ p \in P\}$. The system of equations given by Equations (3.2) and (3.3), i.e., the arc composition and path selection, is called the *algebraic path problem*.

For any efficient path-set has an absorbing property, i.e., $P \in \mathcal{P}_{ij}^{G'*}$ and $P' \in 2^{P_{ij}^{G'}}$,

$$w_P \oplus w_{P'} = w_P,$$

where $w_P \oplus w_{P'} = \mathrm{Min}(w_P \cup w_{P'})$. Conversely, for $P, P' \in 2^{P_{ij}^{G'}}$,

$$w_P \oplus w_{P'} \neq w_P,$$

then w_P does not dominate $w_{P'}$, and P is not an efficient path-set.

3.3.3 Distribution of Order

Consider the sp problem of Chapter 2. The sp rule follows the Bellman's optimality principle (Lemma 2.3.1). The Bellman's principle can be expressed in an algebraic framework: consider an example network shown in Fig. 3.1. Let us suppose $a, b, c \in \hat{\mathbb{Z}}_+$, and $b < c$. The addition composition of the sp rule implies $a + b < a + c$. Algebraically, the minimum operator distributes over the + of the sp rule:

$$\min\{a + b,\ a + c\} = a + \min\{b,\ c\}.$$

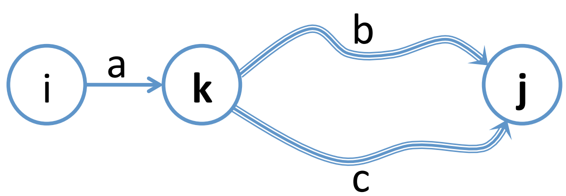


Figure 3.1: Spoon network to illustrate the distribution of order. The network shows two paths from k to j, with weights b and c. There is an arc from i to k with link weight a.

Interestingly, the above notion of "distribution of order extends" to vector metrics too! Consider the bi-obj sp rule (Table 3.1). Let $a, b, c \in \hat{\mathbb{Z}}^2_+$. For the bi-obj sp rule uses Min^{com} non-dominance operator. In this case, the distribution corresponding to the network in Fig. 3.1 is given by

$$\text{Min}^{com}\{a + b,\ a + c\} = \text{Min}^{com}\{\{a\} + \{\text{Min}^{com}\{b,\ c\}\},$$

where the sum of the two sets is the Minkowski sum of two sets, i.e., for sets A, B,

$$A + B = \{a + b : \ a \in A, \ b \in B\}.$$

The above two examples suggest that the rules of routing of distribute the non-dominance operator over the arc composition rule $\otimes$. Before we express this distribution, we define the arc composition composition and path selection rules for subsets of S. We need this abstraction because the non-dominated solutions are subsets of S. For the generic rule-based routing, with arc composition, $\otimes$ and path selection $\oplus(=\text{ Min})$ (non-dominance operator), the operator definitions can be extended to sets as follows. The arc composition of two sets is the Minkowski product, i.e., for sets $A, B \in S$,

$$A \otimes B = \text{Min}\{a \otimes b : \ a \in A, \ b \in B\},$$

and the path composition of two sets is the Minkowski sum, i.e., for sets $A, B \in S$,

$$A \oplus B = \text{Min} A \cup B.$$

We define the algebraic distribution for a general graph G' as follows. Consider the set of paths from i to k, $P_{ik}^{G'}$, and the set of paths from k and j, $P_{kj}^{G'}$. Then,

$$\oplus w_{P_{ik}^{G'}.P_{kj}^{G'}} = \oplus\{\oplus w_{P_{ik}^{G'}} \otimes \oplus w_{P_{kj}^{G'}}\}. \tag{3.5}$$

The algebraic distribution rule states that the computation of the non-dominated solution for a path-set concatenation can be decomposed in terms of the non-dominated solutions of the constituent path-sets.

Name	S	$\oplus$	$\otimes$	⓪	①	routing application
sp	$\hat{\mathbb{Z}}_+$	min	$+$	∞	0	shortest path
sp_q	$\hat{\mathbb{Z}}_q$	min	$+$	∞	0	shortest path (bounded distance)
bw	$\hat{\mathbb{Z}}_+$	max	min	0	∞	widest path (greatest capacity)
bw_q	$\hat{\mathbb{Z}}_q$	max	min	0	∞	widest path (greatest bounded capacity)
rel	$[0, 1]$	max	$\times$	0	1	most reliable path
cup.cap(W)	2^W	$\cup$	$\cap$	$\emptyset$	W	shared link attributes
cap.cup(W)	2^W	$\cap$	$\cup$	W	$\emptyset$	share path attributes

Table 3.2: Semirings used in network routing, where $\hat{\mathbb{Z}}_+ = \mathbb{Z}_+ \cup \{\infty\}$ and $\hat{\mathbb{Z}}_q = \{0, 1, 2, \ldots, q-1, \infty\}$

.

The Equations (3.2), (3.3) and (3.5) together define the necessary properties of any rule of routing. In the next section, we will show that these properties are satisfied by an idempotent semiring constructed from constituent selective semirings.

3.4 Idempotent Semiring Algebraic Path Problem for Routing Rules

It is known that a number of rules for routing can be expressed using semiring algebras. For example, some of the common rules are shown to be instances of specific semiring algebras in shown in Table 3.2 [23]. In particular, the different rules of routing can be expressed by a Semiring Algebraic Path Problem (SAPP) on labeled directed graph. For a detailed exposition on the algebraic path problem, see

Appendix B.4. It is also well-known in the controls community that many multi-criteria path problems can be viewed as algebraic path problems in idempotent semirings [34, 35, 50].

3.4.1 Semirings

A semiring is an algebraic structure $(S, \oplus, \otimes)$ that satisfies the following axioms:

(A1) (S,$\oplus$) is a commutative monoid with a neutral element $⓪$:

$$\begin{aligned} a \oplus b &= b \oplus a \\ a \oplus (b \oplus c) &= (a \oplus b) \oplus c \\ a \oplus ⓪ &= a \end{aligned}$$

(A2) (S,$\otimes$) is a monoid with a neutral element $①$*, and an absorbing element* $⓪$:

$$\begin{aligned} a \otimes (b \otimes c) &= (a \otimes b) \otimes c \\ a \otimes ① = ① \otimes a &= a \\ a \otimes ⓪ = ⓪ \otimes a &= ⓪ \end{aligned}$$

(A3) $\otimes$ *distributes over* $\oplus$:

$$\begin{aligned} a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c) \\ (a \oplus b) \otimes c &= (a \otimes c) \oplus (b \otimes c) \end{aligned}$$

For the rule-based routing framework, the arc composition rule corresponds to $\otimes$ and the path selection rule corresponds to $\oplus$ of the semiring. In particular, for the path selection, $\oplus$ has a special idempotency structure, which is discussed in the following subsection.

3.4.2 Idempotent and Selective Semirings - Path Selection

For an idempotent semiring, the $\oplus$ is idempotent:

$$a \oplus a = a, \; a \in S.$$

It is shown in [22] that this idempotent property induces a canonical (partial) order that is expressed as

$$a \leq b \iff a = a \oplus b.$$

$\oplus$ has an absorbing property.

All the semirings in Table 3.2 are idempotent semirings. A sub-class of idempotent semirings is called *selective semirings* where the $\oplus$ operator is selective:

$$a \oplus b = a \text{ or } b, \; a, b \in S.$$

For example, sp, sp_q, bw, bw_q and rel of Table 3.2 are selective semirings. However, cap.cup(W), cup.cap(W) and the bi-objective sp are idempotent, but not selective. For selective semirings, S becomes totally ordered.

Remember, the constituent metrics, $a_{uv}(l), \; 1 \leq l \leq m$, and the composed constituent path metric, $w_p(l), \; 1 \leq l \leq m$, live in a totally ordered set, $S_l, \; 1 \leq l \leq m$. Consequently, the paths can be directly compared using a minimum operator, denoted by $\oplus_l (= \min)$. Consequently, the comparison for each component can be expressed using a selective semiring, $(S_l, \oplus_l, \otimes_l)$. However, the vector path metric,

w_p, lives in a partially ordered product constituent set, S, and the operators for path selection can be expressed in an idempotent semiring that is formed from the constituent selective semirings. This idempotent semiring is constructed as follows.

Path selection rules, as described in Section 3.2, select paths or path-sets based on metrics. In these cases, some path-sets are dominated by others and the dominated path-sets discarded in the path-selection stage. This is captured by the idempotent property: for $P, Q \in 2^{P_{ij}^{G'}}$, if $w_P \leq w_Q$, then $w_p = w_p \oplus w_q$ (absorbing property).

If the path weights are totally ordered, then idempotency reduces to selectiveness. In most cases, the single metric routing problems can be captured by selective semirings, whereas multi-metric routing problems are captured by idempotent semirings. This is because idempotency naturally lends to a partial order.

We define a semiring on the set of all subsets of S, denoted by $(2^S, \oplus, \otimes)$. Here, the carrier set of the semiring, 2^S, is the set of all subsets of S. The composition operator between two sets $A, B \in 2^S$, is the Minkowski arc composition, i.e.,

$$A \otimes B = \{a \otimes b : \ a \in A, \ b \in B\}.$$

The generalized sum that yields the path-selection operator is defined as

$$A \oplus B = \mathrm{Min} A \cup B.$$

Note that this generalized sum is an idempotent operator. Since this semiring is constructed from the constituent selective semirings (for each constituent metric), it follows that distribution (of order) holds:

$$(A \otimes B) \oplus (A \otimes C) = A \otimes (B \oplus C).$$

Note that this idempotent semiring has satisfies the three necessary properties of rule-based routing:

1. Composition : Equation (3.2).

2. Path-Selection: Equation (3.3).

3. Distribution of order: Equation (3.5).

3.5 Summary

We showed that rules of routing can be expressed as algebraic path problems in idempotent semirings. We showed that the semiring distribution axiom guarantees distribution of order, which is a central theme in most routing rules.

Chapter 4

Topology Control for Rule-Based Routing

4.1 Overview

In this chapter, we develop a generic local pruning policy for topology control (link-state selection) for rule-based routing. The primary contribution in this chapter is in extending the strong pruning condition of SPTC, from Chapter 2, to the rule-based routing framework. The local policies that we develop illustrate the need for the distribution of order property, which is a tool to translate local properties to global properties of directed labeled graphs. In doing so, we show a manifestation of the Bellman's optimality principle that extends to path-sets and minimal elements of partially ordered sets.

This chapter is organized as follows. In Section 4.2, we develop the notation for local pruning in the context of rule-based routing. In Section 4.3, we present a sufficient pruning condition that ensures that globally efficient paths (in the context of rules) are preserved in the global view. Finally, in Section 4.4, we present two example applications of this generic pruning: hop-count lexicographic extension to SPTC and Pareto optimal trusted routing.

4.2 Notations and Definitions

In this section, we extend the notations from Chapter 3 to encompass directed graphs for the topology control problem. Similar to the notations introduced in Chapter 2 for the SPTC problem, we will introduce notions of neighborhoods, local views and global views on directed labeled graphs.

4.2.1 Graphs, Metrics and Views

We will, first, summarize notation on from Chapter 3. In Chapter 3, we introduced the notion of directed labeled graphs, where the arc labels are vector metrics: $G(V, A)$ is a directed graph, where V is the vertex set and $A \subseteq V \times V$ is the directed arc set. Associated with each arc $(u, v) \in A$ is a label of m metrics, denoted by the vector $\underline{a}_{uv}$. Each component $\underline{a}_{uv}(l) \in S_l,\ 1 \leq l \leq m$, where S_l, called the constituent metric set, is a totally ordered set. $S = \times_{l=1}^{m} S_l$, called the product metric set, inherits different partial orders depending on the different vector orders.

The NDC at h discovers all its symmetric neighbors, those that can hear each other. Consequently, h discovers an undirected communication adjacency graph of the local neighborhood. However, in this case, the arc metrics a_{uv}, being asymmetric, induce a direction to each edge. Thus an edge (u, v) of the communication adjacency splits into two arcs $(u, v) \in A$ and $(v, u) \in A$ labeled with a_{uv} and a_{vu} respectively. The NDC discovers this local neighborhood. This is a major difference between the SPTC problem and the generic topology control problem.

It is important to note that the underlying communication graph is an undirected graph, $G^c(V,E)$, because the NDC detects only symmetric neighbors. The labels being asymmetric induce a direction to form a directed graph $G(V,A)$. Consequently, the adjacency of G is identical to that of G^c, but for the direction, i.e., if $(u,v) \in E$, then $(u,v) \in A$ and $(v,u) \in A$.

The hop count hc of a path is the number of arcs/edges in the path. Then the minimal hop count distance between a pair of vertices (i,j) in G (or G^c) is defined as $d_{hc}(i,j) = \min\limits_{p \in P_{ij}^G} hc(p)$. We define the *k-hop neighborhood* for a host $h \in V$ by

$$N_h^k = \{j \in V : d_{hc}(h,j) \leq k\}.$$

k is called the *size of the neighborhood*. The boundary set for the neighborhood N_h^k is given by

$$\partial N_h^k = N_h^k \backslash N_h^{k-1},$$

where $N_h^0 = \{h\}$, and $N_h^k = \emptyset, k < 0$.

G' is a labeled sub-graph of G. For any vertex $i \in V'$, the set of in-arcs incident to i in G' is denoted by $\Omega_i^{G'}$. We will introduce special sub-graphs that arise in compressed topology dissemination link-state routing protocols.

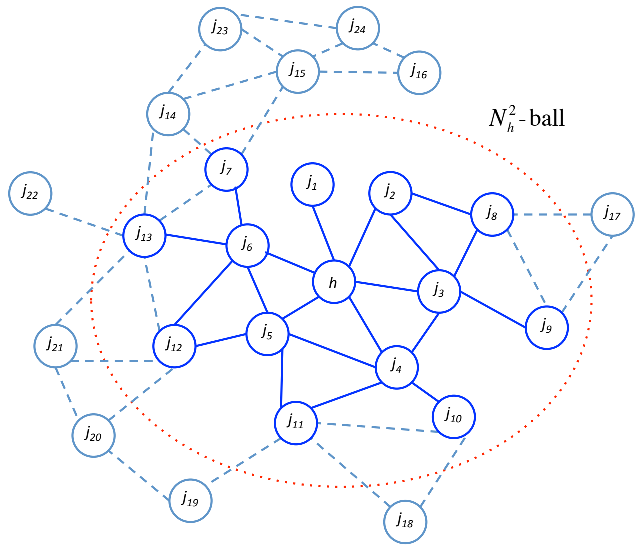
The neighbor discovery protocol exposes the k-hop neighborhood for every host $h \in V$. The host discovers a labeled directed subgraph G_h^{local} that contains only the vertices in N_h^k and all the arcs between them, expect those between any two vertices of the boundary set, i.e, the vertex set is N_h^k, the arc set is $\{(u,v) \in A : \ u,v \in N_h^k \text{ and } \{u,v\} \not\subseteq \partial N_h^k\}$, and the labels a_{uv} for every arc in this arc set. We call this labeled subgraph G_h^{local}, the *local view* for the host h. An example local view for h is shown in Fig. 4.1. Fig. 4.1a illustrates the undirected communication adjacency between the symmetric neighbors discovered in the neighbor discovery phase. This is identical to the local view in the context of SPTC. Fig. 4.1b is the directed graph induced by the asymmetric arc labels. As the figure shows there is a direct correspondence with the undirected graph, where each edge is replaced with two arcs in opposing directions.

Unlike the SPTC problem, for generic rule-based routing, we are interested in rules that work on partially ordered sets. Consequently, the desirable solutions are described in terms of non-dominated metrics and the desirable path-sets correspond to efficient path-sets (Chapter 3). Consequently, we need to define notations for path-set relative to local-views. For any path $p = (j = v_1, v_2, \ldots, v_n = h) \in P^G_{jh}$, the *gateway* of p in G^{local}_h, denoted by γ^h_p, is the last vertex of p that is in the boundary set ∂N^k_h. i.e., $\gamma^h_p = v_t$ if and only if $v_t \in \partial N^k_h$ and $v_s \notin \partial N^k_h, t < s < n$. If the path p never intersects ∂N^k_h, i.e., $v_s \notin \partial N^k_h,\ 1 \leq s \leq n$, then γ^h_p is not defined. For a $p = (j = v_1, v_2, \ldots, v_n = h) \in P^G_{jh}$, with γ^h_p defined, the h-local sub-path, $p^{h-local} = (\gamma^h_p = v_t, v_{t+1}, \cdots, v_n = h)$, such that $v_s \neq \gamma^h_p,\ s > t$, is the sub-path of p that is completely contained in G^{local}_h. The remnant path of p is denoted by $p^{h-not-local} = (v_1, v_2, \cdots, v_t = \gamma^h_p)$.

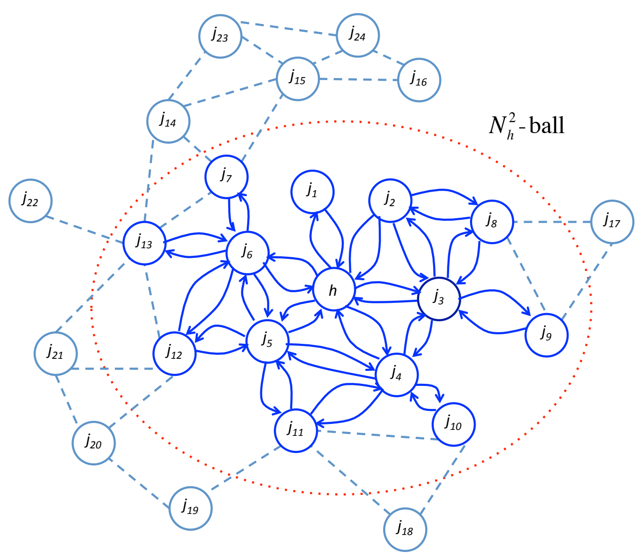
For a path-set $P \in 2^{P^G_{jh}}$, we denote the set of gateway vertices by $\Gamma^h_P = \{\gamma^h_p : p \in P\}$. Fig. 4.2 illustrates the notion of a gateway to a path-set.

Given a path-set $P = \{p_1, p_2, \cdots, p_m\} \in 2^{P^{G'}_{jh}}$, the h-local sub-path-set is a list of h-local sub-paths referenced by a gateway vertex. For any gateway vertex $\gamma \in \Gamma^h_P$,

$$P^{h-local}(\gamma) = \{p^{h-local}_r :\ \gamma^h_{p_r} \text{ is defined and } {p^{h-local}_r}_{(1)} = \gamma\},$$



(a) Undirected Communication Adjacency in Local View



(b) Directed Local View

Figure 4.1: Local View of h: The edges and arcs indicated in solid lines are in G_h^{local} while the those indicated in dashed lines are not.

which is the set of h-local paths of P that have γ as the starting vertex. This is also illustrated in Fig. 4.2. Similarly, the h-not-local sub-path-set is another list, for $\gamma \in \Gamma_P^h$,

$$P^{h-not-local}(\gamma) = \{p_r^{h-not-local} : \gamma_{p_r}^h \text{ is defined and } {p_r^{h-local}}_{(1)} = \gamma\}.$$

Note one means to reconstruct a path set $P \in 2^{P_{jh}^{G'}}$ from the h-local and h-not-local sub-path-sets is

$$P = \cup_{\gamma \in \Gamma_P^h} P^{h-not-local}(\gamma).P^{h-local}(\gamma)$$

The concatenation of paths and path-sets is defined in Appendix A.1.

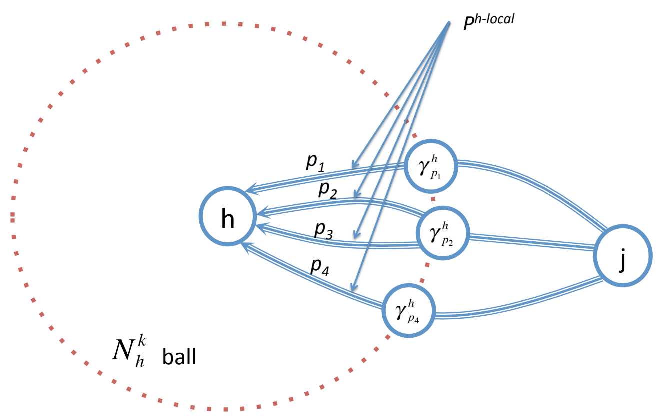


Figure 4.2: Gateway for path-set $P = \{p_1, p_2, p_3, p_4\} \in 2^{P_{jh}^{G}}$ in local view G_h^{local}. The N_h^k ball is indicated as a dotted ellipse. Since paths p_2 and p_3 intersect at the boundary ∂N_h^k, their gateway vertices are identical, i.e, $\gamma_{p_2}^h = \gamma_{p_3}$. The gateway-set for P is $\Gamma_P^h = \{\gamma_{p_1}^h, \gamma_{p_2}^h, \gamma_{p_4}^h\}$. The h-local sub-path-set list is $P^{h-local}(\gamma_{p_1}^h) = \{p_1^{h-local}\}$, $P^{h-local}(\gamma_{p_2}^h) = \{p_2^{h-local}, p_3^{h-local}\}$, and $P^{h-local}(\gamma_{p_3}^h) = \{p_3^{h-local}\}$.

In local pruning algorithms, the host $h \in V$, which has discovered its local view G_h^{local}, chooses a subset of its in-arcs, which we call the *pruned in-arc set* of h. The set of in-arcs incident to h in G_h^{local} is denoted by Ω_h^G. The pruned in-arc set is denoted by $\Omega_h^{pruned} \in 2^{\Omega_h^G}$, where $2^{\Omega_h^G}$ is the power set of Ω_h^G, which is the set of all subsets of Ω_h^G.

Every host h broadcasts its pruned incident in-arc set, Ω_h^{pruned}, and the corresponding arc labels. The corresponding broadcast arc set is given by $E^{broadcast} = \cup_{h \in V} \Omega_h^{pruned}$, and this induces a labeled subgraph $G^{broadcast}$, which we call the *broadcast view*. The broadcast view is common to all the hosts $h \in V$. Fig. 4.3 illustrates the notion of a broadcast view.

Once the broadcast view is available at the host, $h \in V$, it has the final link-state information to compute the paths for routing. This link-state information is the combination of the local view and broadcast view, which is abstracted as the *global view*: At every host station $h \in V$, the *global view* G_h^{global} is the graph union $G_h^{local} \cup G^{broadcast}$. Note that the global view, like the local view, is specific to a host h; every host h sees a different global view. Fig. 4.4 shows the global view corresponding the local and broadcast views of Figure 4.1 and 4.3 respectively.

For rule-based routing, we assume that rules are described in a semiring $(S, \oplus, \otimes)$, where $S = \times_{l=1}^{m} S_l$, and $\oplus$ is the path-selection operator (non-dominance operator) and $\otimes$ is the vector arc composition operator. The vector weight of a path p is

$$w_p = \otimes_{(u,v) \in p} \underline{a}_{uv},$$

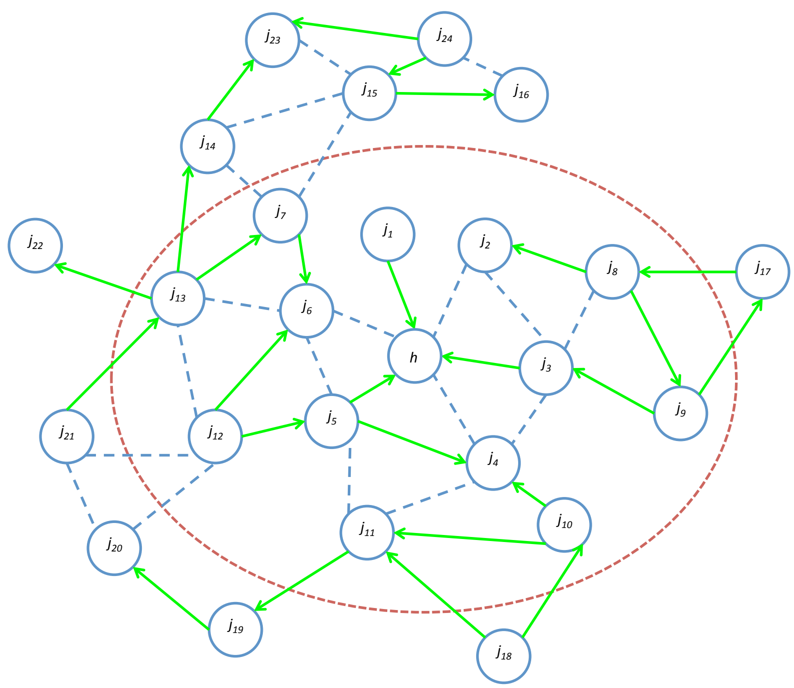


Figure 4.3: Broadcast view $G^{broadcast}$: The arcs in solid lines are broadcast and they constitute the broadcast view, while those in dashed lines are pruned away, i.e., not broadcast.

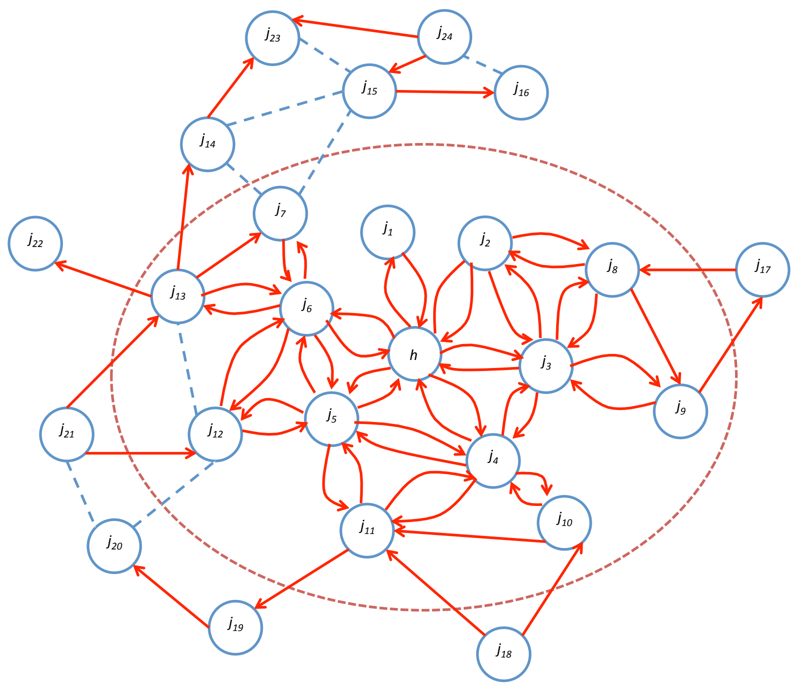


Figure 4.4: Global view of h corresponding the local view of Fig. 4.1 and the broadcast view of Fig. 4.3. The global view is the graph union of the local and global views; it also includes the arc labels. Again, those arcs indicated in solid lines are a part of G_h^{global}, while that are indicated in dashed lines are not.

and the rule for path selection is given by

$$x_{ij}^{G'} = \oplus_{p \in P_{ij}^{G'}} \{w_p\}.$$

Again, we define path-sets, solutions and solution path-sets relative to G_h^{local}. The set of paths from h to any vertex $j \in N_h^k$, $P_{hj}^{G_h^{local}}$, is denoted by $P_{hj}^{h-local}$, which we call the *h-local path-set*. We denote by $x_{hj}^{h-local}$, $j \in N_h^k$, the non-dominated solution to any rule-based routing restricted to G_h^{local}, i.e., $x_{hj}^{h-local} = x_{hj}^{G_h^{local}}$. The corresponding set of efficient path-sets is denoted by $\mathcal{P}_{hj}^{h-local*} (= \mathcal{P}_{hj}^{G_h^{local}*})$, which we call the set of h-locally efficient path-sets.

4.3 Local Pruning for Rule-Based Routing

Similar to the Stable Path Topology Control (SPTC) problem of Chapter 2, the pruning mechanism must ensure that at-least one efficient path-set is preserved after pruning in G_h^{global}. Let $\pi_h^R(G)$ denote a global property, where R is the rule of interest. The property $\pi_h^R(G)$ is said to hold for any $G' \subseteq G$ if at-least one efficient path-set is preserved in G'. Let $\Pi_h^R(G)$ denote the set of all such subgraphs. Then the global constraint is

$$G_h^{global} \in \Pi_h^R(G). \tag{4.1}$$

Next we define a general version of the strong pruning condition of Chapter 2. Let $\pi_h^{\Omega-R}(G_h^{local})$ at $h \in V$ be a local property, where R is the rule of interest. Given the local view G_h^{local}, the property $\pi_h^{\Omega-R}$ is said to hold for $\Omega \in 2^{\Omega_h^G}$, if for all $j \in \partial N_h^k$ there exists a h-locally efficient path-set $P \in \mathcal{P}_{jh}^{h-local*}$ such that $\forall p \in P$, $(p_{(-2)}, h) \in \Omega$. Let $\Pi_h^{\Omega-R}(G_h^{local})$ denote all such Ω's. The strong pruning condition is given by

$$\Omega_h^{pruned} \in \Pi_h^{\Omega-R}(G_h^{local}). \tag{4.2}$$

We will show that the local condition (Equation (4.2)) under some assumptions of the composition rules becomes sufficient to guarantee the global condition (Equation (4.1)). To prove this sufficiency, we need some results similar to Bellman's optimality principle that allows us to translate local properties to global properties. As alluded to, in Chapter 3, these results depend on the principle of "distribution of order." In particular, we want to establish relation between a globally efficient path-set and its sub-path-sets in the local view. The following Lemma captures this notion.

Lemma 4.3.1 *Given an efficient path set $P \in \mathcal{P}_{jh}^{G'^*}$, any of its h-local sub-path-set is h-locally efficient, i.e., for $\gamma \in \Gamma_P^h$, $P^{h-local}(\gamma) \in P_{\gamma h}^{h-local*}$.*

Proof Similar to the proof of Lemma 2.3.1, we assume that $P^{h-local}(\gamma) \notin P_{\gamma h}^{h-local*}$ and derive a contradiction. Let $Q \in P_{\gamma h}^{h-local*}$. Then by definition

$$w_{P^{h-local}(\gamma)} \oplus w_Q \neq w_{P^{h-local}(\gamma)}$$

Consider the concatenation of the path-set $P^{h-not-local}(\gamma)$ and Q, $P^{h-not-local}(\gamma).Q$, then by semiring distribution axiom, *A3*,

$$w_{P^{h-not-local}(\gamma).Q} \oplus w_{P^{h-not-local}(\gamma).P^{h-local}(\gamma)} \neq w_{P^{h-not-local}(\gamma).P^{h-local}(\gamma)}$$

Now consider the path P' constructed from P by replacing the sub-path-set $P^{h-local}(\gamma)$

with Q. Note $P' \in 2^{P^{G'}_{jh}}$. Then,

$$\begin{aligned} w_P \oplus w_{P'} &= w_{\{\cup_{\gamma' \in \Gamma^h_P} P^{h-not-local}(\gamma').P^{h-local}(\gamma')\}} \oplus \\ &\quad w_{\{P^{h-not-local}(\gamma).Q, \cup_{\gamma' \neq \gamma} P^{h-not-local}(\gamma').P^{h-local}(\gamma')\}} \\ &\neq w_{\{\cup_{\gamma' \in \Gamma^h_P} P^{h-not-local}(\gamma').P^{h-local}(\gamma')\}} \\ &= w_P, \end{aligned}$$

which is a contradiction.

Note that above lemma establishes that for any efficient path-set, its h-local sub-path-set is h-locally efficient. This is a more general version of the Lemma 2.3.2.

4.3.1 Strict Inflatory Condition, Sufficiency and Loop-Freedom

We present a sufficient condition for local pruning to ensure loop-freedom, which is called strict inflatory condition [23]:

$$a, b \in S,\ a \otimes b > a, b.$$

This condition is also referred to as strict isotonicity by Sobrinho [51]. This condition is a direct analog to the strict positivity condition of the SPTC problem. We will show that under the strict inflatory assumption, the local condition becomes sufficient.

Theorem 4.3.2 *Under the strict inflatory assumption, if $h \in V$, $\Omega^{pruned}_h \in \Pi^{\Omega-R}_h(G^{local}_h)$, then $G^{global}_h \in \Pi^R_h(G)$.*

Proof We need to show that G_h^{global} has at-least one efficient path-set to any vertex $j \in V$. We will prove by construction that one optimal path is preserved under pruning. There are two cases to consider.

Case 1: $P \in \mathcal{P}_{hj}^{G}*$ is contained in G_h^{local}. Then P is contained in G_h^{global} because $G_h^{global} = G_h^{local} \cup G^{broadcast}$.

Case 2: No $P \in \mathcal{P}_{hj}^{G}*$ is completely contained in G_h^{local}. We will prove a path-set $P \in P_{hj}^{G}*$ contained in G_h^{global} by construction from the terminal vertex to the starting vertex. Consider the pruning at $j = u_1$. Let $P \in \mathcal{P}_{hu_1}^{G}{}^{*}$ be any globally efficient path-set from h to u_1. Consider the u_1-local sub-path-set of P. Since the strong pruning condition, $\Omega_{u_1}^{pruned} \in \Pi_{u_1}^{\Omega-R}(G_h^{local})$ holds at u_1, there exists a path-set $Q \in \mathcal{P}_{\gamma u_1}^{h-local*}$, $\gamma \in \Gamma_P^h$, such that $(u_1, q_{(-2)}) \in \Omega_{u_1}^{pruned}$ for $q \in Q$. Choose any arbitrary $q_{u_1} \in Q$. Let $q_{(-2)} = u_2$, which we call the pruned next-hop of u_1 from h.

By Lemma 4.3.1, the concatenation $Q.P^{h-not-local} \in \mathcal{P}_{hu_1}^{G}{}^{*}$, i.e. is globally efficient. So (u_1, u_2) is an edge of a path in an efficient path-set. Now consider any efficient solution from h to u_2, $P' \in \mathcal{P}_{hu_2}^{G}{}^{*}$. Since the strong pruning condition holds at u_2, all the above arguements holds and we obtain any constituent path q_{u_2} of any efficient path set in $\mathcal{P}_{hu_2}^{G}{}^{*}$. Then $w_{q_{u_1}} = a_{u_1 u_2} \otimes w_{q_{u_2}}$. By the strict inflatory assumption, $w_{q_{u_1}} > w_{q_{u_2}}$.

Repeating the above construction, we get a sequence of vertices $u_1, u_2, u_3, \cdots, u_r$, where u_{s+1} is any pruned next-hop of u_s from h. The sequence of vertices construct a path that is a path of some efficient path-set from h to j. Then $w_{q_{u_1}} > w_{q_{u_2}} > \cdots > w_{q_{u_r}}$. This implies that there can be no loops in pruning. Since there are no loops (each u_r is unique) and V is finite, there exists a u_l in the construction such that $u_l \in \partial N_h^k$. Then there exists a h-locally efficient path-set from h to u_l in G_h^{local}, which contains a path $(h = u_n, u_{n-1}, \cdots, u_l)$. We have shown that the pruning preserves one path of the set of paths that constitute an efficient path-set from h to j. Since the choice of q_{u_s}, $1 \leq s \leq l$ was arbitary, we construct the entire efficient path-set.

4.4 Application of Generalized Pruning

4.4.1 Hop-Count Lexicographic Extension to Loop-Free SPTC

In the SPTC results of Chapter 2, we assumed positivity, i.e., the link weights $a_{uv} > 0$, to ensure loop-freedom. However, there is another means to ensure loop-freedom, which is captured by the generalized pruning. Suppose the weight of the path is defined as vector weight

$$w_p = \begin{bmatrix} \text{link-stability}(p) \\ hc(p) \end{bmatrix},$$

and the vectors are compared in lexicographic order, then it is sufficient to ensures loop-freedom with only $a_{uv} \geq 0$. This is because the lexicographic hop-count extension can be represented in a semiring

$$(\hat{\mathbb{R}}_+ \times \hat{\mathbb{Z}}_+, \text{Min}^{lex}, \begin{bmatrix} + \\ + \end{bmatrix})$$

In this case the vector arc weights, by default, satisfy the inflatory assumption in the lexicographic order because the hop-count is never zero!

4.4.2 Topology Control for Pareto Optimal Trusting Routing

The Pareto optimal trusted routing is an routing strategy where routing paths are selected based on both the delay and the trust of the path. The weight of a path p is

$$w_p = \begin{bmatrix} \text{delay}(p) \\ \text{trustworthiness}(p) \end{bmatrix},$$

where the delay is given by the sum of the delays along the path and the trustworthiness is the product trustworthiness of a path. In the context, trustworthiness can be throught of the the reliability of the path. This notion of trustworth is called packet-dropping trust.

This vector metric can be compared in using a modified componentwise order $\leq_{delay-trust}$:

$$\begin{aligned} & \begin{bmatrix} \text{delay}(p) \\ \text{trustworthiness}(p) \end{bmatrix} \leq_{delay-trust} \begin{bmatrix} \text{delay}(q) \\ \text{trustworthiness}(q) \end{bmatrix} \\ \iff \quad & \text{delay}(p) \leq \text{delay}(q),\ \text{trustworthiness}(p) \geq \text{trustworthiness}(q) \end{aligned}$$

Let $\text{Min}^{delay-trust}$ indicate the non-dominance function for this order relation. Then the Pareto optimal trusted routing problem can be expressed as

$$(\hat{\mathbb{R}}_+ \times [0,1], \text{Min}^{delay-trust}, \begin{bmatrix} + \\ \times \end{bmatrix})$$

As along as the both the link metrics, trust and delay, are non-zero, the inflatory condition is satisfied and the generalized local pruning theorem will hold.

4.5 Summary

In this chapter, we presented a generalized local pruning condition for rule-based routing. We show that the strict inflatory property, which is analogous to the strict positivity condition in SPTC, is a sufficient condition to ensure loop-free pruning under this generalized pruning condition. To illustrate the diversity of this framework, we show two example applications: hop-count lexicographic extension to SPTC and Pareto optimal trusted routing.

Chapter 5

Conclusions and Future Work

We formulated the topology control problem for QoS guarantees as a multi-agent graph pruning problem. The agents have access to only their local neighborhood information. The notion of local neighborhood is abstracted using subgraphs called local views. The agents, using this constrained view of the network, select a minimal local topology to construct a pruned graph for routing. This subset of link state information is broadcast in the network, and it constitutes another subgraph called broadcast view. Consequently, the total information available for routing at every host is the graph union of its local view and the broadcast view, which we call the global view. We then pose the QoS preservation as a constraint on the global – it must preserve the globally desirable paths for routing described by a rule for routing.

We identify a class of rules that can be described as an algebraic path problem in idempotent semirings – allows specifying global properties of paths from local properties of the graph. We show that this class of rules satisfy, what we call, the "distribution of order" property that enables local pruning; in essence, this distribution of order property is Bellman's optimality principle extended to path-sets and minimal elements of partially ordered sets. This becomes necessary when you have rules defined over multiple metrics – vector metrics.

The primary contribution of this dissertation is identifying a policy for pruning, which depends only on the local neighborhood, but guarantees that the pruned graph preserves desirable paths described by generic rules of routing. We show that for this local policy to ensure loop-free pruning, it is sufficient to have what is called an inflatory arc composition property.

Thus, the two primary contributions of this dissertation are the following:

1. Extension of algebraic routing to vector metrics, where notions of optimum and optimality are replaced by that of non-dominance and efficiency, respectively. We present a version of Bellman's optimality principle extended to multiple vectors, minimal elements and efficient path-sets.

2. Identification of sufficient local pruning conditions for rule described using idempotent semirings – strict inflatory arc composition – that guarantee loop-free pruning.

5.1 Future Work

In developing the pruning algorithms, we used the strict inflatory arc composition to break ties. This yields a proof method of path construction that ensured loop-freedom in pruning. However, many semirings do not satisfy the strict inflatory assumption. A classic example is the $(\hat{\mathbb{Z}}_+, \min, \max)$ semiring. In this case, the arc composition is a selective operator,

$$a, b \in \hat{\mathbb{Z}}_+,\ \max\{a, b\} = a \text{ or } b,$$

and consequently, the inflatory condition is never satisfied.

The inflatory condition is only a sufficient condition to break ties, and we are looking for alternative methods to break ties to ensure loop-freedom. One of the directions towards future work is identifying a suitable extended metric, similar to the hop-count lexicographic extension of the SPTC problem, that would ensure that ties are broken. Interestingly, applying the hop-count lexicographic extension to the $(\hat{\mathbb{Z}}_{+}, \min, \max)$ semiring, breaks the semiring structure.

On the other hand, it could be that for selective arc compositions, there are no extended metrics which ensure loop-freedom. As future work, we also investigating on such non-existence results.

Appendix A

Graph Theory and Problems in Combinatorics

A.1 Elementary Graph Theory

A graph is a mathematical abstraction used to represent binary relations/ adjacencies between a pair of entities. In this section, we will define terms commonly used in graph theory. Since this dissertation concerns both directed and undirected graphs, we will distinguish between them with different notation.

A directed graph G is an ordered pair of sets $(V,\ A)$. The former, V, is the set of vertices, and the latter, A, is the set of arcs. The set E is a subset of the set of ordered pairs of V. A vertex $v \in V$ is said to be adjacent to another vertex $u \in V$ if $(u, v) \in A$. Note that there is a direction associated with the arc (u, v), i.e., the arc goes from u to v. An undirected graph G is, again, an ordered pair of sets $G(V,\ E)$, where V is the vertex set and E is the edge set. Unlike the arcs of a directed graph, the edges have no direction associated with them: if $(u, v) \in E$, then $(v, u) \in E$. In other words, an edge is unordered pair of vertices $(u, v),\ u, v \in V$. If $(u, v) \in E$, then vertices u and v are adjacent to each other. Thus, undirected graphs are a special kind of directed graphs, where if $(u, v) \in A$, then $(v, u) \in A$. For the rest of the section, we will define terms w.r.t directed graphs; the corresponding definitions for undirected graph follow from the above statement.

A graph $G' = (V', A')$ is a subgraph of G if $V' \subseteq V$ and $A \subseteq A$ with A' is restricted to $V' \times V'$. If $V' = V$, then G is a spanning subgraph of G.

A path is an ordered sequence of vertices, such that an arc exists between two successive vertices: $(v_1, v_2, \ldots, v_n)$ is a path of G' if and only if $(v_i, v_{i+1}) \in A'$, $1 \leq i \leq n$. The vertex v_1 is the initial vertex of the path and v_k the terminal vertex. The set of paths whose initial vertex is i and terminal vertex is j is denoted by $P_{ij}^{G'}$. The path set $P_{ii}^{G'}$, which contains the set of paths whose initial and terminal vertex are i, includes the self-loop (i, i) and the empty path (i). Two paths can be concatenated if the terminal vertex of the first path is the starting vertex of the second path. For $p = (i = u_1, u_2, \cdots, u_m = k) \in P_{ik}^{G'}$ and $q = (k = v_1, v_2, \cdots, v_n = j) \in P_{kj}^{G'}$, the concatenated path is denoted by $p.q = (i = u_1, u_2, \cdots, u_m = k = v_1, v_2, \cdots, v_n = j)$.

A path with no repetitions of vertices is an elementary path. A cycle is a path in which the initial and terminal vertices coincide. A set of paths is independent if their common vertices are exactly their initial and terminal vertices. Such paths are also called internally disjoint. If there is a path from i to j, then j is reachable from i. For an undirected graph, since the edges are symmetric, if j is reachable from i, then i is reachable from j. If every vertex is reachable from every other vertex, the graph is connected. A connected directed graph is strongly connected.

The set of subsets of paths from $i \in V$ to $j \in V$ is denoted by the power-set $2^{P_{ij}^{G'}}$. The elements of this power set are denoted in characters of the calligraphic alphabet, i.e., $\mathcal{P} \in 2^{P_{ij}^{G'}}$. For two path-sets $P \subseteq P_{ik}^{G'}$ and $Q \subseteq P_{kj}^{G'}$, the concatenated path-set is Minkowski concatenation, denoted by $P.Q = p.q : \ p \in P, q \in Q$.

Since we deal with different subgraphs in this dissertation, we usually annotate the different quantities, such as the set of paths, with a superscript notation indicating the subgraph to which the quantity belongs. For example, in the above explanation, the set of paths in the subgraph G' from i to j is denoted by $P_{ij}^{G'}$.

A.2 Classical Shortest Path Problem

Given a directed graph $G(V, A)$ with a real valued edge weight function $a : A \rightarrow \hat{\mathbb{R}}$, i.e., $a_{uv} \in \hat{\mathbb{R}}$. For non-existent arcs, $(u, v) \notin A$, $a_{uv} = \infty$. The weight of a path $p = (v_1, v_2, \cdots, v_n)$ is the sum of the weights of its arcs:

$$w_p = \sum_{i=1}^{n-1} a_{v_i v_{i+1}}.$$

The shortest path weight from vertex i to vertex j in G is the smallest path weight among all paths that start from i and end at j in G:

$$\min_{p \in P_{ij}^{G}} w_p$$

If there is no path from i to j in G, i.e., P_{ij}^{G} is empty, the shortest path weight is defined to be ∞.

If a negative cycle exists in G and is reachable from i, then there is no shortest path to any destination j. This is because the negative cycle can be traversed infinitely many times to yield a path length of $-\infty$. If there are no negative cycles, then the shortest path weight will be unique; although, the number of shortest might not be. If all the cycles are positive, then the shortest paths will be elementary.

The Single Source Shortest Path problem (SSSP) amounts to finding the shortest paths from a given vertex i to all other vertices of G. The Bellman-Ford and Dijsktra algorithms are two popular algorithms to solve the SSSP. The All-Pairs

Shortest Path Problem (APSP) amounts to finding the shortest paths from every vertex $i \in V$ to every other vertex of G. The Floyd-Warshall algorithm is a popular algorithm to solve the APSP.

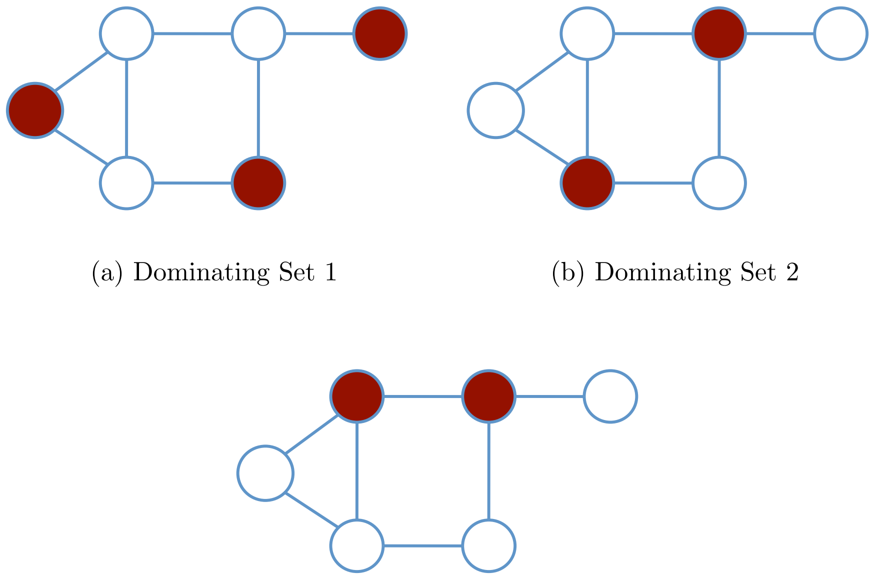
A.3 Connected Dominating Sets

Given a graph $G(V, E)$, where V is the vertex set and E is the edge set, a set of vertices $D \subseteq V$ is said to be a Dominating Set (DS) if for any $v \in V$, either $v \in D$ or there exists $u \in D$ such that $(u, v) \in E$. If the subgraph induced by D is connected, the D is called a Connected Dominating Set (CDS). For example, Fig. A.1 shows three dominating sets for an example graph, where Subfigure A.1c is a CDS and the other two are dominating sets, but not connected dominating sets.

Connected dominating sets constructions arise in broadcast problems in wireless multi-hop networks because a CDS forms an overlay network that guarantees network-wide broadcast: activating the nodes corresponding to the CDS vertices ensures that the information is broadcast, via flooding, to all the nodes in the network. A typical problem is to identify the minimal CDS, the one with the minimum size, $|D|$. This problem is shown to be a NP-complete. However, there are other problems such as finding the max-capacity CDS, which have polynomial-time solutions.

A.4 Set Cover Problem

The set cover problem, as the name signifies, is a problem concerned with sets. Given a universe U, which is a set, of elements. A cover, $\mathcal{C}$, is set of subsets of U,



(a) Dominating Set 1

(b) Dominating Set 2

(c) Connected Dominating Set

Figure A.1: Dominating Sets of G

i.e., $\mathcal{C} \subset 2^U$, such that $\cup_{C \in \mathcal{C}} C = U$.

In the set covering decision problem, the question is whether there is a set cover $\mathcal{C}$ of size k or less. In the set covering optimization problem, and the objective is to find a set cover $\mathcal{C}$ that uses the fewest number of sets. The decision version of set covering is NP-complete, and the optimization version of set cover is NP-hard.

Appendix B

Semiring Algebra

B.1 Order theory

Consider a set X. A *partial order* relation on X is a binary relation $\leq$ such that $\forall x, y, z \in X$ satisfies:

i. *Reflexivity* $x \leq x$

ii. *Antisymmetry* $x \leq y$ and $y \leq x \Rightarrow x = y$

iii. *Transitivity* $x \leq y$ and $y \leq z \Rightarrow x \leq z$

The corresponding *strict order* relation for $x, y \in X$ is

$$x < y \iff x \leq y,\ x \neq y.$$

In a partially ordered set, not all elements are necessarily comparable, i.e., $x || y \Rightarrow x \nleq y$ and $y \nleq x$. Here $||$ is the *incomparability* relation. Another important order relation is the *covering relation*:

$$x \prec y \iff (x < y \text{ and } x \leq z < y \Rightarrow x = z).$$

The covering relation $x \prec y$ implies that there exists no other element in between x and y in the ordered set X. In this case, x is called the *covered element* of y, and y

is called the *covering element* of x. A *totally ordered* set X satisfies an additional *trichotomy* condition:

$$x, y \in X \Rightarrow x \leq y \text{ or } y \leq x.$$

Another characteristic of ordered sets is that they satisfy the *duality principle*: Given an ordered set X, we can construct its dual ordered set X^∂ by defining $x \leq y$ to hold in X^∂ *iff* $y \leq x$ in X. $\perp \in X$ is the *bottom* element if $\perp \leq x, \forall x \in X$. Dually, the *top* element $\top$ is the bottom element of X^∂.

Given an set X and order relation $\leq$, we can define the minimal elements of a any $S \subset X$: an element $x \in S$ is a minimal element of S, there exists no $y \in S$, $y \neq x$ such that $y \leq x$. $\text{Min}(S)$ denotes the set of all minimal elements of S. Any $x \in \text{Min}(S)$ is said be a non-dominated point in S. The function that Min that works on sets in S is called the *non-dominance* function. If the set S is totally ordered the non-dominance function Min reduces the standard min operator.

B.1.1 Orders in Multi-criteria Optimization

Multi-criteria optimization problems are represented using the tuple

$$(P, w, S, \leq),$$

where P is called the decision set, S is the vector valued objective set, $w : P \to S$ is a vector valued function that maps the decision set, P, to the objective set, S. The pair $(S, \leq)$ is an ordered set, ordered by $\leq$.

A decision point $p \in P$ is said to efficient if and only if $f(p)$ is a non-dominated

Order Type ($\leq$)	Definition	Non-Dominance Function
Componentwise, $\underline{x} \leq_{com} \underline{y}$	$\underline{x}(i) \leq \underline{y}(i) \quad i = 1, 2, \ldots, m$	Min^{com}
Lexicographic, $\underline{x} \leq_{lex} \underline{y}$	$\underline{x}(k) < \underline{y}(k)$ or $\underline{x} = \underline{y}$,where $k = \min\{i : \underline{x}_i \neq \underline{y}_i\}$	Min^{lex}
Max-order, $\underline{x} \leq_{MO} \underline{y}$	$\max\{\underline{x}(1), \underline{x}(2), \ldots, \underline{x}(m)\} \leq \max\{\underline{y}(1), \underline{y}(2), \ldots, \underline{y}(m)\}$	Min^{MO}

Table B.1: Table of Orders and Induced Laws for $\underline{x}, \underline{y} \in S = \times_{1 \leq l \leq m} S_l$, where S_l is a totally order set.

in S, i.e.,

$$f(p) \in \mathrm{Min}(S).$$

For most problems in multi-criteria optimization, the objective set S, is composed of totally ordered sets $S_l,\ 1 \leq l \leq m$, where $S = \times_{1 \leq l \leq m} S_l$. In such cases, there are three common orders in multi-criteria optimization: the definition for these orders is shown in Table B.1. Note that the componentwise order is called Pareto order.

B.2 Canonically Ordered Monoids

A monoid is an algebraic structure $(S, \oplus)$ that satisfies the following axioms, for $a, b, c \in S$:

$$
\begin{aligned}
&a \oplus b \in S \\
a \oplus (b \oplus c) &= (a \oplus b) \oplus c \\
a \oplus \textcircled{0} &= a,
\end{aligned}
$$

where $\textcircled{0} \in S$ is called the neutral element of the monoid.

A monoid $(S, \oplus)$ is said to be commutative if $a, b \in S$,

$$a \oplus b = b \oplus a.$$

A commutative monoid $(S, \oplus)$ is said to be *canonically ordered* if the set S is canonically ordered, i.e.,

$$a, b \in S,\ a \leq b \iff \exists c \in S \text{ such that } b = a \oplus c,$$

and the order relation is compatible with the internal law, i.e.,

$$a, b, c \in S\ a \leq b \ \Rightarrow\ a \oplus c \leq b \oplus c.$$

B.3 Semirings, Rings and Diods

A semiring is an algebraic structure $(S, \oplus, \otimes)$ that satisfies the following axioms:

(A1) (S,⊕) is a commutative monoid with a neutral element $⓪$:

$$\begin{aligned} a \oplus b &= b \oplus a \\ a \oplus (b \oplus c) &= (a \oplus b) \oplus c \\ a \oplus ⓪ &= a \end{aligned}$$

(A2) (S,⊗) is a monoid with a neutral element $①$*, and an absorbing element* $⓪$:

$$\begin{aligned} a \otimes (b \otimes c) &= (a \otimes b) \otimes c \\ a \otimes ① = ① \otimes a &= a \\ a \otimes ⓪ = ⓪ \otimes a &= ⓪ \end{aligned}$$

(A3) $\otimes$ *distributes over* $\oplus$:

$$\begin{aligned} a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c) \\ (a \oplus b) \otimes c &= (a \otimes c) \oplus (b \otimes c) \end{aligned}$$

The class of semirings can be naturally subdivided into two disjoint sub-classes depending on whether the law $\oplus$ satisfies one of the following two properties:

1. The law $\oplus$ endows the set S with a group structure.

2. The law $\oplus$ endows the set S with a canonically ordered monoid structure.

The two cases cannot be satisfied simultaneously. Case 1 leads what is commonly called as a *ring* structure, and Case 2 gives a *Dioid* structure. In this dissertation, our focus will be on dioids.

The idempotent semiring class, for which the $\oplus$ is idempotent, i.e.,

$$a \in S,\ a \oplus a = a,$$

induces a canonical order in $(S, \oplus)$. Consequently, all idempotent semirings are dioids.

B.4 Semiring Algebraic Path Problems

To motivate the Semiring Algebraic Path Problem (SAPP) we will introduce three examples.

The first example is that of the shortest path computation, which is used in several applications such as data network routing and web mapping. Consider

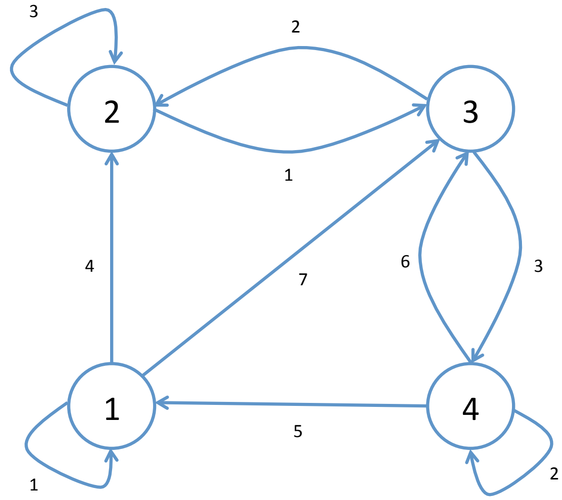


Figure B.1: Example network for shortest path computation

the directed graph shown in Figure B.1 with arc weights denoted by a_{uv}. For this computation, the weight of the path p is given by the rule $w_p = \sum_{(u,v)\in p} a_{uv}$, and the shortest path weight (aggregate metric) between a pair of vertices i, j is given by the rule $x_{ij} = \min_{p \in P_{ij}} w_p$. Clearly, the rules of composition can be described by the $(\hat{\mathbb{Z}}_+, \min, +)$ semiring algebra ($\textcircled{0}$=∞, $\textcircled{1}$=0). Further, these compositions have a structure that can be expressed by a system of equations. Let the weighted adjacency matrix of graph in Figure B.1 be denoted by

$$\mathbf{C} = \begin{bmatrix} 1 & 4 & 7 & \infty \\ \infty & 3 & 1 & \infty \\ \infty & 2 & \infty & 3 \\ 5 & \infty & 6 & 2 \end{bmatrix}.$$

The artificial weights $\infty = \textcircled{0}$ are used for non-existent arcs. Consider the shortest path from i to j. If $i \neq j$, then this path is of the form $(i = u_0, u_1, \ldots, u_l = j)$. For this shortest path, the sub-path $p' = (u_1, u_2, \ldots, u_l = j)$ must be the shortest path from u_1 to j, and consequently, the shortest path metric is given by $x_{ij} = c_{ik} + x_{kj}$, for $k = u_1$. Thus, the shortest path metric computation for $i \neq j$ can be written as $x_{ij} = \min_{k \in V}(c_{ik} + x_{kj})$. For $i = j$, we also need to consider the empty path from j to j. For the shortest path computation, the weight of an empty path is $0 (= \textcircled{1})$. Thus, the shortest path computation from j to j can be expressed as $x_{jj} = \min\{\min_{k \in V}(c_{jk} + x_{kj}), 0\}$. For all pairs of vertices, we can express these computations as a system of equations:

$$\begin{aligned} x_{ij} &= \min_{k \in V}(c_{ik} + x_{kj}), \text{ for } i \neq j, \text{ and} \\ x_{jj} &= (\min_{k \in V}(c_{jk} + x_{kj})) \min 0. \end{aligned}$$

For the example in Figure B.1, the unique solution of shortest path lengths to system of equations is

$$X = \begin{bmatrix} 0 & 4 & 5 & 8 \\ 9 & 0 & 1 & 4 \\ 8 & 2 & 0 & 3 \\ 5 & 8 & 6 & 0 \end{bmatrix}.$$

Note that for each pair of vertices i and j, the solution corresponds to exactly one path from i to j in G.

The next example is a bi-objective version of the shortest path problem. Consider an example network shown in Fig. B.2. It is identical to the network in the previous example, Fig. B.1, except for the weights, which are extended to vector

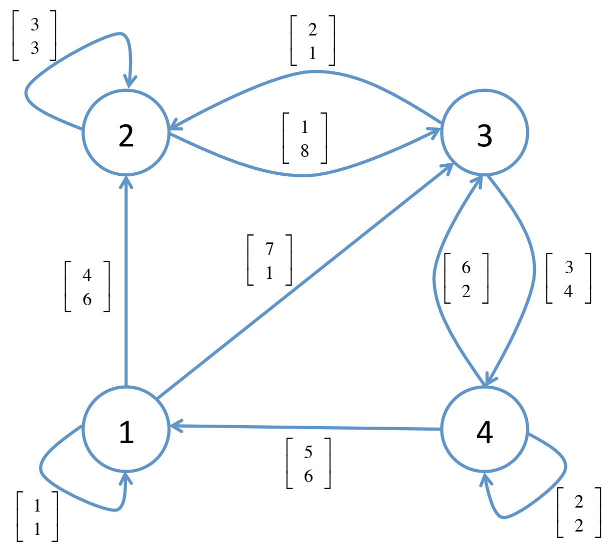


Figure B.2: Example network for bi-objective shortest path computation

weights. In the case, the weight of a path p is given by vector addition $w_p = \sum_{(u,v)\in p} \underline{a}_{uv}$. Consider the paths from vertex 1 to vertex 4: path $(1,3,4)$ has a weight $[10,5]^T$ and path $(1,2,3,4)$ has a weight $[8,18]^T$. Each of the paths has a smaller value for one of the two metrics. In such a setting, optimality is usually defined in a Pareto sense [19]. A vector $\underline{v} \in \hat{\mathbb{Z}}^2_+$ is said be Pareto efficient with respect to a subset $F \subseteq \hat{\mathbb{Z}}^2_+$ if there does not exist in F a vector $\underline{v}' \neq \underline{v}$ that is componentwise smaller than or equal to v. A set of paths is said be to Pareto efficient if its vector weights are Pareto efficient. The Pareto efficient path problem is defined in terms of path-sets rather than paths (Section 6.7 of [22]), and the corresponding Pareto solutions are subsets of $\hat{\mathbb{Z}}^2_+$. For closure, the arc weights need to be in $2^{\hat{\mathbb{Z}}^2_+}$, which is the power-set of $\hat{\mathbb{Z}}^2_+$. For the example in Fig. B.2, the weighted adjacency matrix is given by

$$\mathbf{C} = \begin{bmatrix} \{[1,1]^T]\} & \{[4,6]^T\} & \{[7,1]^T\} & \{[\infty,\infty]^T\} \\ \{[\infty\,\infty]^T\} & \{[3,3]^T\} & \{[1,8]^T\} & \{[\infty,\infty]^T\} \\ \{[\infty,\infty]^T\} & \{[2,1]^T\} & \{[\infty,\infty]^T\} & \{[3,4]^T]\} \\ \{[5,6]^T\} & \{[\infty,\infty]^T\} & \{[6,2]^T\} & \{[2,2]^T\} \end{bmatrix}.$$

Consequently, for the arc composition, we need to define a rule that works on efficient sets (corresponding to Pareto efficient paths). For example, the Pareto efficient paths from vertex 1 to vertex 4 of Fig. B.2, i.e., $(1,3,4)$ and $(1,2,3,4)$, are composed of the Pareto efficient paths from vertex 1 to vertex 3, i.e., $(1,3)$ and $(1,2,3)$ respectively, and the arc $(3,4)$. The composition can be expressed using the rule x_{14} = Pareto efficient vectors of the set $\{x_{13} + c_{34}\}$. The path-set composition rule for two path-sets selects all the Pareto efficient vectors in the union of the weights of the two path-sets. Formally, for $X, Y \in 2^{\hat{\mathbb{Z}}^2_+}$, the arc composition rule is

$$X +^P Y = \text{Pareto efficient vectors of the set } X + Y,$$

where $X + Y = \{x + y : \; x \in X, \; y \in Y\}$, and the path composition rule is

$$X \text{ Min } Y = \text{Pareto efficient vectors of the set } X \cup Y.$$

(Note that this Min operator is different from the standard min operator for single-metric path problems)

The Pareto efficient paths are then given by

$$\begin{aligned} x_{ij} &= \text{Min}_{k \in V}(c_{ik} +^P x_{kj}), \text{ for } i \neq j, \text{ and} \\ x_{jj} &= (\text{Min}_{k \in V}(c_{jk} +^P x_{kj}))\text{Min}\emptyset. \end{aligned}$$

Again, the tuple $(2^{\hat{\mathbb{Z}}_+^2}, \mathrm{Min}, +^P)$ forms a semiring with $①=\{[0,0]^T\}$ and $⓪=\emptyset$ (where the Pareto efficient vector of $\emptyset$ is defined to be $[\infty, \infty]^T$).

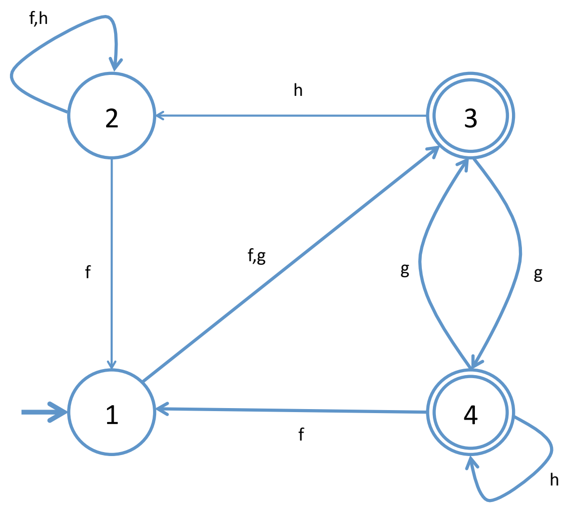


Figure B.3: Transition diagram of a finite automaton. The initial state is 1 and the final states, marked by double circles, are 3 and 4

The last example is the language accepted by a finite automaton. A finite automaton is a machine that reads words (sequence of symbols over some alphabet Σ) and either accepts them or rejects them. It is described by a *transition diagram*. Consider a transition diagram, represented by a directed graph, shown in Figure B.3. The vertices are the states of the automaton. Vertex 1 is the start state, and vertices 3 and 4, which are marked with double circles, are the final states. The arcs are labeled by subsets of letters from $\Sigma = \{f, g, h\}$. The automaton starts in the start state and reads symbols of an input word one by one. A label z on arc (u, v) means the following: If the automaton is in state u and the next symbol that it reads is in z, it can go to state v. When the automaton is in state u and there is no arc labeled z that leaves u, the automaton cannot continue and stops. The automaton is said to *accept* a word if there is a sequence of state transitions leading from the start state to the final state while reading this word. In other words, let $p = (v_0, v_1, \ldots, v_l)$ be a path from the start state v_0 to a final state v_l. If z_i is in a label of the arc (v_{i-1}, v_i), for $1 \leq i \leq l$, then the word $z_1 z_2 \ldots z_n$ is finitely accepted by the automaton. For example, the automaton shown in Figure B.3 accepts the word $fgghhfhfff$ because it leads from state 1 to state 3 via the path $(1, 3, 4, 3, 2, 2, 2, 2, 2, 1, 3)$. Thus, in essence, the automaton defines a formal language (subset of words) over the alphabet Σ. The problem of determining the formal language is "for each each pair of start and final states (i, j) of a finite automaton, determine the set x_{ij} of words that can lead from the initial state i to the final state j."

We need more notations to solve the above problem: A *word* is a finite sequence of letters/symbols. We denote an *empty word* by ϵ, which contains no symbols. The concatenation of two words a and b is denoted by $a.b$. If A and B are sets of words, then $A.B$ denotes the set $\{a.b \ : \ a \in A, b \in B\}$. The concatenation of any word a with a non-existent label $\emptyset$ is defined to be absorbing: $a.\emptyset = \emptyset.a = \emptyset$. Again we can introduce artificial weights of $\emptyset$ for the non-existent arcs of the directed graph.

Similar to the shortest path problem, we can set up a system of equations. Let the arc labels be denoted by $c_{uv} \subseteq \Sigma$, $(u, v) \in A$. The non-existent arcs have a $\emptyset$ weight and the empty paths have a ϵ weight. x_{ij} denotes the set of words that can lead the automaton from i to j. For $i \neq j$, the set of words that can lead from

i to j via k is $c_{ik}.x_{kj}$. Now, we need to take the union over all possible states $k \in V$ to obtain $x_{ij} = \cup_{k \in V} c_{ik}.x_{kj}$. If $i = j$, we need also consider the empty path, and consequently, we get the following system of equations.

$$
\begin{aligned}
x_{ij} &= \cup_{k \in V}(c_{ik}.x_{kj}), \text{ for } i \neq j, \text{ and} \\
x_{jj} &= (\cup_{k \in V}(c_{jk}.x_{kj})) \cup \epsilon.
\end{aligned}
$$

The above computations can be described by the $(\Sigma^*, \cup, .)$ semiring, where Σ^* is the free monoid over the alphabet Σ [22], which is the set of all finite words of the alphabet Σ.

In the above seemingly different examples, the computations of the aggregate metric over the two different semirings appear to have a common structure: Instead of computing the weight of every path, w_p for all $p \in P_{ij}$ and then computing the aggregate metric by path composition, the semiring distribution (Axiom *A3*) factors out the common terms of the computation, thereby expressing the aggregate metric in terms of the aggregate metrics of the intermediate vertices. This can be generalized as follows. For a directed graph $G(V, A)$ labeled with elements from an arbitrary semiring $(S, \oplus, \otimes)$ $(c_{uv},\ (u, v) \in A)$, artificial arc weights of ⓪ for the non-existent arcs, and empty path weight ①, the generalization of the computation of the above examples is given by

$$
\begin{aligned}
x_{ij} &= \oplus_{k \in V}(c_{ik} \otimes x_{kj}), \text{ for } i \neq j, \text{ and} \\
x_{jj} &= (\oplus_{k \in V}(c_{jk} \otimes x_{kj})) \oplus ①. \qquad \text{(B.1)}
\end{aligned}
$$

This fixed point equation (Equation (B.1)) is called Semiring Algebraic Path

Problem (SAPP).

Bibliography

[1] Lim . and Kim C. Flooding in wireless ad hoc networks. *Computer Communications Journal*, pages 353–363, 2001.

[2] S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey. Route-lifetime assessment based routing (rabr) protocol for mobile ad-hoc networks. In *IEEE International Conference on Communications*, volume 3, pages 1697–1701, 2000.

[3] Othman A. Al-Amoudi, Mohamed S. El-Azhari, Mike Woodward, and Irfan Awan. Performance analysis of angle routing in manets. In *Lecture Notes in Computer Science*, pages 21–29, 2008.

[4] Th. Aure and F. Y. Li. Load-balancing in multi-homed olsr networks using airtime metric and radio-aware path selection: implementation and testing. White paper, 2008.

[5] J. S. Baras, S. Perumal, V. Tabatabaee, K. Somasundaram, and P. Purkayastha. Loss network models and multiple metric performance sensitivity analysis for mobile wireless multi-hop networks. In *4th Wireless Internet Conference*, 2008.

[6] Perkins C. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, July 2003.

[7] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing: Special Issues on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[8] Balakrishnan H. Chen B., Jamieson K. and Morris R. Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, pages 481–494, 2002.

[9] Z. Cheng and W. B. Heinzelman. Exploring long lifetime routing (llr) in ad hoc networks. In *International workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 203–210, 2004.

[10] T. Clausen, C. Dearlove, and J. Dean. Mobile ad hoc network (manet) neighborhood discovery protocol (nhdp). Draft-IETF, October 2009.

[11] T. Clausen, C. Dearlove, and P. Jacquet. The optimized link state routing protocol version 2. Draft-IETF, September 2009.

[12] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). RFC, Oct 2003.

[13] D. S. J. De Couto. *High-throughput routing for mullti-hop wireless networks*. PhD thesis, MIT, 2004.

[14] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *9th Internation Conference on Mobile Computing and Networking*, pages 134–146, 2003.

[15] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks: Shortest path is not enough. *ACM SIGCOMM Computer Communication Review*, 33:83–88, 2003.

[16] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[17] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-radio wireless mesh networks. In *ACM International Conference on Mobile Computing and Network*, pages 114–128, 2004.

[18] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability-based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communication*, 4(1):36–45, 1997.

[19] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[20] Dai F. and Wu J. Distributed dominant pruning in ad hoc wireless networks. In *ICC*, 2003.

[21] M. Gerharz, C. d. Waal, M. Frank, and P. Martini. Link stability in mobile wireless ad hoc networks. In *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, pages 30–42, 2002.

[22] M. Gondran and M. Minoux. *Graphs, Diods and Semirings - New Models and Algorithms*. Springer, 2008.

[23] T. G. Griffin. The stratified shortest-paths problem. In *COMNETS*, 2010.

[24] Z. J. Haas, M. R. Pearlman, and P. Samar. The zone routing protocol for ad hoc networks. IETF Draft, July 2002.

[25] Y. Han, R. J. La, A. M. Makowski, and S. Lee. Distribution of path durations in mobile ad hoc networks: Palm's theorem to the rescue. *The International Journal of Computer and Telecommunications Networking: Special Issue on Network Modeling and Simulation*, 50:1887–1900, 2006.

[26] S. Haykin. *Communication Systems*. Wiley, 2000.

[27] Sucec J. and Marsic J. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. Technical report, Rutgers University, 2000.

[28] Wu J. and Li H. On calculating connected dominating set for efficient routing in ad hoc wireles networks. In *International Workshop Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, 1999.

[29] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance analysis of olsr multipoint relay flooding in two ad hoc wireless network models. Technical report, INRIA, September 2001.

[30] D. Johnson and Y. Hu. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. RFC, Feb 2007.

[31] Baras J.S., Tabatabaee V., Purkayastha P., and Somasundaram K. Component based performance modeling of the wireless routing protocols. In *IEEE ICC Ad Hoc and Sensor Networking Symposium*, June 2009.

[32] J. Kleinberg and É. Tardos. *Algorithm Design*. Addison Wesley, 2005.

[33] C. E. Koksal and H. Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):1984–1994, 2006.

[34] V.N. Kolokoltsov. Idempotent structures in optimization. *Journal of Mathematical Sciences*, 104(1):847–880, 2001.

[35] V.N. Kolokoltsov and V.P. Maslov. *Idempotent Analysis and its Applications*. Kluwer Academic Publishers, 1997.

[36] K. Kowalik, B. Keegan, and M. Davis. Making olsr aware of resources. In *International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1488–1493, 2007.

[37] S. Lee, B. Bhattacharjee, A. Srinivasan, and S. Khuller. Efficient and resilient backbones for multihop wireless networks. *IEEE Transactions on Mobile Computing*, 7(11):1349–1362, 2008.

[38] G. Lim, K. Shin, S. Lee, H. Yoon, and J. S. Ma. Link stability and route lifetime in ad-hoc wireless networks. In *Proceedings of the International Conference on Parallel Processing Workshops*, page 116, 2002.

[39] J. Moy. Ospf version 2. RFC 2328.

[40] S. Obilisetty, A. Jasti, and R. Pendse. Link stability based enhancements to olsr (ls-olsr). In *IEEE 62nd Vehicular Technology Conference*, pages 306– 310, 2005.

[41] OPNETTechnologies. Opnet modeler. URL.

[42] D. Passos, D. V. Teixeira, D. C. Muchaluat-Saade, L. C. S. Magalhaes, and C. V. N. de Albuquerque. Mesh netwok performance measurements. In *International Information and Telecommunications Technologies Symposium*, 2006.

[43] W. Peng and X. Lu. On reduction of broadcast redudundy in mobile ad hoc networks. In *ACM symposium of Mobile Ad Hoc Networking and Computing*, pages 129–130, 2000.

[44] W. Peng and X. Lu. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology - Beijing China*, 2002.

[45] C. E. Perkins. *Ad hoc networking*. Addison Wesley, 2001.

[46] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*, pages 3866–3875, 2001.

[47] H. Rogge, E. Baccelli, and A. Kaplan. Packet sequence number based etx metric for mobile ad hoc networks. IETF Draft, December 2009.

[48] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In *ACM Internation Symposium on Mobile Ad Hoc Networking and Computing*, pages 245–256, 2003.

[49] N. Sarma and S. Nandi. Route stability based qos routing in mobile ad hoc networks. *Wireless Personal Communications*, 2009.

[50] S. Sitarz. Pareto optimal allocations and dynamic programming. *Annals of Operations Research*, 172(1):203–219, 2009.

[51] J. L. Sobrinho. Algebra and algorithms for qos path computation and hop-by-hop routing in the internet. *IEEE/ACM Transactions on Networking*, 2002.

[52] K. K. Somasundaram and J. S. Baras. Semiring pruning for information dissemination in mobile ad hoc networks. In *Workshop on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks*, 2009.

[53] K.N. Sridhar and M.C. Chan. Stabilty and hop-count based approach for route computation in manet. In *International Conference on Computer Communications and Networks*, pages 25–31, 2005.

[54] Seddigh M. Stojmenovic I. and Zunic J. Dominating sets and neighbour elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, pages 14–25, 2002.

[55] W. Su, S.-J. Lee, and M. Gerla. Mobility prediction in wireless networks. In *MILCOM*, volume 1, pages 491–495, 2000.

[56] A. P. Subramanian, M. M. Buddhikot, and S. C. Miller. Interference aware routing in multi-radio wireless mesh networks. In *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, pages 55–63, 2006.

[57] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 24(2):318–328, 2006.

[58] C.-K. Toh. Associativity-based routing for ad hoc networks. *Wireless Personal Communications*, 4:103–139, 1997.

[59] Lou W. and Wu J. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, pages 111–123, 2002.

[60] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad-hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.

[61] J. Wu and F. Dai. A generic distirbuted broadcast scheme in ad hoc wireless networks. *IEEE Transactions of Computers*, 53(10):1343–1354, 2004.

[62] www.olsr.org. http://www.olsr.org/docs/readme-link-quality.html.

[63] P. Yang and B. Huang. Qos routing based on link stability with dynamic delay prediction in manet. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, volume 1, pages 515–518, 2008.