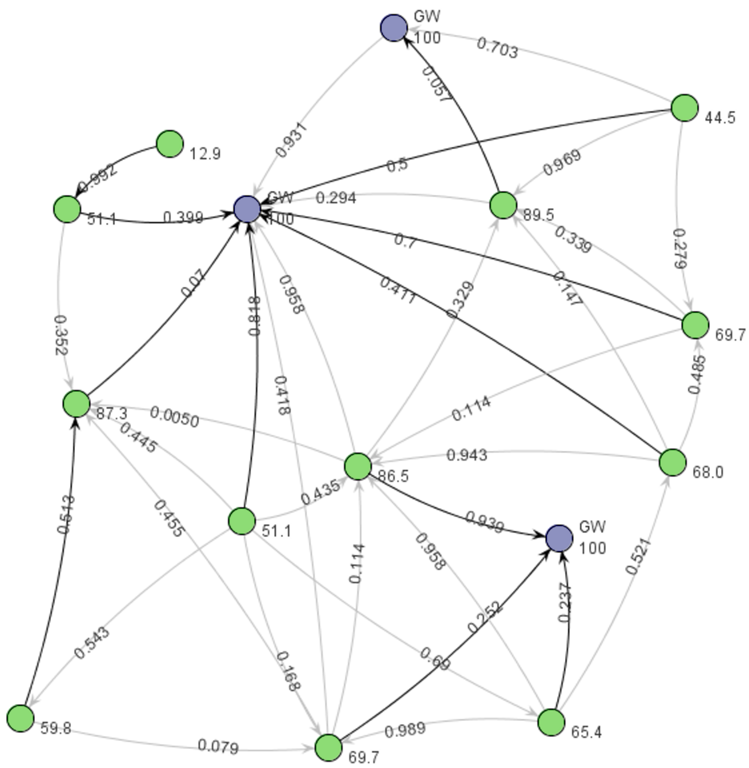
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze



The Impact of the Choice of Metrics for Routing

With a Focus on Wireless Mesh Networking (WMN)

Andreas Weibel
andreas.weibel@gmx.ch

Master's Thesis MA-2006-13
April 2006 to October 2006
Advisors: Rainer Baumann, Mario Strasser, Simon Heimlicher
Supervisor: Prof. Dr. Bernhard Plattner

Abstract

Wireless Mesh Networks (WMNs) are going to change fundamentally how communication systems will work in the near future. However, a number of open questions need to be addressed to make this advance possible. This thesis delves into one of these questions: What routing metrics should be used for WMNs in which situation; and how?

In this thesis, we investigate the basic mathematical properties and definitions of routing metrics. One element of this analysis is constituted by a taxonomy for the classification of metrics. This taxonomy includes five different aspects: The factors that have an impact on the metric, the mathematical properties, the design perspective, the implementation characteristics, and a qualitative evaluation. Using this scheme, 61 routing metrics are studied in this thesis.

Condensing our findings, we propose two metrics: The first metric is suited for energy-restricted networks consists of the energy that is consumed per packet transfer, the remaining battery charge in the involved nodes and the packet loss ratio. The other metric is designed for finding routes with maximal bandwidth and combines the nominal bandwidth with packet loss ratio, as well.

Furthermore, we propose and analyse several methods for integrating additive, multiplicative and concave routing metrics into Field-Based Routing (FBR) algorithms. In Addition, we present a software tool that allows to simulate field construction and route selection functions for various FBR algorithms.

Acknowledgements

The author would like to warmly thank the following persons: Rainer Baumann, Mario Strasser, Simon Heimlicher, Martin May, Vincent Lenders, and Bernhard Plattner for their help and advice; Arno Wagner and Alain Breedt for their technical expertise when it was urgently needed.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

A vision is haunting the telecommunication community: Mobile communication devices that do not depend on costly, centralised infrastructure anymore, but interact directly with each other and organise themselves in distributed, dynamic and highly adaptive networks. These networks – commonly referred to as Wireless Mesh Networks (WMNs) – would provide inexpensive or cost-free, ubiquitous access to the Internet and hence also access to voice or even video communication. These speculations have been fueled by promising new communication technologies such as IEEE 802.11 or WiMAX and new, ever cheaper and more powerful mobile devices.

Wireless Mesh Networks commonly are thought to bear the potential to revolutionise the world of telecommunication like mobile cell phones or the Internet did. The spreading of WMNs would presumably not only go in hand with a technical, but also an economic decentralisation. Some voices predict that the consequences for the telecommunication industry will be as far-reaching as the transition from strictly wired networks to the mobile cell phone networks of today; a transition which was accompanied by the fall of the monopolistic, state-owned mammoth companies of the era before [1].

Currently, WMNs are still far from being ready for market. While there have been promising developments in some fields, especially on the physical layer, a large number of other questions remain open. In this thesis, we will approach one of them: Which routing metric should be used and how?

A broad range of different routing protocols for Wireless Mesh Networks have been proposed. However, the vast majority of these protocols base on hop count metrics. It is widely acknowledged that this is by far not optimal and finding more sophisticated metrics thus represents an important open research issue [2; 3]. While some work has been done on specific problems or specific metrics, there is no comprehensive analysis of routing metrics up to now to the best of our knowledge. Although routing metrics are a fundamental component of any routing algorithm, only few work has been published on the basic aspects of this topic. With this thesis, we aim at closing this gap.

1.2 Related Work

Plenty of literature exists that analyses specific metrics. There are two main focuses of these works: In the area of sensor networks, especially energy-saving metrics have been investigated. And for general networks, traffic-based metrics have been explored. We will present this literature extensively in Chapter 4. The basis for Field-Based Routing, which will be fundamental for the last Chapters of this thesis, were laid by Baumann et al. [4; 5] and Lenders et al. [6; 7].

Only a few works have been published on the fundamental concepts of routing metrics. Most notable are Khanna and Zinkys article from 1989 [8], which approached the problem of self-interference (see Section 4.1.1 of this thesis) for the first time; Wang and Crowcroft's paper on metrics for QoS systems [9], in which they formalise and analyse multi-dimensional metrics; and Sobrinho's works [10; 11] in which he introduces an algebraic system to describe routing algorithms.

1.3 Overview

The remainder of this thesis is organised in the following manner: In Chapter 2, a brief review of the most important terms and concepts of routing and networking in general is given. In the following chapter, we will concentrate on the basics of metrics and, in particular, propose a definition of how routing metrics can be described and analysed. Chapter 4 contains an extensive survey of the routing metrics we found in literature. In Chapter 5, we propose and analyse a number of methods how metrics can be incorporated into Field-Based Routing. In Addition, we present a tool that allows to simulate and study the behaviour of different field calculation and route selection functions for FBR algorithms in Chapter 6. In the terminal conclusions chapter, we present the two metrics that we favour and describe what topics we propose for future work.

1.4 Task Assignment

The goals of this thesis are:

- To create a detailed list of routing metrics
  - To describe and characterise these routing metrics, their aim, their usage, their advantages and shortcomings.
  - If possible, to decompose metrics in fundamental ones.
  - To analyse and categorise the metrics.
  - To do this with a special focus on Wireless Mesh Networks (WMNs).
- To propose a set of optimal metrics for WMNs and test these metrics with Field-Based Routing.
- To deliver a practical and serviceable documentation of the work.

Chapter 2

Fundamentals

In this chapter, we will review some basic concepts of networking and routing, with a special focus on Mobile Ad Hoc Networks (MANETs) and Wireless Mesh Networks (WMNs). Thereby, we want to disambiguate all expressions we use in this thesis and provide the reader with the opportunity to refresh the most important ideas on which this thesis is built.

In the first section, we will recapitulate the basic terminology of networking, state different approaches to categorise networks and shortly describe the concept of reference layers. Section 2.2 is dedicated to MANETs, the following section to WMNs. Finally, Section 2.4 will concentrate onto the topic of network routing, especially for MANETs. As one example for a MANET routing algorithm, the Ad Hoc On Demand Distance Vector (AODV) routing algorithm is presented.

2.1 Networking

A network is an abstract model for electronic communication. It consists of a number of entities:

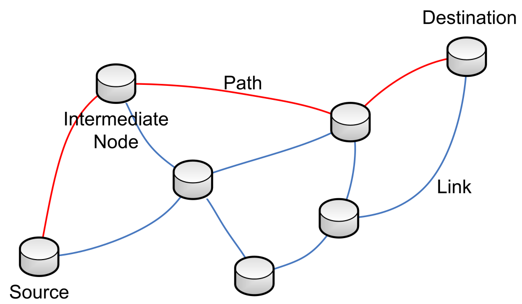


Figure 2.1: Network model: Nodes, links, path (red).

Nodes are the processing, sending and receiving devices. Routers, gateways, network computers or mobile devices can be represented as nodes. Each node has one or more *interfaces* which allows it to communicate with other nodes.

Links are communication connections. Links might be hardware wires or a virtual radio connections. Each link is assigned with a *cost* for using it. This cost is determined by a *metric*.

Traffic is composed of *packets* that are sent over the network links.

Paths are the sequences of links a packet can pass from its source to its destination. Paths are determined and chosen by a *routing algorithm*.

2.1.1 Categorisation

Computer networks can be categorized in manifold manners: The physical layer of a network may be *wired* or *wireless*. A network may be designed for *mobile* or *fixed* nodes. In an *infrastructure* or *cellular* network, a special base station manages the network within its range, whereas in an *ad hoc* network (also called *infrastructureless* or *autonomous* network), no node has a central administrative function. In *multihop* networks, communication is possible over several nodes, while in *singlehop* networks, only direct neighbours can be reached. Often, networks are categorized by their geographical extension (see Table 2.1).

Wireless networks can be categorised in various fashions. One property is the used access technology, for example IEEE 802.11, Bluetooth et cetera. In *multi-rate* networks, the transfer rate is adapted according to the link quality. *Multi-channel* networks provide the ability to have several non-interfering communication channels within the same geographical space.

Category	Typical Range	Example Technologies
Personal Area Networks (PAN)	10 meters	Bluetooth, UWB
Local Area Networks (LAN)	100 meters	Ethernet, WLAN
Metropolitan Area Networks (MAN)	1 kilometer	ATM, FDDI, DQDB
Wide Area Networks (WAN)	Global	Internet

Table 2.1: Geographical categories of networks

2.1.2 Reference Layers

The development of network components is guided by abstract layering models. As each layer is self-contained and only communicates to the upper and lower levels through standardised interfaces, development is greatly facilitated[1]. In this thesis, we will refer to the OSI reference model (see Figure 2.2) when we describe the interaction of different logical layers.

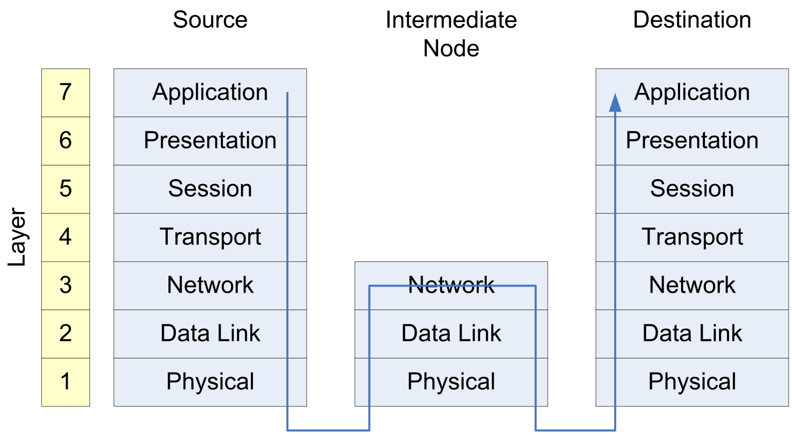


Figure 2.2: OSI reference model: The line depicts the abstract way of a packet.

2.2 Mobile Ad Hoc Networks (MANETs)

Mobile Ad hoc Networks (MANETs) are networks of mobile nodes connected through wireless links. MANETs organize themselves dynamically without using an existing infrastructure or

[1]However, this paradigm is more and more breached by cross-layer approaches. While some authors pin high hopes on cross-layering [12], others highlight the risks that come along with such approaches [13].

centralised administration. As mobile nodes may move arbitrarily, the network topology may change rapidly and unpredictably [14]. Thus, a more precise term for MANETs would be "wireless mobile multihop ad hoc networks".

Compared with traditional wired or wireless networks, MANETs eliminate the need for central infrastructure or administration. They are able to adapt more rapidly to changes in the network or its environment. However, MANETs typically are challenged to cope with a high variation of the capabilities and quality if the involved links and nodes. Furthermore, MANETs often are subject to energy restrictions.

The traditional concept of MANETs bases on the assumption of homogeneous nodes that provide the same functionality and possess similar capacities. As this premise does not hold for many applications – in particular not for Internet access services – the concept of hybrid Wireless Mesh Networks (WMNs) was developed (see later in Section 2.3).

2.2.1 Applications

As tragically happens with many technologies, military applications stand at the beginning of MANET research. The first wireless ad hoc networking applications can be tracked back to DARPA projects in the early 1970s [14]. Yet, in the last few years, civil applications have gained momentum. The advent of new technologies such as Bluetooth or IEEE 802.11 have greatly pushed the development of MANETs in the academical and business sector. One focus of the research was soon laid on sensor networks. However, the conceptual limitedness to homogeneous nodes restrained the success of MANETs.

2.3 Wireless Mesh Networks (WMNs)

Wireless Mesh Networks (WMNs) are a concept related to MANETs. Yet, WMNs specifically are designed to provide Internet access services. In addition, they are characterized by a three-part hierarchy: (1) *Gateways* (sometimes called Access Points or APs) dispose of wired, and typically broadband Internet access. They are located at a fixed place and have no energy restrictions. Usually, gateways provide multiple wireless interfaces. (2) *Wireless routers* typically are relatively immobile as well and have unlimited, or at least high, energy resources. However, wireless routers do not have direct Internet connectivity. (3) *Mobile nodes* (sometimes called nomadic nodes) are highly mobile wireless devices with restricted energy and relatively low communication bandwidth (see Figure 2.3). The requirements for WMNs are somewhat different from the ones for MANETs: Much more attention has to be turned to scalability issues, reliability and throughput quality. Furthermore, WMNs should be able to handle a diversity of devices with entirely different capacities.

WMNs are a fairly young field of research. Publications on this topic are mushrooming only since about 2004. Good surveys on Wireless Mesh Networks have been done by Akyildiz et al. [2; 15] and Bruno et al. [16]. A number of academical research projects are being conducted as well as some first industrial applications (see [2], p. 475ff).

2.4 Routing

Routing protocols are the mechanisms by which network devices collect, maintain, and disseminate information about paths to various destinations in a network. For the actual relaying of incoming network packets e.g. based on routing tables, the term *forwarding* is used [17]. The routing protocol together with the forwarding mechanism is called *routing*.

Routing algorithms can be categorised by various criteria. In *local* routing algorithms, nodes base their decisions on information that is available locally. In *global* routing algorithms, nodes need knowledge about the global state of a network. *Proactive* algorithms constantly maintain information about all possible paths whereas *reactive* routing algorithms might need to gather information before they can forward a packet to a destination they do not know. In *source*

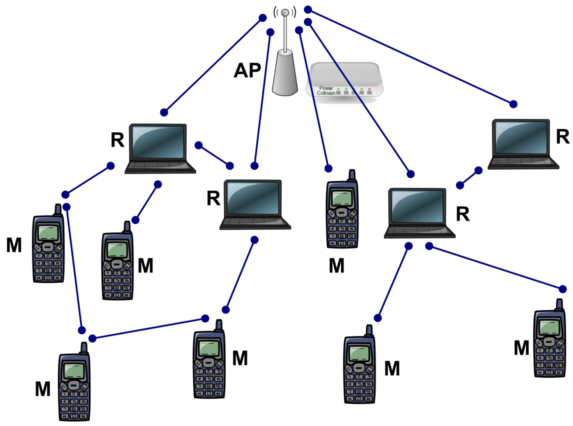


Figure 2.3: Wireless Mesh Networks with one Access Point. AP = Access Point/Gateway, R = Wireless Router, M = Mobile Node

routing, the entire path of a packet is calculated by the source of a packet and stored in the header of the packet. In *hop-by-hop* routing, each node takes the forwarding decision for the incoming packets. Usually, this is done by looking up a table which contains the next hop for all possible destinations. This table is filled and maintained by the routing algorithm [9].

Metrics play a central part in all routing algorithms. They determine the cost that is assigned for a packet travelling over a certain link. Hence, metrics are the basis on which routing decisions are taken.

2.4.1 Routing in MANETs

A large number of routing algorithms for MANETs have been proposed. As this already has been done in various other publications [18; 19; 20], we will abstain from giving an extensive list of proposed protocols. Nevertheless, we will give a short overview on one such protocol on the next page. MANET routing algorithms can be classified according to different criteria[2]:

Routing Information Acquisition Mechanism: In *proactive* (or *table-driven*) routing protocols, each node ceaselessly maintains knowledge on the whole network topology. Usually, this knowledge is stored in routing tables. The most prominent example for proactive routing protocols is DSDV [22]. In *reactive* (or *on-demand*) protocols, nodes obtain the necessary path only when it is required, by using a connection establishment process. Examples for reactive protocols are AODV [23] and DSR [24]. *Hybrid* routing protocols combine the two features: A table is maintained for nearby nodes, for nodes that are located farther away, an on-demand approach is used.

Temporal Information: Routing algorithms can be distinguished between protocols that use only information which was gathered in the past and protocols that attempt to predict the future state of the network.

Routing Topology: Some routing algorithms build up a logically *hierarchical* routing topology while others use a flat scheme. The latter assumes the presence of a globally unique addressing mechanism.

Multipath Algorithms: Most routing protocols specify one unique path per source-destination pair. *Multipath* algorithms establish paths in parallel in order to gain more bandwidth or improve the reliability of a connection.

[2] This categorisation partly bases on the book of Murthy and Manoj [21]. There, a classification of just under twenty routing protocols can be found.

An Example: AODV

The Ad hoc On Demand Distance Vector (AODV) routing algorithm [23] is, together with DSDV and DSR, one of the most popular routing protocol for MANETs in the academical world. It is a classical reactive algorithm. When a node wants to communicate to a node to which it does not already know a route, it broadcasts route request (RREQ) packets across the network. Nodes that receive this messages register their sender in the routing table and forward the message to their neighbours, if they do not know a path to the requested destination. Otherwise, if a node knows a route to the destination of the packet, it answers with a unicast route reply (RREP) message. Sequence numbers are used to guarantee that only up-to-date information is exchanged. Intermediate nodes that relay the RREP message can update their routing table with the route to the destination. After the RREP message arrived at the source node, the sender is able to transmit packets to the requested destination. (These mechanisms are depicted in Figure 2.4.)

As long as packets are sent along the route, it will continue to be maintained. When a source node quits to communicate to the destination, the entries in the routing tables of the intermediate nodes eventually will expire and be deleted. If a link is broken actively, a route error message (RERR) is issued, so that the source immediately can re-initiate the route discovery process.

2.4.2 Routing in WMNs

Currently, most Wireless Mesh Routing protocols are derivatives and adaptations from MANET protocols. One type of protocols that is specifically laid out for WMNs are Field-Based Routing (FBR) protocols. We will focus on this topic in Chapter 5.

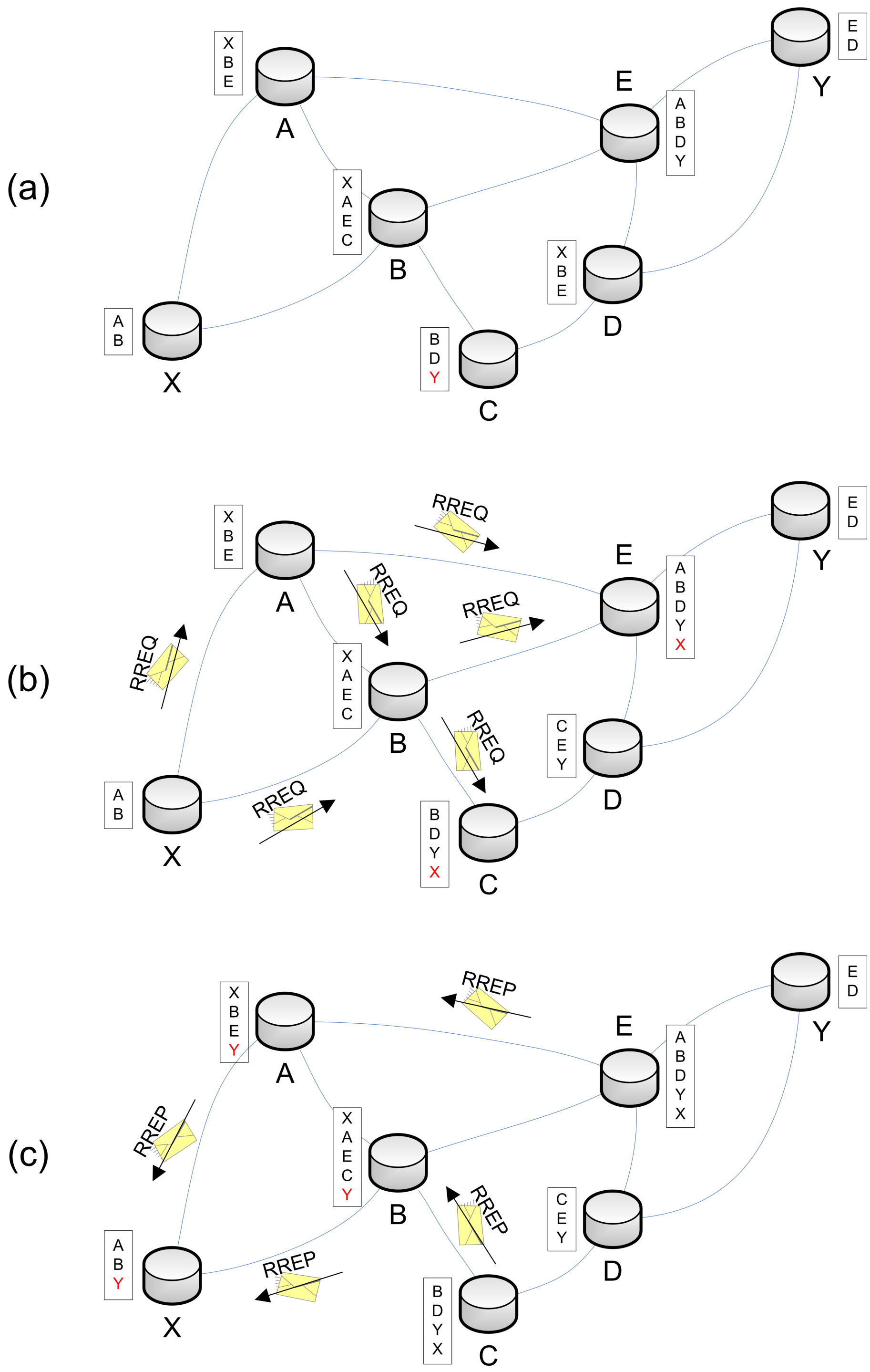


Figure 2.4: The AODV routing protocol. (a) Initial situation: Node X wants to communicate with node Y. All nodes only have information about their direct neighbours, except Node C that already knows a path to Y. (b) X sends out route requests (RREQ) that flood the network until they find a node that knows Y. Intermediate nodes register the sender of the request. (c) Routing replies are sent to X by C and by the neighbours of Y. X will choose the path $X \rightarrow A \rightarrow E \rightarrow Y$ because it contains less hops than the path via C.

Chapter 3

Routing Metrics

Routing metrics are an integral part of every routing algorithm. This chapter is dedicated to obtaining a thorough insight into this topic. In order to examine different metrics, we need a theoretical framework to which we can resort.

In this Chapter, we describe the principles of routing metrics in an informal manner first. In the second Section, we give a short synopsis on the history of metrics. Sections three and four of this chapter contain two of the central parts of this thesis: A mathematical definition of routing metrics and a novel taxonomy which allows to classify routing metrics. In the last section, requirements for good metrics for Wireless Mesh Networks are enumerated.

3.1 The Purpose of Routing Metrics

Basically, routing metrics provide a sense of "distance" between two network nodes. Or, with other words, they describe the "cost" of sending a packet over a network link. Usually, this "distance" does not correspond to the physical distance of two nodes, but the metric depends on the "quality of the link", on the available resources of the involved nodes, or – even more simple – is "one" per hop. In the course of this chapter, we will clarify these rather imprecise terms.

Routing protocols use metrics to determine the "shortest" (or "cheapest") path between two given nodes. Usually, this problem is modelled using graph theory. In this concept, the links correspond to the edges of the graph with the metric defining the weight of the edges (see Figure 3.1). Some routing protocols need metrics to determine which nodes belong to the "neighbourhood" of a station or to assess different paths against each other. For more information on routing and routing protocols, please refer to Section 2.4.

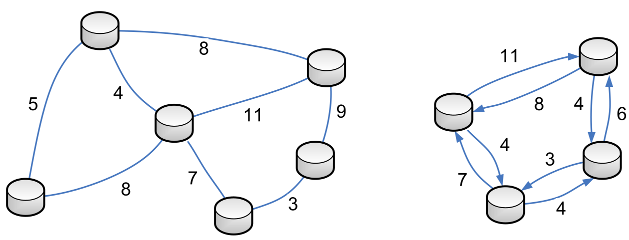


Figure 3.1: Routing Metrics: Example of a bi-directional (left) and a uni-directional (right) metric.

3.1.1 Routing Metrics vs. Performance Metrics

Although there can be certain similarities, routing metrics should not be confused with *performance metrics*. While routing metrics are part of the routing mechanism, performance metrics have a completely different purpose: Performance metrics are used to compare quantitatively different routing algorithms – or different routing metrics. Furthermore, performance metrics are utilised to measure to which extent a chosen optimisation goal is met. Some performance metrics can be found e.g. in [25]. IETF RFC 2330 [26] describes a framework for Internet performance metrics. [27] and [17] give comprehensive lists of implementations for performance metric measurement.

Whenever we use the term "metric" without further clarification in this thesis, routing metrics are meant.

3.2 History

When the first routing protocols were designed in late 1960s and 1970s, metrics were an important issue. Different strategies were proposed, among them the hop count metric or a manual configuration for each network link. Soon, these concepts showed to be too static to be efficient. A number of dynamic metrics were implemented and released to practical use, often basing on the delay of the packets on a link. However, these approaches showed to provoke instable, oscillating network behaviour (see Section 4.1.1) which almost lead to a complete breakdown of large parts of the Internet [28; 29].

With the latest version of the *Revised ARPANET Routing Metric* [8], however, an efficient and highly suitable solution for wired networks was introduced in 1989. It based on sophisticated analysis of the queue lengths in the network routers. As it worked so well, there was no need to research this topic further for more than a decade.

3.2.1 Standard Routing Protocols

Most standard routing protocols do not define metrics explicitly, but leave this task to the implementation developers or even the network administrators. The Routing Information Protocol (RIP) first appeared in 1981 in the "Xerox System Integration Standard – Internet Transport Protocols". Parts of it soon were adopted for ARPANET. A standard for RIP was introduced in IETF RFC 1058 [30]. The only property which the protocol specifies is that all metrics have to have an additive concatenation operator (see Section 3.3.4). Concretely, RIP gives the following instruction for metrics:

> In simple networks, it is common to use a metric that simply counts how many gateways a message must go through. In more complex networks, a metric is chosen to represent the total amount of delay that the message suffers, the cost of sending it, or some other quantity which may be minimized. The main requirement is that it must be possible to represent the metric as a sum of "costs" for individual hops.

The Open Shortest Path First (OSPF) protocol, which is defined in IETF RFC 1247 [31], does define metrics slightly more precisely. It demands that "the cost of a route is described by a single dimensionless metric". Furthermore, OSPF states that "there may be a separate cost for each IP Type of Service" and that the metric value of a link always must be greater than zero. Especially the first statement is interesting: Metrics that depend on the type of traffic are a simple form of multi-dimensional metrics.

The Border Gateway Protocol 4 is described in IETF RFC 1771 [32]. It does not give any precise rules for metrics as well. While it gives some flexibility for metrics inside Autonomous Systems (AS), hop count is used for inter-AS traffic, usually [33].

3.2.2 Recent Developments

With the advent of Quality of Service (QoS) routing protocols, the need for more elaborate metrics was realised (see [9; 17]). The properties of MANETs and Wireless Mesh Networks increase the exigence for more flexible and sophisticated routing metrics even more (see Akyildiz et al. [2]).

3.3 Formal Definition

In this section, we analyse the formal aspects of routing metrics. In literature, we did not find a satisfying and concise mathematical definition of metrics. In the following, we cite a graph-theoretical definition first; then we show which mathematical properties routing metrics do NOT fulfil. At the end of this section, we present our own mathematical definition of routing metrics.

3.3.1 A Graph Theory Approach

In [34], Faragó proposes a graph theoretical definition of routing metrics:

> $\mathcal{F}$**-Metric**: Let P be a path. Given a family $\mathcal{F}$ of sets of edges in the graph, the $\mathcal{F}$-measure (of $\mathcal{F}$-metric) of the path P is defined as the number of edge sets $H \in \mathcal{F}$ that have non-empty intersection with P. The sets are counted with multiplicity. Formally, the $\mathcal{F}$-measure of P, denoted by $\mathcal{F}(P)$, is defined as
>
> $$\mathcal{F}(P) = \sum_{H \cap P \neq \emptyset, H \in \mathcal{F}} m(H)$$
>
> where $m(H)$ is the multiplicity of the set H in $\mathcal{F}$. A path with minimum $\mathcal{F}$-metric is called an $\mathcal{F}$-shortest path.

Faragó proves that every routing metric can be represented by an $\mathcal{F}$-metric. He also shows that his definition can fruitfully be applied to numerous different path finding problems. In his article, Faragó develops the concept of the $\mathcal{F}$-metric only for integer multiplicities as edge weights, although he claims that his definition can be widened easily for real weights. Furthermore, the $\mathcal{F}$-metric is proposed only for additive metrics. An adaptation to multiplicative and concave (see Section 3.3.4) metrics has not been thoroughly investigated yet, but it does appear to be quite straightforward.

As useful Faragó's definition is to explore the graph theoretical aspects of routing protocols, it does not explain how the weight of the edges (or their multiplicity, to be more precise) is generated. Faragó assumes from the beginning that every routing metric can be modelled as any function $h(P)$ that assigns a positive integer value to every path P. We want to look at this part of the problem a bit more profoundly.

3.3.2 Metrics in Mathematics

Before we start to devise a definition for routing metrics, we will have a look on how metrics are defined in mathematics: A metric is a mapping $d : X \times X \to \mathbb{R}$, where X is an arbitrary set, with d such that the following axioms are met [35]:

1. $d(x, y) \geqslant 0$ (non-negativity)
2. $d(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, y) \leqslant d(x, z) + d(z, y)$ (triangle inequality)

For our purpose, the elements of set X would be the network nodes and the mapping would be an arbitrary function that represents the "distance" of two nodes as a scalar value.

Two basic problems arise, when we adapt these rules to routing metrics. First, physical links often are asymmetric. An obvious example is the bandwidth of an ADSL connection. This contradicts axiom 3 of the definition above. Secondly, the triangle inequality – axiom 4 – is not valid. The triangle inequality states that there is never a shorter way than the direct one. However, in computer networks, it happens regularly that an indirect path is better suited than a direct one (see example in Figure 3.2).

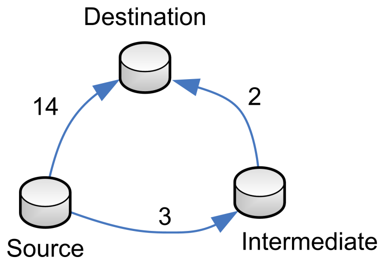


Figure 3.2: Example for the invalid Triangle Equation: The path over the intermediate node is clearly better than the direct link.

In order to circumvent the triangle equation problem, one theoretical loophole is to conceive the space of metrics not as Euclidean but as Riemannian space [36]. This would imply that the metric between two nodes is not defined by the result of the direct mapping $d(y, x)$, but straightly as the shortest path between the two. It seems obvious that this idea does not entail many possibilities for further investigation: The goal of a metric, finding shortest paths, cannot already be innate in its definition.

Groups, Rings and Fields

Thus, we cannot demand routing metrics to be metrics in the mathematical sense. Just as little, it is possible to require that routing metrics be *groups*. A group $(G, *)$ is a set G together with a binary operation $* : G \times G \to G$, satisfying the following axioms [35]:

1. For all a, b, c in G: $(a * b) * c = a * (b * c)$ (associativity)
2. There is an element e in G such that for all a in G, $e * a = a * e = a$ (neutral element)
3. For all a in G, there is an element b in G such that $a * b = b * a = e$, where e is the neutral element from the previous axiom. (Inverse element)

In our case, the elements of the set would correspond to links and combinations of links and the operation would be the combination of these link set elements. The value of the metric would be generated by an additional mapping $G \to \mathbb{R}$. Two elements of the set would be considered equivalent, if the value of the metric is the same.

In a group, every element of a set must have an inverse element according to axiom 2. If the operation of the group is applied to the element and its inverse element, the result would have to be the neutral element. However, there is not necessarily an inverse element for routing metrics. If e.g. the delay of a link is used as metric, the neutral element obviously is a link with a delay of 0 seconds. No inverse elements – besides links with negative delays – exist that produce the neutral element (i.e. a delay of 0 seconds) when they are combined with arbitrary elements. As negative delays do not exist, the delay routing metric consequentially is not a group. From this follows that the delay routing metric is neither a *ring* nor a *field*, too. As we certainly do not want to exclude any existing metrics from our definition, we cannot demand routing metrics to be groups, rings or fields.

Commutative Monoids

The only group theoretical property we may demand from a routing metric is that it is a *commutative monoid*. A commutative monoid $(M, *)$ is a set M together with a binary operation $*$ such that $M \times M \to M$. A commutative monoid must meet the following requirements [37]:

1. For all a, b, c in M: $(a * b) * c = a * (b * c)$ (associativity)

2. There exists an element e in M, such that for all a in M: $a * e = e * a = a$ (neutral element)
3. For all a, b in M: $(a * b) = (b * a)$ (commutativity)

In the case of routing metrics, the elements of set M consist of links or combinations of links. The binary operation may be additive, multiplicative or concave (see Section 3.3.4). All these operations evidently fulfil the associativity and commutativity requirement and offer a neutral element.

3.3.3 Requirements for a Definition

Before we propose an own definition of routing metrics, we will state which prerequisites we pose to such a definition. Basically, it should meet our intuitive notion of routing metrics and give it a mathematically precise form. Furthermore, it should meet the following requirements:

1. Formalise the mapping of the network connection between two nodes to a scalar value.
2. Provide rules on how parts of the network are concatenated and aggregated.
3. Efficient algorithms to find shortest paths should be available for a metric. Bellman-Ford or Dijkstra's algorithms are examples of such algorithms.
4. Guarantee the absence of routing loops.
5. Not inhibit the convergence of routing algorithms.

3.3.4 Proposed Definition

With these specifications in mind, we propose the following definitions:

> Let G be a set of unidirectional communication links[1] and combinations of these links and let
> $$d : G \to \mathbb{R}_0^+$$
> be a mapping from this set to a scalar value. d is called a *single metric* on G. Furthermore, let $\|$ be an operation that aggregates two elements of G in parallel and let $+$ be another operation which concatenates two elements of G in series. The combination operators are chosen such that $(G, \|)$ as well as $(G, +)$ are commutative monoids. In addition, let $\preceq$ be a totally ordered set on d (see special subsection later).

These definitions fulfil the first two requirements stated above. In order to meet requirements three, four and five, we pose two additional demands to the concatenation operator:

- Monotonicity: Metric d must provide that $d(x) \preceq d(x + y) \quad \forall\ x, y \in G$.[2]
- Isotonicity: Metric d must be *isotonic* i.e. from $d(x) \preceq d(y)$ follows that $d(x + z) \preceq d(y + z)$ and $d(x \parallel z) \preceq d(y \parallel z)$.

Sobrinho [10; 11] proves that monotonicity and isotonicity are sufficient and necessary conditions for the convergence of traditional routing algorithms[3]. If these two properties are not given, no minimum weight algorithm with polynomial complexity can exist [38]. Some routing algorithms, e.g. Field-Based Routing, are not reliant on polynomial complexity.

[1]Paxson [26] defines metrics as property of any network entity, including routers and hosts. We restrain metrics to be properties of links (and combinations of these), because this simplifies further investigation. We can do this without losing any generality, as every property of any network entities can be mapped onto a property of the adjoining links.

[2]There is no similar condition for $\|$. $d(x \parallel y)$ may be greater than $d(x)$ (if the metric is designed to minimise interference) or less than $d(x)$ (e.g. for bandwidth).

[3]At first glance, monotonicity and isotonicity might look as interchangeable. However, when multi-dimensional metrics are used, counter-examples can be found quickly (see [11], p. 54).

Combinations

Usually, routing metric values are measured on a single link between two nodes[4]. However, ultimately we are interested in the metrics of whole paths to gateways in order to compare these paths. Therefore, we assume that there are mathematical operations that describe what happens when links are combined to paths. There are two ways how links can be combined : Either in series (concatenation) or in parallel (aggregation), see Figure 3.3.

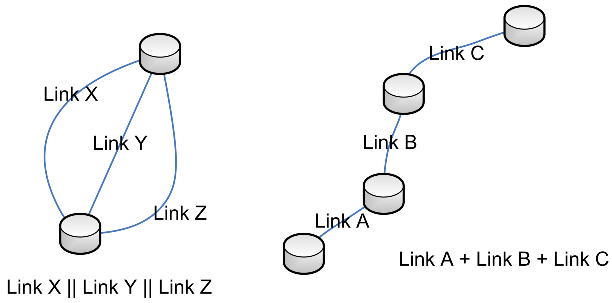


Figure 3.3: Aggregation and concatenation

The concatenation operator is more perspicuous for most metrics. The composition in Appendix A shows which metric requires which operators. We can see, that the addition is the most widely used concatenation operator, sometimes also maximum/minimum and multiplication are found. For many routing algorithms, combining links in parallel does not make sense, as only one link will be chosen at each node. (To be more precise, the aggregation operator is the maximum or minimum function in these cases.) However, the aggregation operator is of high interest when multi-path routing algorithms are applied or the redundancy should be incorporated into a metric (for example in Field-Based Routing algorithms, see Chapter 5).

Ordered Set

Above, we stated that for every metric, a total order must be defined. A total order is characterized by four properties:

- Reflexivity: $d(x) \preceq d(x) \quad \forall\ d(x)$
- Anti-symmetry: If $d(x) \preceq d(y)$ and $d(y) \preceq d(x)$, then $d(x) = d(y)$
- Transitivity: If $d(x) \preceq d(y)$ and $d(y) \preceq d(z)$, then $d(x) \preceq d(z)$
- Totality: $d(x) \preceq d(y)$ or $d(x) \preceq d(y)$ for all $d(x)$ and $d(y)$

In the case of one-dimensional metrics, this order normally will be the ordinary $\leq$ relation for real numbers. When multi-dimensional metrics are used, however, this order can be designed more flexible. For example, it could be made possible that each node or each packet decides individually which dimension (energy, delay, throughput, et cetera) should be weighted more strongly.

3.4 Taxonomy

In the following section, we propose a novel taxonomy for routing metrics. The purpose of this taxonomy is to bring structure into the complex area of routing metrics. It provides a decision-making aid for the choice of appropriate metrics for a given problem.

[4]Performance metrics, contrariwise, often measure end-to-end values. Stephan et al. [39] define metrics which involve more nodes than only a sender and a receiver as *spatial metrics* in their IETF Internet Draft.

Our classification scheme consists of five different aspects under which a metric is investigated: Influence factors, mathematical properties, design goal of the metric, implementation characteristics and finally a qualitative evaluation of the metric. Each of these aspects consists of various sub-classifications.

3.4.1 Factors of Influence

The value of many metrics is governed by various different effects. Often, it is not trivial to identify all these factors. There are two different types of such factors: Environmental factors and network-immanent factors (see Figure 3.4).

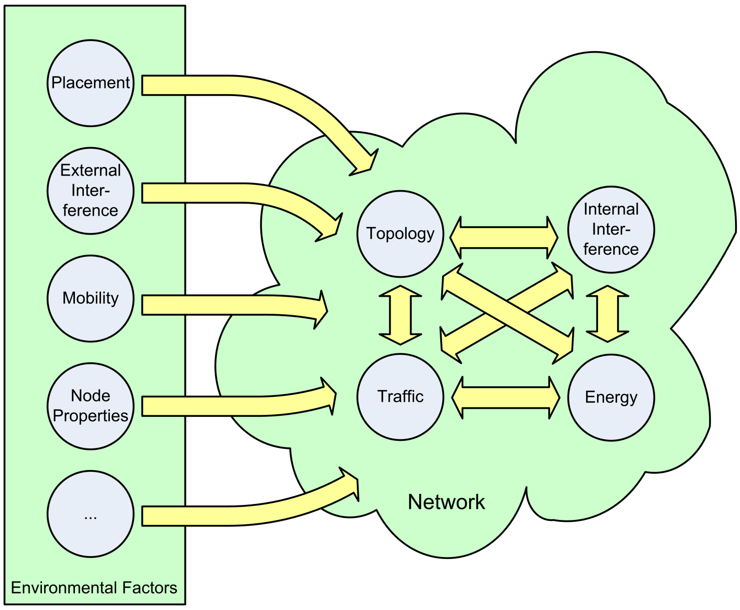


Figure 3.4: Factors of influence: Environmental and network-immanent factors

1. **Environmental Factors** are defined as factors that influence the metric, but are not affected by the network. With other words: Environmental factors are parameters that are not subject to feedback from the network. Examples for environmental factors are the placement and mobility of the nodes, their technical properties or external interference.

2. **Network-Immanent Factors** are defined as factors that depend directly or indirectly from the traffic within a network. Among these factors are congestion, internal interference (inter-flow and intra-flow interference), the consumed energy[5] and e.g. in IEEE 802.11 systems also the topology of the network[6].

In the survey in Chapter 4, the influencing factors of a metric are described using flow diagrams.

[5]Of course, the amount of traffic influences the consumed energy. But topology can have an effect on energy, as well. In systems that adapt transmission energy, one long hops may consume more energy than a series of short hops (see Section 4.5.1).
[6]IEEE 802.11 systems adapt the communication radius according to the present interference. Therefore, the neighbourhood of a node changes in dependence of the competing traffic.

3.4.2 Mathematical Properties

Metrics can be classified into a number of mathematical categories.

1. **Link Combination Operator:** Link metrics are aggregated and concatenated according to certain rules in order to attain the metric of a path (see Section 3.3.4). Each combination operator defines a class. While the concatenation operator is clearly defined for most metrics, the definition of the aggregation operator often is less straightforward.

2. **Dynamic vs. Static:** A metric is dynamic, if the value of the metric changes over time. This is the case for most metrics. Examples for static metrics are the number of interfaces of a node or the maximal energy capacity.

3. **Symmetric vs. Asymmetric:** Let $d_{i,j}$ be the metric value of the link from node i to node j and $d_{j,i}$ the link in the opposite direction. A metric is symmetric if $d_{i,j} = d_{j,i}$ holds for all links and all points in time.

4. **Single-Dimensional vs. Multi-Dimensional Metrics:** Multi-dimensional metrics are not of type real, but they are vectors. Wang and Crowcroft [9] use delay and bandwidth for the design of a two-dimensional metric. Iannone et al. [12] do the same for PLR, interference level and data rate. Sometimes, multi-dimensional metrics are also called *multiple metrics* because they can be decomposed into single-dimensional metrics.

   Wang and Crowcroft [9] prove that finding a shortest path in a network with multi-dimensional metrics is NP-complete, for multiplicative as well as for additive metrics, if it is done in the same fashion as in the single-dimensional case. Therefore, multi-dimensional metrics are not used in practise to the best of our knowledge.

3.4.3 Design Perspective

This aspect describes the design goals of a metric. We divide this aspect into two sub-classification dimensions:

1. **Target Platform:** What system was the metric originally designed for? Most metrics are laid out for standard Internet applications, QoS Internet applications, MANETs or WMNs.

2. **Optimisation Goal:** What is the optimisation goal of the metric? This can be e.g. “minimise energy”, “maximise node lifetime”, “maximise average end-to-end throughput”, “guarantee a minimal bandwidth” or “minimise delay”.

   We point out that a metric which is dedicated to a certain goal must not necessarily be able to fit this goal. For example, the delay metric originally was proposed to minimise delay. In many cases, however, it produces an end-to-end delay which is worse than the hop count metric. Please refer to Section 3.4.5 on evaluation criteria to read more about performance evaluation of metrics.

3.4.4 Implementation Characteristics

The implementation characteristics specify the properties of a metric which are related to practical implementation problems.

1. **Combined Metrics** are metrics that are mathematically combined from other metrics and do not base on own measurements. A typical example is the ETT metric (Section 4.1.10). This metric is calculated by multiplying bandwidth with the ETX metric (Section 4.1.8).

2. **Layer:** This property describes which OSI layer provides the required information. Traditionally, routing metrics are calculated only from information available directly on the network layer. Today, however, many researchers acknowledge the need for cross-layer approaches for designing routing metrics for MANETs and WMNs (see Section 2.1.2) .

3. **Analytically vs Empirically-Defined Metrics:** Some network quantities – for example the available bandwidth – are very well suited for analytical description, but are rather complex to measure. Metrics that are defined using such properties are called *analytically-specified*. Other variables can be measured conveniently, but it is not trivial to interpret them analytically. This kind of metrics is termed *empirically specified* [26].

4. **Information Acquiring Method:** There are various ways how metrics acquire the information they need. We define the following classes:

   Node-related: Information for the metric is acquired from a node without high effort. This may be fixed values like the number of interfaces of a node, configured values like the financial communication costs or variable values like the length of input and output queues.

   Passive Monitoring: Information for the metric is gathered by observing the traffic coming in and going out of a node. In combination with other measurements, this can be used e.g. to estimate the available bandwidth.

   Piggy-back Probing: Measurements are done by including probing information into regular traffic or routing protocol packets without creating own packets for metric measurement. This is a common method to measure delay.

   Active Probing: For this technique, special packets are generated to measure the properties of a link. For more information on the problems of this technique, please refer to Section 4.1.1.

3.4.5 Evaluation

Qualitative and quantitative evaluation and comparison of metrics cannot be done in a simple manner. The assessment of a metric depends on various factors, among them are the applied performance goals, the utilized routing protocol and the postulated scenario in terms of placement, mobility and communication capabilities of the nodes. Thus, the evaluation criteria and the resulting grading of metrics are subject to assumptions of average cases which are not perfectly objective. A truly satisfying evaluation would only be possible by deploying all metrics in various application scenarios, embedded in various routing protocols, implemented in various simulators and even in hardware. Of course, this would be beyond the scope of this master's thesis. Nevertheless, it is possible to make qualitative statements on metrics using general assumptions and earlier studies on this topic. Hence, we tried to rate the metrics as thoroughly as possible and predict their performance as accurate as we could basing on all information and related studies that we disposed of.

In order to provide developers with a concise and fast decision help, we condensed the central evaluation criteria of metrics in six categories, each having four quality classes. Three categories rate the metric with respect to the performance indicators, two categories evaluate the ease of implementation (complexity and measurability) and one category specifies how widely used a metric is.

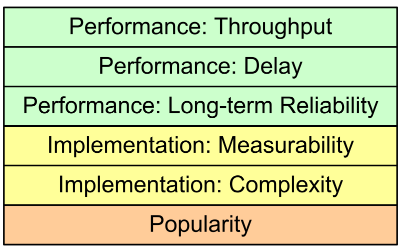


Figure 3.5: Six Evaluation Criteria

I-III Three Performance Criteria

Metrics are evaluated along three performance criteria that we identify as fundamental: Throughput, delay and long-term reliability. These criteria, first, are orthogonal i.e. independent from each other and, secondly, they describe the properties of a network which are of interest for the user of a communication network.

For the evaluation, we always suppose an optimal implementation which has been tuned perfectly. The four quality classes are defined as: (++) has a clearly positive effect, (+) tends to have a positive effect in certain situations, (0) neither positive nor negative effect can be anticipated and (-) negative effects will probably prevail.

IV Implementation – Measurability

The *measurement criteria* consists of five sub-criteria. If none of the sub-criteria applies, the metric is rated with (++), for one sub-criteria the rating is (+), for two is is (0). If more than two sub-criteria apply, the metric is rated with (-). The five sub-criteria are:

- Oscillation Stability: Is the metric vulnerable to self-interfering effects (see Section 4.1.1)?
- Cross-Layering: Does the metric need access to information provided by other layers?
- Overhead: Does the metric induce extensive measurement overhead? We define that a measurement overhead is extensive if more than one message exchange is performed per measurement. For example, a ping message is not considered extensive, whereas PATHCHAR or packet pair algorithms are.
- Freshness: Is there an inherent delay between the reaction of the metric to changes of the measured value? Are there transient effects?
- Additional devices: Is data from an additional data source (e.g. a geographical positioning system) needed?

V Implementation – Complexity

The better the *complexity criteria*, the less are the computation and memory requirements that are implied by the metric. In fact, there are almost no metrics with high processing demands[7]. We assume elevated memory requirements whenever a series of measurements has to be stored in order to compute the current value of a metric.

VI Popularity

The *popularity criteria* indicates how widely used a metric is and on how much experience one can build. This criteria is composed of three sub-criteria:

- Is a metric well known i.e. is it cited often in literature?
- Is it in practical use outside the academical world?
- Is it well studied in literature?

3.4.6 Summary

In the following table, the aspects and categories of our taxonomy are summarised. We give an example for each sub-classification in italics. We will use this table format throughout the survey in Chapter 4.

[7] Theoretically, multidimensional metrics are np-complete and, therefore, require high processing efforts. However, we do not know of any practical implementations of multidimensional metrics.

Factors of Influence	see Figure
Mathematical Structure	Concatenation operator: *Addition* Aggregation operator: *Minimum* Other mathematical properties: *Dynamic, asymmetric, one-dimensional*
Design Goals	Target Platform: *Internet, QoS protocols* Optimisation goal: *Minimise average end-to-end delay*
Implement. Characteristics	Layer: *Measurement possible on L2-L7* Information acquiring method: *Monitoring, piggy-back or active probing possible* Other properties: *Empirically and analytically defined, not combined*
Evaluation	Performance: Throughput = + Performance: Delay = ++ Performance: Reliability = 0 Implementation: Measurability = 0 Implementation: Complexity = 0 Popularity = ++

3.5 Requirements for WMN metrics

- A metric must comply with a number of mathematical properties (see Section 3.3).
- Principally, a metric should be computable in real world implementations [40]. Many seductive metrics were proposed that based on good ideas but were not realisable in practise. Paxson [26] uses the term *well-defined* for metrics that can be implemented and the word *ill-defined* for the opposite.
- A metric should be computable in distributed manner without global knowledge. Information gathering from many other nodes implicates additional demand for network resources, which should be avoided [40].
- A metric should provide up-to-date information. There should not be a considerable lag between the change of the measured variable and its representation in the metric value.
- A metric should be a good indicator of network performance. Either the metric should measure a performance variable (throughput, delay or long-term reliability, see Section 3.4.5) directly or it should measure a value that permits to predict a performance variable [40; 28].
- A metric should lead to stable routes. As Internet traffic shows a highly fluctuating characteristic, traffic-based metrics may produce large variations. Especially in proactive routing protocols, rapidly changing link weights can cause high amounts of undesired routing overhead. Yang et al. [38; 41] claim for this reason even to abstain from load-based metrics completely. Although we acknowledge the inherent risks of load-based metrics, we consider this view too rigid. With appropriate techniques, the variation of such metrics can be filtered to suitable proportions.
- A metric should not be susceptible to oscillation effects (cf. Section 4.1.1) [28].
- A metric should be feasible to compute with the resources of nodes. Some nodes provide only limited processing, memory and energy capacities. Also these nodes should be able to use the metric. [40]
- A metric should be independent from any specific protocol [40]. Whether this should include a reluctance against cross-layer concepts is subject of a debate (see Section 2.1.2).

Chapter 4

Survey

This Chapter gives an extensive overview of routing metrics. In the following sections, we will explain, analyse and classify metrics according to the taxonomy presented in Section 3.4. The metrics are arranged in five sections, based on their functional usage: Traffic-based, radio-related, topology-based, mobility-based and geographical, and finally energy-related metrics.

4.1 Traffic-Based Metrics

When using wireless networks for Internet or other communication applications, many design goals are related to network traffic: High throughput, small delay, or limited variance of the connection quality. It is an obvious idea to incorporate traffic indicators into routing algorithms. Many metrics were proposed that attempt to do this. However, measuring traffic variables is a sophisticated challenge and the risk of obtaining unstable network behaviour is high.

Some implementation problems are common to all traffic-based metrics. In the first section, some general considerations regarding the measurement of traffic are presented. In the subsequent sections, single traffic metrics, their strengths, their limitations and their pitfalls will be discussed.

4.1.1 Traffic Measurement

Generating Probing Packets

Many metrics require some kind of probing messages in order to measure the quality of a link or a path. This technique entails different challenges.

Several authors point out that probing does not always reflect what a metric wants to measure. For example, routing traffic or ICMP packets, which both are used regularly for probing, often have higher priority in routing queues than normal traffic. These so-called *out-of-band* measurements may not reflect the network condition normal traffic is subject to. On the other hand, if the probing packets are interlaced with the regular traffic (so-called *intrusive* or *in-band* measurement), the probes themselves influence the amount of traffic. Ferguson and Huston [17] compare this effect with the Heisenberg Uncertainty Principle. Lundgren et al. [42] and later Zhang et al. [43] observed that the different properties of unicast and broadcast communication in IEEE 802.11 systems may lead to similar effects: Probes that are sent using the broadcast mechanism will report neighbours that are not reachable using unicast communication. Both papers call this phenomenon the *grey zone* problem.

Another problem are the intervals at which probing messages are generated. An overly small interval produces unnecessarily large amount of overhead, while the risk of missing important information increases with greater intervals. RFC 3432 [44] is dedicated to such considerations.

Oscillation Stability

Certain metrics tend to provoke oscillating traffic patterns. This effect can be explained as follows: Once a link is recognized as good, it attracts a lot of traffic. Consequently, the link becomes jammed and it is assigned with a worse metric value. As traffic starts to route around this link, its metric value increases again and the effects starts anew. This phenomenon, called *self-interference*, was observed already in the first Internet applications [8].

The level of oscillation stability may not only depend on the principal characteristic of a metric but also on the implementation details. In many cases, the risk of self-interference can be reduced by smoothing out short-term peaks in the value of a metric (see next section). However, this method conflicts with the goal that metrics should represent the current state of the network. Furthermore, the choice of parameters for smoothing is rather delicate.

Filtering Measurements

Some network parameters that are central for certain metrics are subject to high variation, e.g. delay or link loss ratio. Usually, it is desired that short-term variations do not influence the value of a metric, as this could cause disproportionate adaptations of the metric and would increase the risk of self-interference. Therefore, metric measurements should be filtered over time. Floyd et al. [45] propose three methods how this can be done by applying weighted averages of the measurements:

1. **Dynamic History Window:** An average is computed over a number of previous measurements. The size of the measurement window depends on the current transmission rate.

2. **Fixed History Interval:** An average is computed over a fixed number of previous measurements.

3. **Exponential Weighting Moving Average (EWMA):** Measurements are weighted with exponentially less the older they are. This can be stated with the following formula [33]. The current value is recalculated whenever a new sample of the delay measurements arrives.

   $$d_{new} = \alpha * d_{old} + (1 - \alpha) * d_{sample}$$

   with $\alpha \in [0, 1]$ being the weighting factor, d_{sample} being the new sample, d_{old} the current metric value and d_{new} the newly calculated value. In fact, this calculation implements a discrete low pass filter [46].

Floyd et al. show that the Average Loss Interval method results in smoother TCP traffic in wired networks. Anderson et al. [33] confirm this finding for overlay networks. However, it is not clear whether this results can be translated to MANETs one-to-one (see e.g. Section 4.1.6 on packet loss metrics).

4.1.2 Delay

Name	Delay, latency, transmission time
Factors of Influence	see Figure 4.1
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined (see following paragraphs) Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: QoS protocols, ARPANET Optimisation goal: Minimise average end-to-end delay
Implement. Characteristics	Layer: Measurement possible on L2-L7 Information acquiring method: Monitoring, piggy-back or active probing possible Empirically and analytically defined
Evaluation	Performance: Throughput = + Performance: Delay = ++ Performance: Reliability = 0 Implementation: Measurability = 0 Implementation: Complexity = 0 Popularity = ++
Adaptations	Expected Delay, Round-Trip Time
Main References	Draves et al. [47], Khanna et al. [8]

Figure 4.1: Dependencies of the delay metric

The delay metric measures the time to send and receive a unicast packet from one node to another. There are three principal approaches on which this measurement may base: Active probing, piggy-back normal traffic or piggy-back routing information exchange messages.

Delay can be subdivided into six different phases, see Figure 4.2. The overall delay is composed from queuing delays (Q_S and Q_R) , processing delays (P_S and P_R), transmission delay T and propagation delay P. Given a bandwidth ρ, the transmission delay for a packet of b bits equals b/ρ. Thus, the overall delay D follows the equation

$$D = P_S + Q_S + P + \frac{b}{\rho} + Q_R + P_R$$

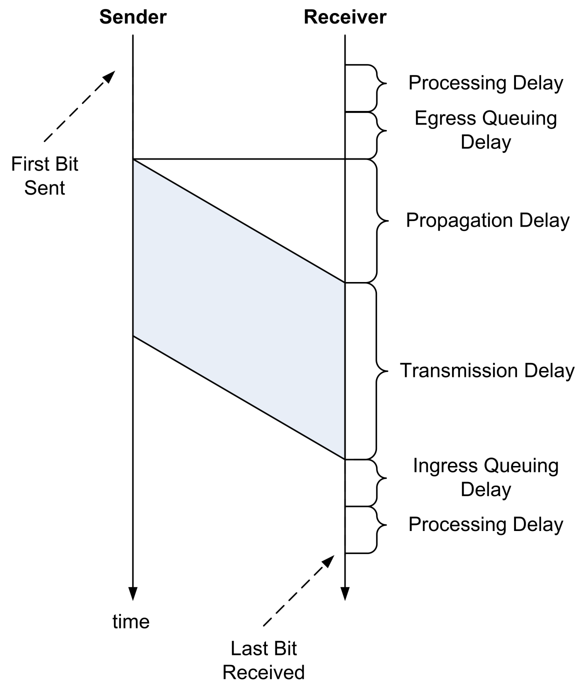


Figure 4.2: Terminology of delay phases

Moving Averages

The delay of a link normally is subject to considerable variance. For this reason, most protocols do not use only the currently measured value to determine the metric for a link, but they utilize a weighted average. Usually, an Exponential Weighting Moving Average (EWMA) is applied (see Section 4.1.1). For example, Gupta and Kumar [48] propose to use EWMA for their MANET routing algorithm, calling it *Expected Delay.* To the best of our knowledge, there was no evaluation published on how the EWMA parameter is set appropriately in MANETs or WMNs (see Section 7.3, Future Work).

Unidirectional Delay

When a delay is measured only in one way, it is necessary to synchronise the internal clocks of the two involved nodes. Therefore, most protocols do not use unidirectional delays, but round-trip times. Nevertheless, IETF RFC 2679 [49] describes a one-way delay metric for IP traffic. Recently, Shalunov et al. [50] drafted another unidirectional delay measurement protocol for the Internet.

The Round-Trip Time (RTT) quantifies the bidirectional delay of a link: A probe carrying a timestamp is sent to a neighbouring node. This node returns the probe immediately. Thus, the time is measured for a packet to travel to the neighbouring node and back again.

Different factors have to be taken into account when the advantages and disadvantages of RTT and unidirectional delays are weighted against each other. In a scenario with many asymmetric links and a lot of unidirectional traffic, a unidirectional delay metric will represent the capacity of links better than an RTT metric. On the other hand, time synchronisation is not required when RTT is used, as only the timestamp of the probing node needs to be evaluated.

Round-Trip Time measurements often are implemented using ICMP echo-request (ping) messages or information from TCP[1]. A definition for a more sophisticated protocol is given in IETF RFC 2681 [52]. For a more general overview of probing, please refer to Section 4.1.1.

Adya et al. propose to use RTT for their "Multi-Radio Unification Protocol" [53].

Combination of Links

Unidirectional delay and RTT can be measured either for each link separately or for an entire path. The sum of the delay of concatenated single links will result in the costs of the entire path. When a multi-path routing algorithm is used, multiple probes travelling on different paths will report different delays. There is a number of ways, how this values can be aggregated: Minimal delay, maximal delay, medium delay et cetera.

Performance and Instabilities

The deployment of unidirectional delay metric as well as of RTT can lead to unstable routes or oscillating load allocation when links are heavily utilised: Links with low load are given an attractive metric value and subsequently are overflown with traffic, while other links starve and only receive a better metric value after a while. For example, the first versions of ARPANET suffered from this problem (see Khanna and Zinky [8]). Draves et al. [47] call this phenomenon *self-interference* (see Section 4.1.1. Draves [47] measured in a test-bed experiment that RTT performed 3 to 6 times worse than Minimal Hop Count, Packet Pair or ETX metrics in terms of TCP throughput. He also attributes this to self-interference problems.

In order to reduce this issue, the parameters of this metric (e.g. the EWMA weighting factor, see Section 4.1.1) must be carefully adjusted. Further, Draves et al. propose to deploy a Packet Pair Delay metric (see Section 4.1.5) to decrease the impact of this phenomenon.

[1] IETF RFC 1323 [51] gives some important hints on this technique. For example, retransmitted packets should not be taken into account for RTT measurement, as these packets will produce a distorting effect.

4.1.3 Delay Variation

Name	Delay variation, jitter
Factors of Influence	see Figure 4.3
Mathematical Structure	Concatenation operator: Addition (with restrictions, see following paragraphs) Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: QoS protocols Optimisation goal: Minimise delay of streams
Implement. Characteristics	Layer: Measurement possible on L2-L4 Information acquiring method: Monitoring or piggy-back probing possible Empirically and analytically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = + Implementation: Complexity = 0 Popularity = 0
Adaptations	
Main References	IETF RFC 3393 [54]

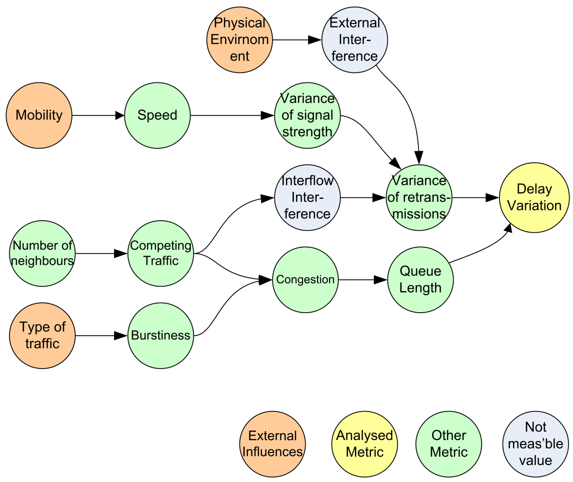


Figure 4.3: Dependencies of delay variation metric

Delay variation is the variation of a delay metric with respect to some reference value e.g. average delay or minimum delay (see IETF RFC 3393 [54]). In this section, we will assume that classical statistical variance[2] is used to describe the *delay variation*. Often – but not always – people utilize the term *jitter* interchangeably. As jitter is used also with slightly different meaning in other domains, we will avoid this term in this thesis.

The delay variation metric is of special interest for applications which are subject to some real-time restrictions, for example interactive voice or video communication. In such applications, the

[2]The variance $Var(X)$ is defined as the expected value $E((X-\mu)^2)$ with μ being a mean value and X a series of measurements.

delay variation determines – among other things – the size of the buffers which are necessary to smooth the audio and/or video stream.

Like the delay metric, the delay variation can be measured either with active probing or piggyback on normal or routing traffic. Unlike plain delay, delay variation is not reliant on time synchronisation, even for unidirectional measurements [54].

Due to the self-similarity ("burstiness") property of Internet traffic, delay variation will only converge in the long-term, if it converges at all. For more information on this topic, please refer to [55].

Combination of Links

When the delay variation is measured for a single-path routing algorithm on an entire path, the value is measured directly. But how do we calculate the metric for a path, if the Delay Variance is measured on each link separately? If the delay values of the concatenated links were uncorrelated, the sum of the single link delay variance measurements would represent the delay variance or the entire path. However, we claim that this is not the case. If one router is contented, the chances are high that two consecutive links will both report higher delays. Likewise, both delays will decrease simultaneously, as soon as traffic volumes become smaller.

When two links with the delay series D_1 and D_2 are concatenated, the resulting variance can be expressed with the following equation:

$$Var(D_1 + D_2) = Var(D_1) + Var(D_2) + 2 * Cov(D_1, D_2)$$

with $Cov(D_1, D_2)$ being the covariance of the delays of the two links. The calculations becomes more complicated when three or more links are involved. To the best of our knowledge, all systems that use delay variation measure the metric on the entire path and not on each link separately. Therefore, covariance does not play any role in these metrics. If we envision a system without the opportunity to measure delay variation on the complete path, but for each link separately, it is not clear how the covariance could be calculated.

4.1.4 Queue Length

Name	Queue length
Factors of Influence	see Figure 4.4
Mathematical Structure	Concatenation operator: Addition or minimum (see following paragraphs) Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: Internet, QoS Optimisation goal: Minimise delay, load-balancing
Implement. Characteristics	Layer: Measurement on L3 Information acquiring method: Node-related Empirically defined
Evaluation	Performance: Throughput = + Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = 0 Implementation: Complexity = 0 Popularity = +
Adaptations	
Main References	

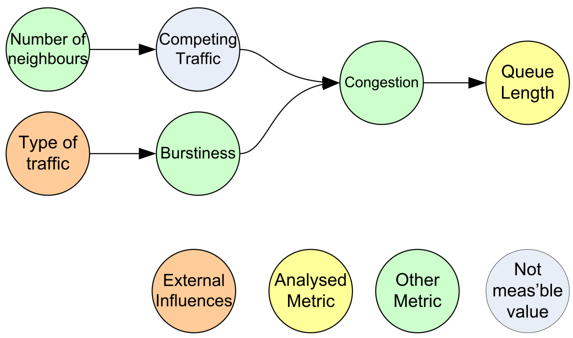


Figure 4.4: Dependencies of queue length metric

Every network node has an ingress and an egress queue. In these queues, incoming and outgoing packets are stored if the interface is not able to forward them immediately. The queue length gives an indication for the current state of the device: If the queue is empty, the device can process more traffic, if the queue is full, the interface is contented and cannot handle more packets.

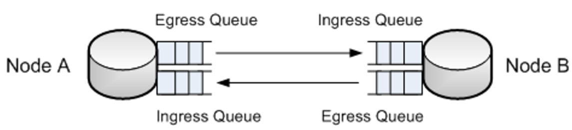


Figure 4.5: Queuing in network nodes

Usually, the queue length is limited to a few packets and routers start to drop packets when the queue is full. However, some other strategies were proposed to avoid congestion, most notably Random Early Detection [56].

Queue lengths are properties of network nodes. Though, we defined metrics to be properties of network links (see Section 3.3.4). There are three straightforward ways to map queue length to link metrics:

- Add the sizes of the ingress and egress queues at both ends of the link.
- Use the maximal value of the four queues.
- Use each queue size as one single metric.

Measurement

The most obvious and reliable method to obtain information on the status of remote queues are SNMP request messages. If this is not a viable solution for any reason, the queuing delay in congested nodes can be estimated with Packet Pairing (see Section 4.1.5).

Combination of Links

There are various ways how a path metric can be calculated when the metric values of the single links are known. When links are concatenated, we propose two options: *Adding* the values of the single links or taking the *minimum* value. The latter strategy focusses on the avoidance of bottleneck routers, while the first one minimizes the overall queue lengths.

4.1.5 Bandwidth, Capacity, Throughput

Name	Bandwidth, capacity, flow capacity, throughput, transfer rate
Factors of Influence	see Figure 4.6
Mathematical Structure	Concatenation operator: Minimum Aggregation operator: Addition Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: Internet, QoS Optimisation goal: Maximise Throughput, load-balancing
Implement. Characteristics	Layer: Measurement on L3-L7 Information acquiring method: Active probing Analytically defined
Evaluation	Performance: Throughput = ++ Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = - Implementation: Complexity = 0 Popularity = ++
Adaptations	
Main References	Lai and Baker [57], Carter and Crovella [58]

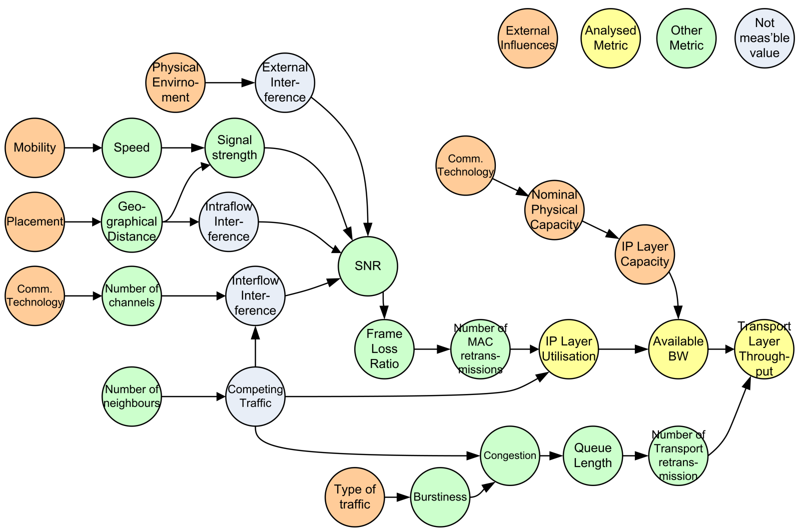


Figure 4.6: Dependencies of the bandwidth metric

Bandwidth and related routing metrics indicate the capacity of data which can be sent over a link within a given time. From the perspective of a node, this is equal to the transfer rate of a link. Many factors other than theoretical physical bandwidth have a significant effect on this metric, e.g. packet loss ratio [57]. Bandwidth metrics are quite popular, especially for QoS applications.

There is a considerable lack of clarity in the terminology of this area. In Table 4.1, we will shortly summarise the most important terms used in the relevant literature. If not stated otherwise, the definitions are adopted from Chimento and Ishac [59].

Nominal Physical Link Capacity	The theoretical maximum amount of data that a link can support on the physical layer, without concurrent traffic and in a perfect physical environment.
IP Layer Capacity	The maximum number of IP layer bits that can be transmitted from a sending node and correctly received by a receiver over the link or path during the interval [T, T+I], divided by I, under the assumption that all network resources are free of contention. Some people use the terms *flow capacity*, *bottleneck bandwidth* [60] or *base bandwidth* in the same sense.
IP Layer Usage	The actual number of IP layer bits correctly transmitted from any source during the interval [T, T+I], divided by I.
IP Layer Utilisation	The IP Layer Usage divided by the IP Layer Capacity.
IP Layer Available Capacity	IP Layer Capacity * (1 - IP Layer Utilisation). Often, the term *available bandwidth* is used [60].
Transport Layer Throughput	The actual number of IP layer bits correctly transmitted from any source, as experienced by an application using the transport layer. Sometimes, this value is also called "goodput".
Free slots	Sometimes, the measurement of bandwidth is reduced to the determination of the number of free communication slots. Lin and Liu [61] use this approach in their QoS routing protocol with TDMA and CDMA slots.
Bulk Transport Capacity (BTC)	The expected long term average data rate of a single congestion-aware transport connection (e.g. a TCP connection) when big amounts of data are transferred [62].
Congestion Avoidance Capacity (CAC)	The data rate of a fully specified implementation of a congestion avoidance algorithm, with the restriction that the retransmission time-out and slow-start algorithms are not invoked [62].
Maximum flow capacity of an IP cloud	For a given entry and an exit point, the greatest transmission rate achievable, in bits per second, if we had all of the links and routers in the cloud at our disposal. A cloud is defined as a combination of links which may contain various disjoint and conjoint paths [62].

Table 4.1: Bandwidth-related metrics: Definitions

Combination

The bandwidth metric is concave, i.e. when the links of a path are concatenated, the minimal bandwidth determines the cost of the whole path [63; 64].

$$BW(path) = \min_i BW(link_i)$$

When a multipath routing protocol is used, the bandwidth of two parallel links can be aggregated additively.

Measuring Bandwidth

A vast number of bandwidth measuring techniques has been proposed. Often, TCP throughput is used to measure available bandwidth. This makes sense if we want to measure the current *Transport Layer Throughput*. However, this passive measurement method does not reflect the available bandwidth if the link is not fully loaded.

An active bandwidth measurement technique are PATHCHAR algorithms [65; 66]. This method tries to correlate round-trip times of probe packets with the Maximal Transfer Unit (MTU) of the single links of a path. To achieve reasonable accuracy, PATHCHAR needs to send up to 10 MB of test data [57], which is a prohibitively high burden for many networks – in particular for MANETs.

Packet Pair Delay

A third way to measure bandwidth is packet pairing. The packet pair method (sometimes called Packet Inter-Arrival Time method) was designed to measure the the queueing delays at intermediate nodes and the destination node of a packet. However, it also is possible to infer from the queueing delay to the bandwidth of a link, if packets of different sizes are examined.

To compute this value, a node sends out two consecutive packets to another node. The destination acknowledges the packets. If the packet encounters a bottleneck, the two packets are dispersed (see Figure 4.7). The delay between the arrival of the two packets can be used to estimate the time the packets were delayed due to queueing (see Keshav [67]).

Various research has been conducted on this topic. Carter and Crovella [58] present a thorough analysis of the problems that arise when applying packet pairing. They also synthesise two tools which are designed to measure base bandwidth as well as available bandwidth. Packet pairing suffers from the problem that the maximal amplitude of the measurable bandwidth is limited by the size of the probing packets. This effect becomes especially troublesome, when an asymmetric link is examined or when asymmetric traffic (e.g. HTTP connections) is used as piggy-back probes. Lai and Baker [57] propose a method to *detect*, at least, when this problem arises. Paxson [60] proposes two methods to search for Packet Bunch Modes (PBM) in order to find a robust bandwidth estimation: A sender-based method and a receiver-based one. Lai and Baker [57] propose a compromise between the two, which they call Receiver Only Packet Pair (ROPP).

Another implementation is described by Draves et al. [47]. They send first a small packet, then a large one, making the metric sensitive to bandwidth, as well. In contrast to other implementations, Draves et al. let the destination calculate the delay between the two packets and inform the sender about the delay instead of leaving this task to the sender.

Packet pair delay is unaffected by the egress queueing of the sending node. If the two packets are of the same size, the packet pair delay also is independent of channel loss rates. Apart from statistical variance, the packet pair metric is determined only by queue bottlenecks at intermediate nodes and the destination node. The packet pair metric is less susceptible to self-interference than e.g. RTT (see Section 4.1.2), but it is not completely immune against this phenomenon neither [47].

Compared to most other techniques, the packet pair method requires a relatively high quantity of overhead messages. There are a number of tools that are designed to measure packet pair delay in the Internet. A summary of such tools can be found in Lai and Barker [57].

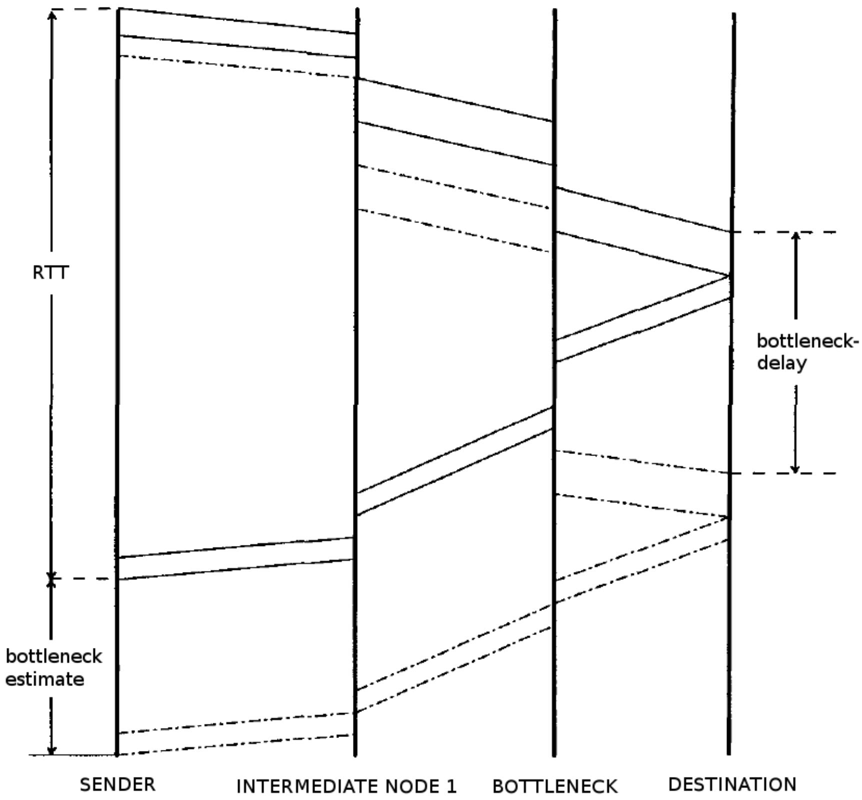


Figure 4.7: Packet pair method. Solid line = first packet, dashed line = second packet. (Figure adopted from [67])

Performance

Chang et al. [64] compared in a simulation experiment Minimal Hop Count with a bandwidth metric. They found that the throughput of the bandwidth metric was 20% higher the with hop count. However, it is not clear which technique Chang's metric uses to determine the capacity of a link.

4.1.6 Packet Loss Ratio

Name	Packet Loss Ratio (PLR), delivery rate
Factors of Influence	see Figure 4.8
Mathematical Structure	Concatenation operator: Multiplication Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: Wireless networks Optimisation goal: Maximise throughput, minimise energy consumption
Implement. Characteristics	Layer: Measurement on L2-L4 Information acquiring method: Passive monitoring, piggy-back or active probing Analytically and empirically defined
Evaluation	Performance: Throughput = + Performance: Delay = + Performance: Reliability = + Implementation: Measurability = + Implementation: Complexity = 0 Popularity = ++
Adaptations	ETX, mETX (see Sections 4.1.8 and 4.1.9), LQM
Main References	RFC 2680 [68]

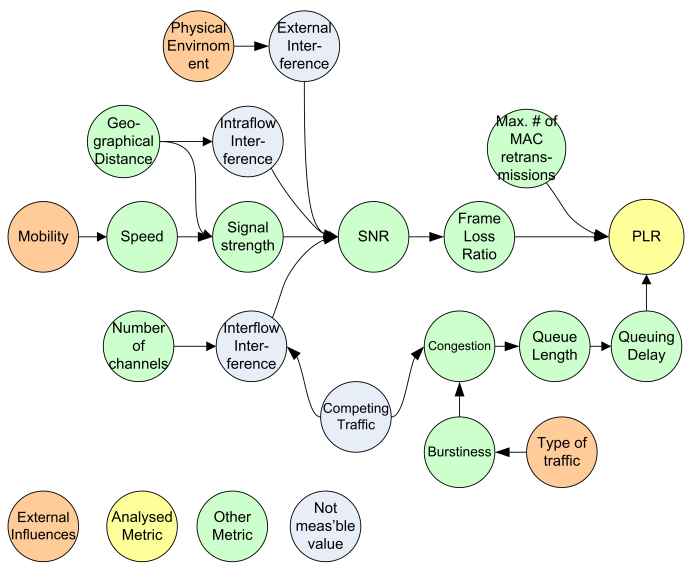


Figure 4.8: Dependencies of the packet loss ratio metric

Packet Loss Ratio (PLR) is a crucial variable for all applications. A high loss ratio degrades the communication quality of non-reliable protocols (e.g. for voice or video applications). With reliable transfer protocols, it potentially forces a high number of retransmissions, slows down communication and reduces the usable bandwidth.

One-Way vs. Round Trip Loss Ratio

Similar to the delay metric, PLR can either be measured on a round-trip basis, or only one-way. On one hand, round-trip measurements imply less implementation effort. On the other hand, many applications produce asymmetric traffic, e.g. file transfer traffic, web traffic or multimedia broadcasts. For these tasks, one-way PLRs are of more interest than bi-directional metrics. Packet loss ratios, in fact, often are highly asymmetric due to asymmetric queuing. Therefore, one-way packets reflect better the network conditions to which packets are subject. IETF did only issue RFCs for one-way PLRs: RFC 2680 [68] and RFC 3357 [69]. When TCP is used, the round-trip loss ratio can be determined by counting the missing packets in the current receiving window [70]. A special case of symmetric PLR is the Expected Transmission Count metric (ETX) which is described in Section 4.1.8.

Weighting

Andersen et al. [33] and Floyd et al. [45] all find that the unweighted average of losses for a given number of past packets is better suited for PLR metrics than EWMA (see Section 4.1.1). This can be attributed to the bursty nature of packet losses. RFC 3357 [69] proposes some definitions and statistical description methods for packet loss characteristics.

Combination

When links are concatenated, the likelihood of packet losses increases according to the laws of probability. Thus, the PLR of an entire path consisting of concatenated links l is computed with the following formula [33]:

$$PLR_{path} = 1 - \prod_{l \in path} (1 - PLR_l)$$

The calculation of aggregated links cannot be defined in such a straightforward fashion. We propose to use the average of the link PLRs. This can be enhanced by weighting the links according to their link bandwidth.

Causes for Packet Loss

In wired Internet networks, the main cause for packet losses are overfull routing queues, routine maintenance, unexpected outages or router reboots. These losses mostly occur in bursts [71]. In wireless networks, physical reasons such as interference or node mobility emerge as other major causes for packet loss. It is vital to differentiate between the different causes, as the appropriate reaction to such losses may be fundamentally different.

A first method to distinguish between different loss causes is outlined by Biaz and Vaidya [72]. Fu et al. [70] improved this method and proposed to combine four different metrics heuristically in order to identify the network state. These four metrics are: Inter-packet delay difference (a synonym for packet pair delay, see Section 4.1.5), short-term throughput, Packet Reordering Ratio PRR (see Section 4.1.7), and PLR.

Link Quality Metric

Yarvis et al. [73] used a modified PLR metric: Every link is given a weight according to its PLR. However, this weights were not handled multiplicatively, but additively. In a real world experiment, they measured an improvement of 20 to 32% in terms of end-to-end loss rates. With growing networks size, the improvement reduced to 2 to 4%. However, it is probable that this decay is a result of the limited processing power of the devices they deployed.

4.1.7 Packet Reordering Ratio

Name	Packet Reordering Ratio (PRR)
Factors of Influence	see Figure 4.9
Mathematical Structure	Concatenation operator: Multiplication Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: Internet, QoS Optimisation goal: Minimise delay
Implement. Characteristics	Layer: Measurement on L3-L4 Information acquiring method: Passive monitoring, piggy-back or active probing Analytically and empirically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = ++ Performance: Reliability = 0 Implementation: Measurability = + Implementation: Complexity = 0 Popularity = -
Adaptations	
Main References	Morton et al. [74]

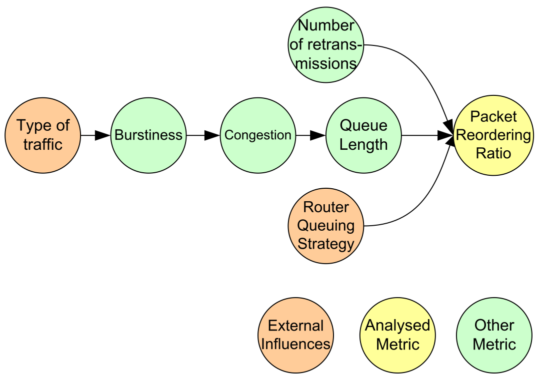


Figure 4.9: Dependencies of the packet reordering ratio metric

A packet is reordered if it arrives before its predecessor. The frequency of such reordering events can be interpreted as routing metric. This variable is of special interest for real-time applications or video and voice communication. For the latter tasks, the packet reordering ratio is one of the factors that determines the delay and delay variation experienced by the user.

The Packet Reordering Ratio (PRR) can be measured easily if the used transfer protocol uses incrementing sequence numbers. PRR always is higher or equal than Packet Loss Ratio by definition. Recently, Morton et al. [74] proposed a PRR standard for IETF.

4.1.8 Expected Transmission Count

Name	Expected Transmission Count (ETX)
Factors of Influence	see Figure 4.10
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, symmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Maximise throughput
Implement. Characteristics	Layer: Measurement on L2-L4 Information acquiring method: Passive monitoring, piggy-back or active probing Analytically defined, combined metric
Evaluation	Performance: Throughput = ++ Performance: Delay = + Performance: Reliability = + Implementation: Measurability = + Implementation: Complexity = 0 Popularity = ++
Adaptations	mETX, ENT (see Section 4.1.9)
Main References	De Couto et al. [75]

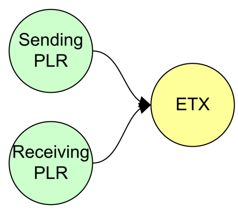


Figure 4.10: Dependencies of the ETX metric

Expected Transmission Count (ETX) probably was the first metric specifically designed for MANETs. Starting with the observation that minimal hop count is not optimal for wireless networks, De Couto et al. proposed a metric that bases on bidirectional loss ratios [75]. It aims to predict the number of transmissions (including retransmissions) required to send a packet over a link. This is an appealing concept: Minimizing the number of transmission does not only optimize the overall throughput, it does also minimize the total consumed energy if we assume constant transmission power levels [76].

Let d_f be the expected forward delivery ratio and d_r be the reverse delivery ratio i.e. the probability that the acknowledgement packet is transmitted successfully. Then, the likelihood that a packet arrives and is acknowledged correctly is $d_f \cdot d_r$. De Couto et al. assume that each attempt to transmit a packet is statistically independent from the precedent attempt, independent of the packet size et cetera, i.e. the sending attempt can be considered a Bernoulli trial. Then, the expected number of transmissions is:

$$ETX = \frac{1}{d_f \cdot d_r}$$

The delivery ratios are measured using broadcast probes at link-layer level. Therefore, ETX only makes sense for networks with link-layer retransmission, for example for 802.11b.

Although ETX bases on delivery ratios, which have a direct effect on throughput, ETX is independent from link load in a first approximation[3]. In other words: ETX does not try to route around congested links. On one hand, this is a disadvantage of this metric. On the other hand, ETX does not run the risk of self-interference because of this.

Combination

The ETX of a path is defined as the sum of the metric values of the links that form this path. De Couto et al. did not specify how link metrics should be aggregated.

Performance and Popularity

Measurements conducted by De Couto et al. [75] on a static test-bed network show that ETX performs up to two times better than minimal hop-count for long links in terms of throughput. This advantage reduces with increasing mobility of the nodes. ETX is one of the few non-hop-count metrics that has been implemented in practice in MANETs, namly in olsrd [77].

[3] On the physical layer, however, the number of retransmissions depends on the interference that is caused by competing traffic as well as from interference stemming from the same data flow. It is not clear how substantial the impact of this effect is in an average communication setting.

4.1.9 mETX, ENT and EDR

Name	modified ETX (mETX), Effective Number of Transmissions (ENT), Expected Data Rate (EDR)
Factors of Influence	see Figure 4.11
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, symmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Maximise throughput
Implement. Characteristics	Layer: Measurement on L2-L4 Information acquiring method: Passive monitoring, piggy-back or active probing Analytically defined, combined metric
Evaluation	Performance: Throughput = ++ Performance: Delay = + Performance: Reliability = + Implementation: Measurability = + Implementation: Complexity = 0 Popularity = -
Adaptations	
Main References	Koksal and Balakrishnan [76]

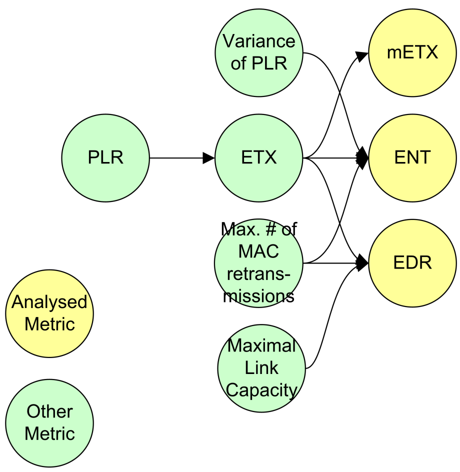


Figure 4.11: Dependencies of the mETX, ENT and EDR metrics

Packet loss can vary significantly over different orders of magnitude depending on the radio technology of a system. In fact, packet loss probability of 802.11 systems show significant long-term dependence. This can lead to poor performance when the average of the packet loss ratio is taken as basis for ETX. Koksal and Balakrishnan [76] propose two metrics that approach this problem:

modified ETX is defined as $mETX = exp(\mu + \frac{1}{2}\sigma^2)$, with μ being the estimated average packet loss ratio of a link and σ^2 the variance of this value. Like ETX, mETX is additive over concatenated links.

Effective Number of Transmissions is defined as $ENT = exp(\mu + 2\delta\sigma^2)$. The parameter δ depends on the number of subsequent retransmissions which will cause the link layer protocol to give up a sending attempt.

Expected Data Rate (EDR)

Park and Kasera [78] try take into account the non-optimality of IEEE 802.11 medium access scheduling in order to predict the capacity of a link. Their Expected Data Rate (EDR) metric uses ETX, the medium access contention window and the maximal capacity of a link.

4.1.10 Expected Transmission Time

Name	Expected Transmission Time (ETT)
Factors of Influence	see Figure 4.12
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Maximise throughput
Implement. Characteristics	Layer: Measurement on L2-L4 Information acquiring method: Active probing Analytically defined, combined metric
Evaluation	Performance: Throughput = ++ Performance: Delay = + Performance: Reliability = - Implementation: Measurability = - Implementation: Complexity = 0 Popularity = -
Adaptations	Weighted Cumulative ETT (WCETT), Multi-Channel Routing metric (MCR), see Section 4.1.11
Main References	Draves et al. [79]

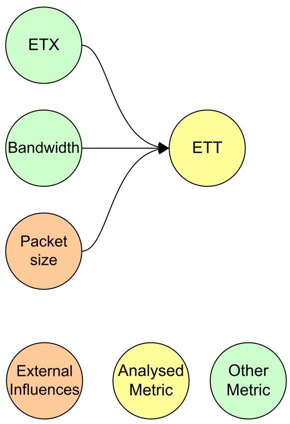


Figure 4.12: Dependencies of the ETT metric

Draves et al. [79] observed that ETX (see Section 4.1.8) did not perform optimal under certain circumstances. For example, ETX prefers heavily congested links to unloaded links, if the link-layer loss rate of congested links is smaller than on the unloaded links. Therefore, the Expected Transmission Time metric (ETT) has the goal to incorporate throughput into its calculation.

Let S be the size of the probing packet and B be the measured bandwidth of a link, then the ETT of this link is defined as follows:

$$ETT = ETX * \frac{S}{B}$$

Draves et al. propose to use packet pairing (see Section 4.1.5) to measure the bandwidth on each link. Of course, this sharply increases the measuring overhead. ETX is measured as described in Section 4.1.8.

4.1.11 WCETT and Other Interface-Dependent Metrics

Name	Weighted Cumulative Expected Transmission Time (ETT), Multi-Channel Routing metric (MCR), Metric of Interference and Channel-switching (MIC), PARMA
Factors of Influence	see Figure 4.13
Mathematical Structure	Concatenation operator: Addition / more sophisticated terms (see following paragraphs) Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Maximise throughput (by optimising channel usage)
Implement. Characteristics	Layer: Measurement on L1-L3 Information acquiring method: Node-dependent and active probing Analytically defined, combined metrics
Evaluation	Performance: Throughput = ++ Performance: Delay = + Performance: Reliability = - Implementation: Measurability = - Implementation: Complexity = 0 Popularity = -
Adaptations	
Main References	Draves et al. [79], Yang et al. [41]

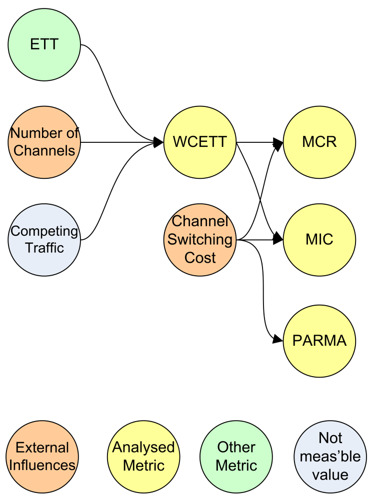


Figure 4.13: Dependencies of the WCETT, MCR, MIC, PARMA metrics

Many wireless technologies, including 802.11a/b/g, provide multiple non-overlapping channels. It is obvious that this property should be exploited by link layer protocols and just as obvious to devise metrics take advantage of this feature as well.

Draves et al. [79] propose a special technique to compose the single links of the ETT metric and call it Weighted Cumulative ETT (WCETT). They suggest to compute the path metric not just as the sum of the metric values of the links that form this path. When we simply sum over the link metrics, we neglect the fact that concatenated links may interfere with each other, if they use the same channel.

Draves et al. first assume that communication on the same channel always interferes. This forces the links to split up the bandwidth. Let k be the total number of channels of a system. Then, the sum of transmission times of hops on channel j is:

$$X_j \; = \sum_{\text{Hop i is on channel j}} ETT_i \qquad 1 \leq j \leq k$$

The total path throughput will be dominated by the bottleneck channel, which has the largest X_j. Draves et al. propose to use a weighted average between the maximum value and the sum of all ETTs. This results in the formula:

$$WCETT \; = (1-\beta) * \sum_{i=1}^{n} ETT_i + \beta * \max_{1 \leq j \leq k} X_j$$

with $0 \leq \beta \leq 1$ being a tunable parameter. Draves et al. [79] describe different interpretation possibilities for this parameter.

WCETT has a serious drawback: It is not immediately clear if there is an algorithm that can compute the path with the lowest weight in polynomial time or less. If a traditional algorithm for finding the shortest path is applied, there might even be the risk of routing loops [41].

Performance

In their static test-bed implementation, Draves et al. measured that WCETT outperformed ETX by a factor of two and minimal hop count by a factor of four, when two different 802.11 radio cards per station were used.

Multi-Channel Routing Metric MCR

Kyasanur and Vaidya [80] extend WCETT so that it takes into account the cost of changing channels. Let $InterfaceUsage(i)$ be the fraction of time a switchable interface was transmitting on channel i and let $p_s(j)$ be the probability that the used interface is on a different channel when we want to send a packet on channel j. If we assume that all the current interface idle time can potentially be used on channel j, we can estimate $p_s(j)$ as

$$p_s(j) = \sum_{\forall i \neq j} InterfaceUsage(i)$$

Let $SwitchingDelay$ denote the switching latency of an interface. This value can be measured offline. Then, the cost of using channel j is measured as

$$SC(c_i) = p_s(j) * SwitchingDelay$$

Kyasanur and Vaidya want to prevent that paths are chosen which require frequent channel switching. Therefore, they include the switching cost into the ETT metric. This results in the following definition for their metric:

$$MCR \; = (1-\beta) * \sum_{i=1}^{n} (ETT_i + SC(c_i)) + \beta * \max_{1 \leq j \leq k} X_j$$

Metric of Interference and Channel-switching (MIC)

Independently from Kyasanur and Vaidya, Yang et al. [41] proposed another metric which incorporates the cost of switching channels. Furthermore, their metric takes also into account the influence of inter-flow interference. MIC of a path p is defined as follows:

$$MIC = \alpha \sum_{\text{link l} \in \text{p}} ICU_l + \sum_{\text{node n} \in \text{p}} CSC_n$$

In this equation, α is a tunable weighting factor and CSC_n refers to the cost for a node to switch from one channel to another. ICU_l is the *Interference-aware Resource Usage*. This value is computed from the ETT of a link and the number of nodes which may interfere with a transmitted packet.

PARMA

A very similar concept is followed by the "PHY/MAC Aware Routing Metric for Ad-Hoc Wireless Networks with Multi-Rate Radios (PARMA)" proposed by Zhao et al. [81]. This metric is defined as follows for a path p:

$$PARMA(p) = \sum_{links \in p} \left(\frac{Packet\_Size}{Link\_Speed} + T_{access}\right)$$

where T_{access} is the the medium access time spent by the packet getting access to a link. Zhao et al. assume that this value is available from the MAC layer.

4.2 Radio Information

The physical layer of wireless networks is far more complex than its wired counterpart. Especially the phenomenon of interference turns out to be not only a challenge when developing the physical or MAC layers of network, but it should also be taken into account for routing purposes.

4.2.1 Signal strength, SNR

Name	Signal Strength, Signal-to-Noise Ratio (SNR), Interference
Factors of Influence	see Figure 4.14
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Minimise transmission energy / Minimise interference
Implement. Characteristics	Layer: Measurement on L1 Information acquiring method: Passive monitoring or active probing Analytically and empirically defined
Evaluation	Performance: Throughput = + Performance: Delay = + Performance: Reliability = + Implementation: Measurability = ++ Implementation: Complexity = ++ Popularity = 0
Adaptations	
Main References	Dube et al. [82]

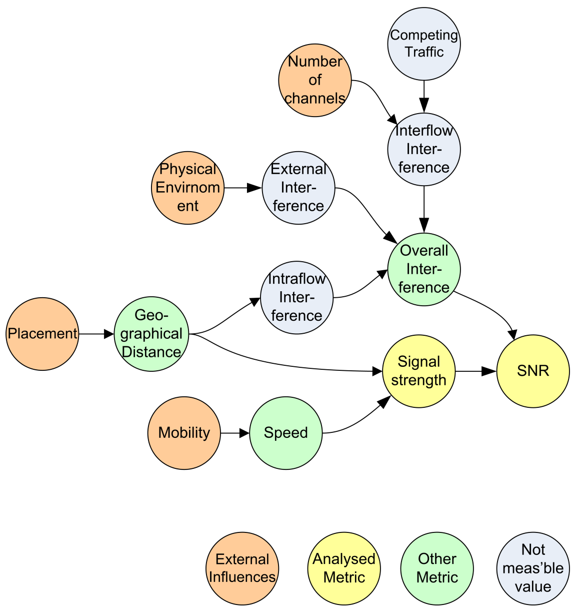


Figure 4.14: Dependencies of the signal strength metric

Many wireless technologies – among them IEEE 802.11 – use beacon packets in order to detect neighbouring hosts. Through these beacons, network devices can measure the signal strength at which a packet is received. Signal strength can be considered as an indication for the link quality and the distance between two nodes.

Dube et al. [82] use signal strength[4] for their Signal Stability-based Adaptive Routing protocol (SSA). By simulation, they found that the number of route reconstructions was up to 60% smaller with a signal strength-based metric than with a hop count metric. This was especially apparent in dense networks.[5] De Couto et al. [3] show that the signal strength of IEEE 802.11 systems does not correlate with the delivery rate of a link, but only with the geographical distance of two communicating nodes.

Signal-to-Noise Ratio

The signal strength alone does represent the quality of a communication link only to a limited degree. The amount of information that can be transferred on a channel depends on the present noise as well. Usually, the Signal-to-Noise Ratio (SNR) is used as measure for channel quality. IEEE 802.11 systems, for example, adapt the sending speed according to the measured SNR value [83].

Although SNR is an excellent indicator for link capacity in theory, it is not the case in reality. Lampe et al. [84] show that SNR is an optimal predictor for packet error rate only when the occurring noise is solely additive white Gaussian noise. They propose to make use of supplementary information about the quality of a channel instead. Such information, e.g. the channel transfer function, is provided by some receiver architectures.

Interference Estimation

Iannone and Fdida [85] propose a method to predict the interference which is caused when a link is used, if this value is not directly available from the physical layer. They do not directly estimate the interference, but they use a heuristic instead which allows to compare the potential interference of different links with each other. This heuristic takes into account the current transmit power level P, the maximal power level of the device P_{max}^2, and the number of neighbours $N(P)$ which can be used at a certain power level.

$$I(P) = \frac{N(P)}{N(P_{max}) + \beta} \sqrt{\frac{P^2 + P_{max}^2}{2 * P_{max}^2}}$$

[4] In fact, an Exponentially Weighted Moving Average (EWMA, see Section 4.1.1) of the current signal strength value was used.

[5] Dube et al. found also that their SSA protocol performed significantly worse when they included location stability into their algorithm [82]. We suppose that this probably was caused by the simulation setting they used. Further, we suspect that the positive results that Dube et al. measured were due to the fact that signal stregth was a better predictor for link stability than the location stability metric they used (see Section 4.4.3).

4.2.2 Medium Time Metric

Name	Medium Time Metric (MTM), Estimated Transmission Time (ETT)
Factors of Influence	see Figure 4.15
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Minimise the time a packet occupies the medium
Implement. Characteristics	Layer: Measurement on L2-L3 Information acquiring method: Passive monitoring or active probing Analytically defined, combined metric
Evaluation	Performance: Throughput = 0 Performance: Delay = + Performance: Reliability = + Implementation: Measurability = - Implementation: Complexity = 0 Popularity = -
Adaptations	
Main References	Awerbuch [86; 83]

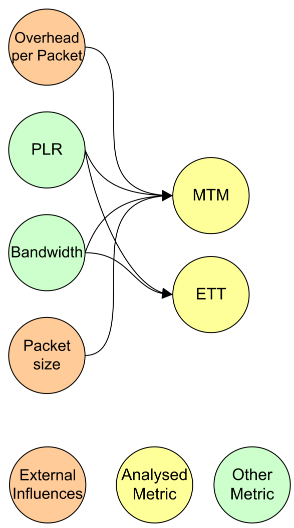


Figure 4.15: Dependencies of the MTM and ETT metrics

In wireless networks, individual links may interfere. Transmissions compete for the medium with other transmissions in the same geographical area. The longer the physical distance of a hop, the higher is the energy necessary for the transmission and the more other hops are affected. Additionally, In IEEE 802.11a/b/g networks, the data rate is higher the closer two hosts are located. Traditional hop count metrics favour long links, which may be highly suboptimal in wireless networks. The Medium Time Metric (MTM) proposed by Awerbuch et al. [86; 83] aims at minimising the time during which the physical medium is consumed instead. The MTM of a packet p on a path π is defined as follows:

$$MTM(\pi, p) = \sum_{\forall e \in \pi} \tau(e, p)$$

where $\tau(e, p)$ is the time required to transfer packet p over link e. This value is composed of the following components.

$$\tau(e, p) = \frac{overhead(e) + \frac{size(p)}{rate(e)}}{reliability(e)}$$

Link overhead can be computed from standards and specifications as well as from the type and configuration of the used wireless device. The packet size should be easily available through the routing protocol. Link transfer rate and reliability usually are known to the MAC layer. However, this information often is not accessible to higher network layers because the techniques used for auto-rate selection on the MAC layer is considered proprietary. It is possible to estimate the values for transfer rate and link reliability by probing. Though, this information produces unnecessary overhead and less accurate results than inter-layer communication would. Therefore, Awerbuch et al. [83] would favour that radio card manufacturers provide a standard interface in order to enable access to this information by higher network layers. Although we agree with them principally, one should not expect that all problems of measuring transfer rate or link reliability be solved at once thereby.

Performance

Awerbuch et al. [83] measured a en-to-end throughput which was equal to minimum hop count and ETX in short distances. When the distances were larger, minimum hop count and ETX found routes with a few hops. MTM selected multi-hop paths with more hops but higher capacity. For this reason, the resulting end-to-end throughput was up to 20 times higher with MTM than with the other metrics.

Similar Metrics

Although they base on different motivations, the MTM metric is basically the same as the Expected Transmission Time (ETT) metric described in Section 4.1.10. One difference is that MTM, unlike ETT, includes a term for communication overhead. Furthermore, ETT uses explicitly ETX as measurement for reliability while this is not defined for MTM.

Aguayo, Bicket et al. [87; 88] use again a very similar metric for their WMN implementation. However, in this metric, the size of a packet is assumed to be constant at 1500 Bytes and the overhead is neglected. Confusingly, they use the name Estimated Transmission Time (ETT) for their metric. Though it is very similar, this metric should not be mixed up with the *Expected* Transmission Time metric described in Section 4.1.10. The latter metric does not assume constant packet sizes and uses ETX as link reliability measure. For a link e, the Estimated Transmission Time metric is defined as follows:

$$\text{Estimated Transmission Time} = \frac{1}{reliability(e) * rate(e)}$$

4.3 Topology

Topology metrics base on an abstract concept of networks. They only take into account the presence or absence of links as well as the neighbourhood relations of the network nodes. Although this approach abstracts from a broad range of variables, it is widely used due to the simplicity it offers.

4.3.1 Number of Neighbours

Name	Number of neighbours
Factors of Influence	see Figure 4.16
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: Internet, MANETs Optimisation goal: Minimise number of hops
Implement. Characteristics	Layer: Measurement possible on L2-L7 Information acquiring method: Monitoring, piggy-back or active probing possible Analytically defined
Evaluation	Performance: Throughput = - Performance: Delay = ++ Performance: Reliability = - Implementation: Measurability = ++ Implementation: Complexity = ++ Popularity = -
Adaptations	
Main References	

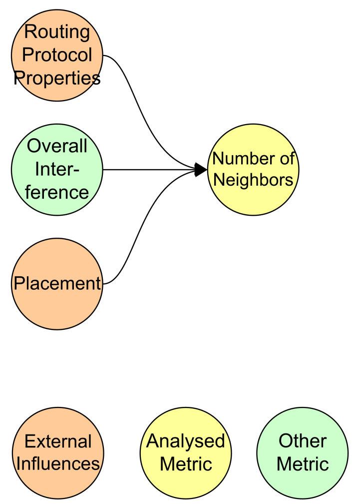


Figure 4.16: Dependencies of the number of neighbours metric

The Number of Neighbours metric is defined by how many other nodes can be reached from a node at a given moment. This idea is appealing because it might make it possible to benefit from the small world problem. This phenomenon was first described for social networks by Milgram [89] and later for computer networks (for example by Kleinberg [90]). The small world theory states that, in certain networks, there is always a short path between any two nodes. This is the case because in these “small world networks”, a class of highly interconnected nodes exists. These nodes play the role of wormholes and relate distant network regions. Such nodes typically are characterized by a high node degree i.e. a large number of neighbours. We suppose that in MANETs, this kind of topology will not occur usually. However, this structure is symptomatic for WMNs, where routers fulfil the function of wormholes the Internet.

We should be aware that neighbourhood not necessarily is reciprocal in wireless networks. If one node can contact another, this does not mean that this is also possible in the other direction. Therefore, the number of neighbours metric is unidirectional, i.e. the indegree not compulsorily equals its outdegree.

The number of neighbours is a property of a network node. However, we defined metrics to be properties of network links in Section 3.3.4. Nevertheless, there are various ways how we can map indegree and outdegree of the two adjacent nodes of a link to the link itself.

Measurement

If the number of neighbours is not available from the MAC layer, the transport layer or the application layer, the degree of a node can be found by regular probing. The indegree can be measured by listening to the incoming probes. The outdegree can only be determined if probing messages are acknowledged. Probing brings along several problems, as stated in Section 4.1.1.

4.3.2 Number of Paths to a Node

Redundancy is a central criteria for MANET and WMN routing algorithm. Let there be one particular node with a special function in a network. Then, the number of disjoint paths to this node traversing a link can be considered as a metric for that link. In the case of WMNs, this metric could be defined as the number of paths to an Internet gateway. Many different paths to an Internet gateway means high tolerance against link failures. However, the number of paths to an Internet gateway should not simply be used as a single metric, because many paths to a gateway can also signify a long distance to the next gateway.

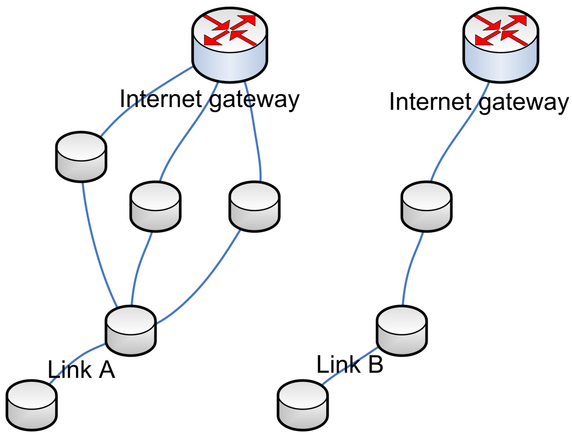


Figure 4.17: Link A has lower cost than Link B, as more paths lead from Link A to an Internet gateway than from Link B.

Surprisingly few reflections have been made in literature on the relationship between redundancy and metrics and many questions remain open. It is not trivial to determine the number of paths from one node to another and a number of considerations have to be kept in mind. For example, whether only completely disjoint paths should be counted or partly disjoint paths as well.

4.3.3 Hop Count

Name	Hop count, minimal hop count
Factors of Influence	see Figure 4.18
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, symmetric, one-dimensional
Design Goals	Target Platform: Internet, MANETs Optimisation goal: Minimise number of hops
Implement. Characteristics	Layer: Measurement possible on L2-L3 Information acquiring method: Piggy-back or active probing possible Analytically and empirically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = ++ Implementation: Complexity = ++ Popularity = ++
Adaptations	Widest shortest path
Main References	

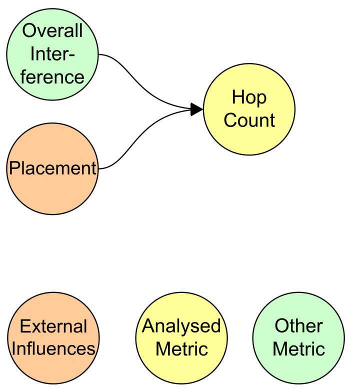


Figure 4.18: Dependencies of the hop count metric

The concept of the hop count metric is unbeatable in its simplicity: Every link counts as one equal unit, independent from the quality or other characteristics of the link. The ease of implementation made hop count to the most widely used metric for MANETs by far. It is implicitly or explicitly used e.g. in OLSR [91], DSR [24], DSDV [22] and AODV [23].

Performance

De Couto et al. [3] as well as Yarvis et al. [73] observed in their tests on two different static test-bed environments using DSDV routing protocol, that using minimal hop count will not result in optimal performance. This is because the selection of minimum hop paths prefers long links. In multi-rate wireless networks, this typically results in paths with links operating at low rates [83]. Because of polynomial energy consumption in radio transmission, longer links are more expensive than several concatenated shorter ones in terms of energy. (Of course, this is only the case, if the used devices are able to adapt their transmission energy.)

But when nodes become mobile, things begin to look quite different. Draves et al. [47] compared hop count, RTT, packet pair, and ETX in a mobile test bed. They find that minimal hop count outperforms the quality-aware routing metrics under the presence of mobility and high channel variability. They attribute this finding to the quicker reaction of hop count to fast topology changes. ETX needs some time until a stable value for the packet loss ratios is determined. RTT and packet pair did not even find a stable state at all due to self-interference. However, it is not clear whether Draves et al. chose the probing technique for ETX, RTT and packet pair adequately (see Section 4.1.1).

Syrotiuk and Bikki [92] show that congestion and packet loss may have a big influence on the path found by the hop count metric. This problem can be reduced by giving higher routing priority than normal traffic to the probes that test links.

4.3.4 Widest Shortest Path and Shortest Widest Path

In many implementations, an arbitrary path is chosen, if there are several paths with minimal length that each have the same number of hops. Guerin et al. [93] propose to choose the path with the largest available bandwidth in this case, the so-called Widest Shortest Path. Wang and Crowcroft [9] propose a Shortest Widest Path algorithm i.e. from all paths with the highest bandwidth, the path with the fewest hops is chosen. However, Wang and Crowcroft show in their paper that it is not possible to find such a shortest widest path within polynomial time.

4.4 Mobility, Geography

The use of the geographical position of nodes is an often cited approach to simplify routing in ad hoc networks. In such systems, it is assumed that each node knows its physical location as well as the positions of its neighbours. this information can be acquired e.g. from a Global Positioning System (GPS).

Location information can be included into routing algorithms in various ways. To design a special metric is only one of these ways. For a survey on location-based routing – often called geocasting – please refer to [94].

4.4.1 Geographical Distance

Name	Geographical distance
Factors of Influence	Only placement, no network-immanent factors
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, symmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Find “short” path
Implement. Characteristics	Information acquiring method: External position acquisition Analytically and empirically defined
Evaluation	Performance: Throughput = + Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = + Implementation: Complexity = ++ Popularity = +
Main References	

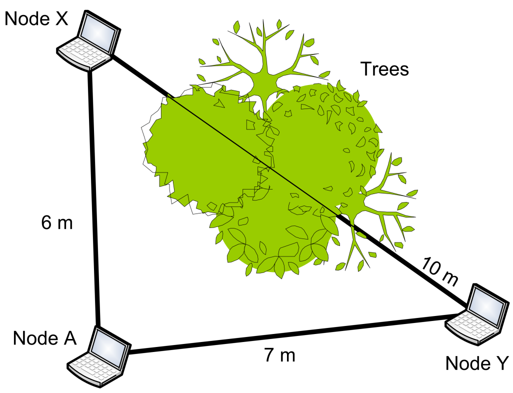


Figure 4.19: Although the direct link from X to Y is less distant, it is probably more efficient to route traffic over A and accept a geographically longer path.

The most straight-forward application of location information is to use the geographical distance as metric. However, this value does not represent a very useful information in many cases. Although distance has an influence e.g. on signal strength, other factors usually are more important. The quality of a link can be significantly decreased by obstacles such as walls or trees (see Figure 4.19). This is better reflected directly by a signal strength metric than by using the geographical distance.

Distance to Destination

Another way to use positioning information is to attribute the geographical distance to the destination as cost of a link instead of the distance between neighbours. However, this implies that the metric is dependent on the destination which contradicts to the definition of metrics that we proposed in Section 3.3.4. In fact, this kind of rather reflects a special routing algorithm than a metric.

Several papers propose to combine geographical distance to destination with other metrics. Seada et al. [95] couple distance with packet reception rate. They find in their tests that this solution is especially apt for systems that use automatic repeat request (ARQ) techniques. Similarly, Zhang et al. [43] propose a metric that combines distance to destination with the expected latency. In their huge test bed implementation, they measure that their "expected MAC latency per unit-distance to the destination" metric (ELD) performed considerably better than Seada's metric and better than ETX in terms of latency, energy consumption and especially route stability.

4.4.2 Speed

Name	Speed of nodes
Factors of Influence	Only placement, no network-immanent factors
Mathematical Structure	Concatenation operator: Addition or maximum Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric (symmetric for relative speed), one-dimensional
Design Goals	Target Platform: Highly dynamic MANETs Optimisation goal: Find stable paths
Implement. Characteristics	Information acquiring method: External position acquisition Analytically and empirically defined
Evaluation	Performance: Throughput = + Performance: Delay = + Performance: Reliability = 0 Implementation: Measurability = + Implementation: Complexity = ++ Popularity = +
Adaptations	Relative speed of two nodes
Main References	

Quality and stability of a link are highly dependent on the speed of a node. The faster a node moves, the higher is the probability that a link with this node will break within short time. One technique for measuring the speed of a node is to have information on the current position of a node. This can be acquired e.g. using a GPS system. alternatively, Basu et al. [96] propose to estimate the relative velocity of two nodes by measuring the alteration of the signal strength.

Johansson et al. [97] define a metric for the average relative speed. Let $l(N,t)$ be the position of a node N at time t. Then, the relative velocity of the two nodes X and Y is

$$v(X,Y,t) = \frac{d}{dt}(l(X,t) - l(Y,t))$$

The mobility between a pair of nodes X and Y is defined by Johansson et al. as

$$M_{XY} = \frac{1}{T}\int_{t_0 \leq t \leq t_0+T} |v(x,y,t)|\, dt$$

4.4.3 Link Lifetime

Name	Link lifetime, link longevity, link availability, connectivity
Factors of Influence	see Figure 4.20
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined Other mathematical properties: Dynamic, symmetric, one-dimensional
Design Goals	Target Platform: MANETs Optimisation goal: Maximise long-term reliability of a path, minimise number of path recalculations
Implement. Characteristics	Information acquiring method: Various techniques Analytically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = + Performance: Reliability = ++ Implementation: Measurability = + Implementation: Complexity = 0 Popularity = 0
Adaptations	
Main References	Toh [98] (Associativity-Based Routing)

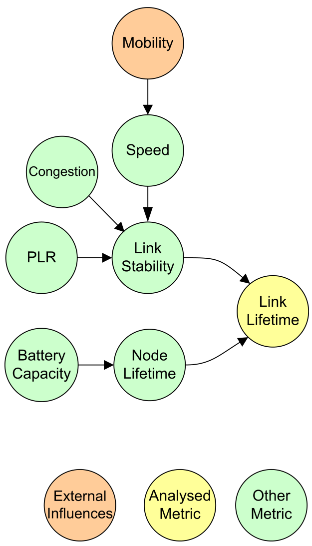


Figure 4.20: Dependencies of the link lifetime metric

A link exists as long as both communicating nodes are up and running and they have a radio connection that allows them to transfer data. In mobile wireless networks, finding routes that are stable over time is one of the most central challenges. The first routing algorithm which explicitly took into account the longevity of links was Associativity-Based Routing (ABR) proposed by Toh [98]. He based his routing algorithm on his observations of the movement of mobile communication devices in an office building. He found that mobile hosts (or rather the people who carried them) usually rest for several minutes before they start to move again. The longevity of these pauses varies heavily and seem to form a Pareto distribution (see Fig. 4.21). However, Toh did not investigate this any further.

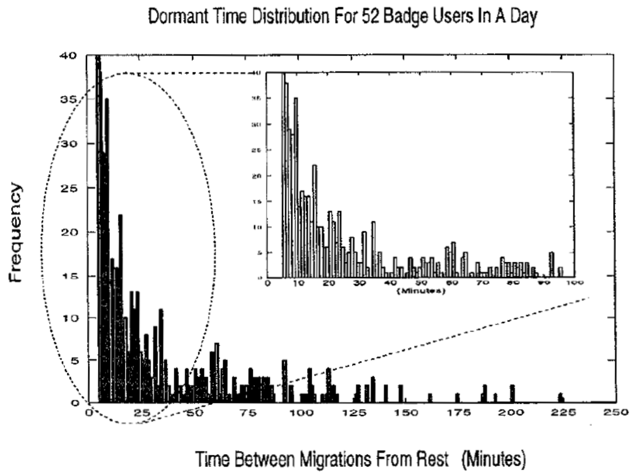


Figure 4.21: Mobility distribution of mobile nodes in an office environment. Figure adopted from [99].

Mcdonald and Znati [100] propose another routing metric, which defines a probabilistic measure of the availability of links that are subject to link failures caused by node mobility. They base their considerations on a random walk model. Each node is characterised by three values that describe the statistical distribution of the mean and variance of the speed of a node as well as an average interval time. Together with an estimated communication radius, Mcdonald and Znati derive a sophisticated function which estimates the expected availability of a link.

Various other metrics were proposed, based on other mobility models. Among them are the metrics described by Gerharz et al. [101] and Jiang et al. [102] that estimate the average residual lifetime of a link. However, all of these concepts base on the assumption that all nodes have similar mobility characteristics. In mesh networks, this obviously is not the case. For this kind of networks, we propose to devise special mobility metrics that use distinct node classes.

Connectivity

A simplified variety of link lifetime is the Connectivity metric. It has only two boolean values for a link, each of them indicating whether communication is possible in the one of the two directions at a given moment. A connectivity metric is described in IETF RFC 2678 [103].

4.4.4 Other Geographic Metrics

Punnoose et al. [104] propose to predict the propagation environment of a link using various information inputs. In particular, their protocol includes the use of terrain maps which is coupled with a location information system. Of course, this kind of prediction is only possible if accurate, detailed, up-to-date, three-dimensional map information is available. The cost for the acquisition of such data will be prohibitive in many cases, especially for indoor usage.

For large-scale applications, even weather forecasts could be used as input information. Like this, it would be thinkable to route traffic around thunderstorms, for example[6].

[6]For another aspect of the connection between weather data and wireless communication, please refer to [105].

4.5 Energy-Based

Unlike in wired networks, energy efficiency is a major concern in mobile networks. Sensor networks as well as portable communication devices only have restricted battery lifetime. How the problem of restricted energy in a network can be approached has been subject to various work. Feeney [106], and Stojmenovic and Lin [107] give good overviews of this topic . In this section, we will first give a short introduction into the basic principles of energy-restricted networks, and then describe the metrics which were proposed in this context.

4.5.1 Fundamentals

The energy consumed for sending and receiving a packet over a path is influenced by various factors:

- The transmission energy of a packet over one link from node i to node j can idealisingly be modelled as

$$e_{i,j} = h_{i,j}^{\alpha} + k \qquad 2 \leq \alpha \leq 4$$

  where α depends on the medium environment[7]. The variable k models a fix processing overhead for sending and receiving the packet and $h_{i,j}$ stands for the geographical distance. The equation shows that it is necessary to find a balance between minimising the transmission power and minimising the number of hops. However, these considerations only apply, if the MAC layer of the used nodes provides adaptive power control.
- In the presence of interference – be it external, inter-flow or intra-flow interference – the energy necessary for successful transmission increases. As above, only devices with power control can take advantage of this opportunity to save energy.
- Every time a packet must be retransmitted, the energy consumption increases, of course. Retransmissions may be caused at various places of the reference layer model.
- Routing overhead should be minimised as far as possible. Some strategies were proposed how this kind of traffic can be reduced. One such strategy bases on the idea to select the links over which packets are sent, instead of broadcasting routing information to all neighbours. See e.g. Ramanathans and Rosales' [109] or Meng and Rodoplus [110] works on topology control.
- Due to overhearing, high density of a network can have a considerable deteriorating effect on energy consumption [111].

There are different goals energy based routing and the corresponding metrics may follow:

- To minimise overall energy consumption.
- To maximise the time until the first node runs out of energy. This is equivalent to keeping the energy load in all nodes as equable as possible.[8] However, this implies that all nodes of a network will fail at the same time and the network will collapse entirely all of a sudden.
- In certain scenarios, the goal can be that those nodes fail first which are not vital to the functioning of the network. In other words: The goal is to maximise the time until the first message cannot be transmitted due to a network partition. Li et al. [112] prove that it is not possible to find optimal paths in this sense within polynomial time.

[7] See Feeney [106]. A more detailed investigation is given by Heinzelman et al. in [108].

[8] A very striking example for such a scenario are sensor networks that are reloaded at daytime using solar cells and need to keep their service up over night.

4.5.2 Energy Consumed per Packet

Name	Energy consumed per packet, transmission energy, Minimal Total Power metric (MTPR)
Factors of Influence	see Figure 4.22
Mathematical Structure	Concatenation operator: Addition Aggregation operator: Not defined (see following paragraphs) Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs, Sensor networks Optimisation goal: Minimise the total power consumption
Implement. Characteristics	Information acquiring method: Node-related Analytically and empirically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = 0 Performance: Reliability = + Implementation: Measurability = + Implementation: Complexity = ++ Popularity = +
Adaptations	
Main References	Singh et al. [113], Scott and Bambos [114]

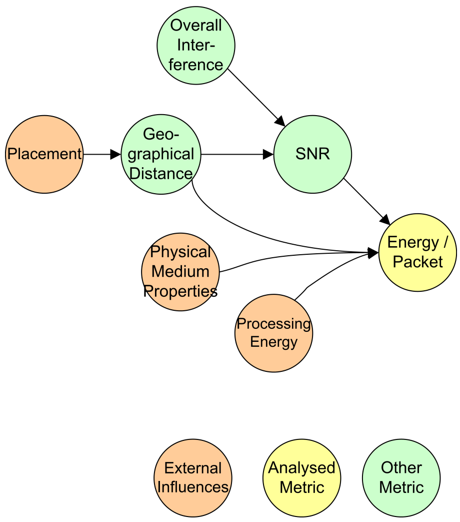


Figure 4.22: Dependencies of the energy per packet metric

Minimising the energy that is used per packet is the most basic approach for energy metrics. As this strategy minimises the overall energy consumption, too, this metric is called Minimal Total Power Routing metric (MTPR) by Scott and Bambos [114]. Singh et al. [113] formalise the idea of this metric: Let $e_{i,j}$ denote the energy consumed for transferring and receiving a packet from node i to the neighbouring node j. Then, the total energy required to send a packet from s to d is

$$E = \sum_{\text{all links on path n}} e_{n_i,n_{i+1}}$$

with n_i being the nodes forming the path from s to d. Thus, the concatenation operator of this metric is the addition. The aggregation operator is not clearly defined. We propose to take the average of the links, weighted by traffic distribution.

This metric will prefer multiple short hops over fewer longer hops. While this provides a minimal energy path and produce less interference, it occupies a big amount of other network resources.

A disadvantage of this metric is its lack of sensitivity for the remaining battery lifetime. This means that it is quite probable that some nodes will be burdened heavily with traffic and spend their energy much faster than others. Although, it might be sensible in some cases that specific nodes are loaded with more traffic, this must be done in well considered fashion.

In order to reduce the risk that bottleneck nodes are burdened to high, Michail and Ephremides [115] propose to weight the transmission energy with the number of free channels of a node at a given moment. Of course, this metric is appropriate for multi-channel systems only. Another approach to avoid the overload of some nodes is to take into account the remaining battery capacity of a node. This method is described in more detail in Section 4.5.3.

Measurement

Neither Singh et al. nor Scott and Bambos explain in detail how they envision to measure the energy used per transmission. However, it is clear that some information must be provided by the physical layer of a node. Combined with the model described in Section 4.5.1, an estimate of the consumed energy can be calculated.

4.5.3 Remaining Battery Capacity

Name	Remaining battery capacity
Factors of Influence	see Figure 4.23
Mathematical Structure	Concatenation operator: Addition (Minimum for MMBCR) Aggregation operator: Not defined Other mathematical properties: Dynamic, asymmetric, one-dimensional
Design Goals	Target Platform: MANETs, Sensor networks Optimisation goal: Maximise the network lifetime
Implement. Characteristics	Information acquiring method: Node-related Analytically defined
Evaluation	Performance: Throughput = 0 Performance: Delay = 0 Performance: Reliability = ++ Implementation: Measurability = 0 Implementation: Complexity = ++ Popularity = +
Adaptations	Energy cost per packet, MBCR, MMBCR, CMMBCR
Main References	Singh et al. [113], Chang and Tassulias [116], Kim et al. [111]

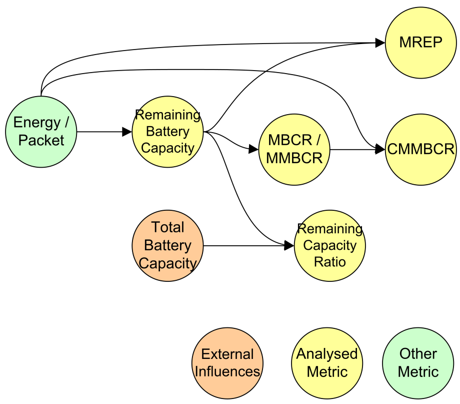


Figure 4.23: Dependencies of the remaining battery capacity metric

One approach to balance energy consumption over a network is to use the battery capacity of a node as basis for metrics. A metric that only bases on the current capacity of the node battery is used by Sheu et al. [117]. The *ratio of battery remaining capacity* R_{brc} is defined as

$$R_{brc} = \frac{E_i}{E_{max}} = \frac{\text{Battery remaining capacity}}{\text{Battery full capacity}}$$

Singh et al. [113] propose to supply each node with a cost value $f_i(E_i)$ that depends on its current battery capacity E_i. Singh et al. call this strategy Minimum Battery Cost Routing (MBCR). They propose as one possible choice for f_i

$$f_i(E_i) = \frac{1}{E_i}$$

Gupta and Das [118] propose to define three levels of nodes: Nodes which dispose of less than 10 percent of their initial battery capacity are to be avoided whenever there is an alternative path. If an node has left 10-20 percent of battery capacity, it should not be used unnecessarily. Otherwise, a node is not treated specially.

Minimal Maximum Battery Cost Routing

Let $f_i(t)$ be a battery cost function for host n_i and $E_i(t)$ the residual battery capacity at a given moment. The less energy remains in a node, the higher the cost function of this node should be. Singh et al. [113] propose to use $1/E_i(t)$ as cost function. Their Min-Max Battery Cost Routing (MMBCR) metric chooses the path with the least maximal such cost function. In other words, let r_o be the chosen path and r_* the set of all possible paths. Then the chosen path fulfils

$$Cost(R_o) = \min_{r_j \in r_*} \max_{\forall n_i \in r_j} f_i(t)$$

On one hand, MMBCR considers the weakest node over a path and thereby provides a balanced energy load. On the other hand, there is no guarantee that MMBCR minimises the total energy consumed over a path.

Combining Battery Capacity with Energy per Packet

Toh [119] merges MTPR and MMBCR into one single strategy called Conditional Max-Min Battery Capacity Routing (CMMBCR). First, CMMBCR searches paths using MTPR, with the restriction that all nodes need to have a remaining battery capacity that exceeds a threshold value γ. If there is no such path, MMBCR is used.

Chang and Tassulias [120] formulate an algorithm that takes into account remaining battery capacity and necessary transmission energy for their Maximum Residual Energy Path (MREP) algorithm. Let $e_{i,j}$ be the energy consumed to send one packet over the link from node i to node j and E_j be the residual energy at node j. Chang and Tassulias define two metrics for the link from i to j:

The remaining energy of a node:

$$d_{i,j} = \frac{1}{E_j - e_{i,j}}$$

and the residual capacity of a node in terms of packets that can be delivered with the remaining energy:

$$d_{i,j} = \frac{e_{i,j}}{E_j}$$

In a simulation with dynamically moving nodes, both metrics came quite close to a theoretically predicted average node lifetime[9]. However, Chang and Tassulias did not compare these two metrics to the hop count metric. This would have made it easier to estimate the benefit of their metrics.

Refining their work in [116], Chang and Tassulias proposed a more general formula:

$$d_{i,j} = e_{i,j}^{x_1} {E_j}^{-x_2} {E_i}^{x_3}$$

where x_1, x_2 and x_3 are adjustable parameters. In a simulation, Chang and Tassulias showed that with reasonable parameters, the theoretical maximal values for average lifetime, worst-case lifetime and transfer reliability can be reached.

A very similar metric is described by Michail and Ephremides [115]. However, they do not multiply the terms for transmission energy and residual battery capacity, but they add the two terms. We suppose that the effects are comparable.

Kim et al. [111] compare MTPR, MMBCR and CMMBCR. Their first finding was that considering overhearing does have a significant impact on the results of simulations. Further, they found that the intuitive speculation that MTPR is appropriate in dense networks (the overall energy consumption is minimised, redundant paths can be found easier), whereas it is more important to avoid network partition in sparse networks (and thus, MMBCR performs better). It would be interesting to compare those metrics to simple hop count, especially when overhearing is taken into account.

[9] The maximal values were calculated using linear programming methods.

4.5.4 Other Energy-Based Metrics

Battery Prevention

Chiasserini and Rao [121] proposed a metric that takes into account the physical properties of batteries. This metric favours bursty traffic over constant traffic in order to optimise the long-term recharging ability of the battery. Although this is an interesting approach for some specific sensor network applications, bursty traffic should be avoided usually.

Packet Loss Ratio

Some authors propose to enhance energy-aware routing with packet loss information. Banerjee and Misra [122; 123] use the following metric:

$$d_{i,j} = \frac{e_{i,j}}{1 - p_{i,j}}$$

with $p_{i,j}$ being the packet loss ratio on the link from node i to node j and $e_{i,j}$ being the energy needed to transfer a packet from i to j. This formula is useful if packets are retransmitted from node to node. It does not hold if the utilized transfer protocol supports only end-to-end retransmissions. We will take up this idea in Section 7.3.1.

Interference

Michail and Ephremides [115] combine energy aspects with the attempt to reduce blocking of transmissions due to interference. Their Power and Interference-based Metric (PIM) is defined as:

$$d_{i,j} = \frac{e_{i,j}}{e_{max}} + \frac{|B_{i,j}|}{|B|}$$

where e_{max} is the pre-determined maximal value for transmission energy, $|B_{i,j}|$ is the number of links that are blocked due to interference when node i sends a message to j. $|B|$ is the overall number of links in the network.

4.6 Other Concepts

4.6.1 Security

The enforcement of security and cooperation is a basic need of WMNs. Often, game theoretic approaches are proposed for this aim. For example, Michiardi and Molva [124] propose a frameworks, which penalises non-cooperation with bad reputation. This bad reputation can be mirrored with a metric. Similar models can be found in [125] and in [126].

Another security related metric is mentioned in [26]: What is the average response time of the network administrator in case of a problem? Of course, this would be rather absurd for MANETs and WMNs.

4.6.2 Billing

In some networks, there might be links that can be used for free (e.g. open WLAN access points) and other links that are billed for (e.g. GSM). It is clear that cheaper links are preferred over more expensive links. This can be modelled using metrics as well.

4.6.3 Application-Specific Metrics

It is thinkable to construct metrics basing on information provided by the application layer. With this concept, task similar to the applications of overlay networks (see Andersen et al. [33]) could be devised. To the best of the authors's knowledge, this has not been investigated up to now.

4.6.4 MTU

The Maximum Transmission Unit (MTU) refers to the largest size of a packet a link can pass onwards. In most Ethernet LANs, the MTU is 1500 bytes. The MTU of a path is the minimal value of the MTUs of the links the path consists of. If IP packets with larger size are sent over this path, they are fragmented.

4.6.5 Node Resources

Metrics can be devised from the information on the capability of a node. Such capabilities can be the available memory of a node, the available processing resources or the number of interfaces. In WMNs, the most central property of a node is to which of the three classes (gateway, router and mobile node) it belongs.

4.6.6 Learning Networks

A completely different approach to assign link costs is intelligent learning. Barolli et al. [127] propose genetic learning algorithms. In fact, a network can be considered as an artificial neural network. The link weights represent the weights of the neural networks. A global target function for the optimisation of the network can be chosen according to the needs of the network. Thus, any kind of learning technique for neural networks can be used to choose optimal weights for each link. Güneş [128] describes a similar routing algorithm that bases on the methods that ants used to find optimal paths from the ant-hill to their nourishment sources.

At the moment, all theses techniques have not yet been tested in practise. However, we assume that they are only suited for very static networks.

Chapter 5

Metrics in Field-Based Routing

Most of today's routing algorithms either base on a hierarchical, more or less static, naming scheme or they do not scale well to big networks. Field-Based Routing (FBR) is a promising concept for Wireless Mesh Networks (WMNs) to achieve scalability without resorting to externally imposed hierarchy. A potential field based algorithm to route around congested parts of networks was proposed by Basu et al. [129]. Later, the idea was taken up by Lenders and Baumann at the Communication Systems Group at ETH Zurich who used the concept to propose local, proactive anycast routing algorithms for the use in MANETs and WMNs.

Most MANET routing algorithms base on the assumption that in a network, full many-to-many unicast communication must be provided. In a WMN, however, the main challenge consists in finding a way from a node, that wants to send a packet, to a gateway that will route this packet into the Internet. This task corresponds to an anycast problem with one single service – a task which Field-Based Routing is perfectly suited for. The backward path from the Internet to a node can be determined without big effort using source routing techniques.

In this chapter, we will shortly present the basic concept of two such FBR algorithms. Both algorithms base on hop count metrics solely and do not distinguish between metrics with different combination operator. We will propose two straightforward methods to enhance these algorithms with more sophisticated metrics. Then, we will show that these techniques still suffer from serious drawbacks. In Sections 5.4 and 5.5, we will propose several algorithms which overcome these problems.

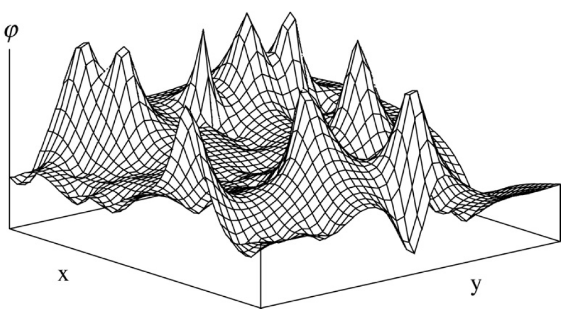


Figure 5.1: Example of a potential field with 10 gateways. (Figure adopted from [6])

5.1 Terminology

Field-Based Routing algorithms can be split into two parts: The *field construction* defines how the nodes compute their potential. *Packet forwarding* determines how links are chosen for sending packets to the Internet and how the forwarding is performed. Field construction can be further subdivided into *field propagation mechanism*, which describes how information about the field potential is spread across the network, and *field calculation mechanism*. The latter defines how the potential of each node is computed. Packet forwarding consists of *route selection mechanism* and *packet relaying*. While the former determines - based on the potential value and other variables - to which neighbour a node will forward packets, packet relaying describes how the forwarding actually is done. In this chapter, we will concentrate mainly on the field calculation mechanisms and the route selection mechanism.

5.2 Existing FBR Algorithms

In this section, we will shortly describe the two existing Field-Based Routing algorithms. For a more detailed description of the two routing protocols, please refer to the respective papers.

5.2.1 Lenders' Algorithm

This algorithm was proposed by Lenders et al. [6] and earlier in [7]. It alludes highly on the idea of electrostatic fields: Each gateway is associated with a charge Q, which represents its capacity to transfer traffic to the Internet. In regular intervals, the gateway floods the network with information about its capacity. The farther away a node is located (in terms of number of hops), the lower is the contribution of this charge to the potential of this node. In order to compute its potential, each node sums up the contributions of all gateways it got notice. Hence, all nodes in the network will be associated with a potential value and can calculate the potential values of their neighbours. More formally, the potential φ of a node n is defined as

$$\varphi(n) = \sum_{\text{all gateways j}} \frac{Q_j}{|n - n_j|}$$

where Q_j is the capacity of a gateway and $|n - n_j|$ the distance (measured in minimal number of hops) between node n and gateway j. When a node wants to send a packet, it forwards it to the neighbour with the highest potential (see Figure 5.2).

While this algorithm stands out through its straightforwardness and perceivability, there is an inherent flaw in this concept: Due to superposition of the fields of multiple gateways, it can happen that local field maxima emerge at ordinary nodes. As a result, packets sent to these nodes would be stuck in this node and not be forwarded to an Internet gateway. However, this problem can be resolved: Each node can detect if it represents a local maximum. If this is the case, packets can be forwarded directly to a gateway in the vicinity. Yet, this method may lead to routes which are considerably longer compared to a shortest path.

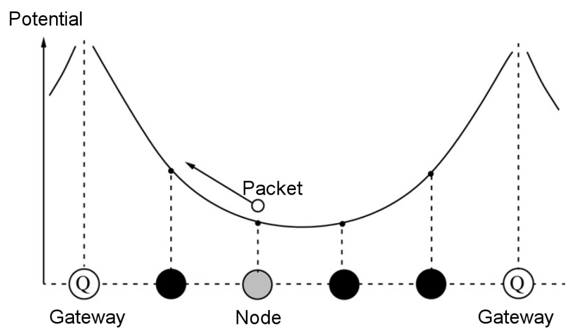


Figure 5.2: Lenders' Algorithm: The packet sent by the node is routed in the direction with the steepest gradient. (Figure adopted from [6])

5.2.2 Baumann's Algorithm

Baumann et al. [4; 5] proposed an algorithm which approaches some of the problems that are inherent to other protocols. Just like it is characteristical for Field-Based Routing protocols, Baumann's algorithm postulates that gateways are assigned with a value representing their capacity to forward packets to the Internet. In contrast to Lenders' algorithm, this information is not flooded throughout large parts of the network. Instead, each node calculates its potential value only from the potential values of its direct neighbours. This calculation is described in Algorithm 1.

Algorithm 1 Field calculation function of Baumann's algorithm

```
Sort x_i in ascending order
l = 0
f(0) = 0
while f(l) < x_i do
    f(l + 1) = f(l) + (x_i − f(l)) · κ
    l = l + 1
    Go to next node in sorted list
end while
f_final = f(l)
```

In this listening, the input values x_i represent the potential values of the neighbours i of the node which calculates its potential. An example of this calculation is shown in Figure 5.3.

Baumann shows that the problem of local maxima can be avoided using this algorithm. The conductivity parameter $1 > \kappa > 0$ controls to which extent the density of a region will influence the potential values of the nodes. A κ close to 1 results in shortest path routing (in terms of hops), while the redundancy of a node gains in importance if κ is lowered.

An alternative to this function is presented in Appendix B.

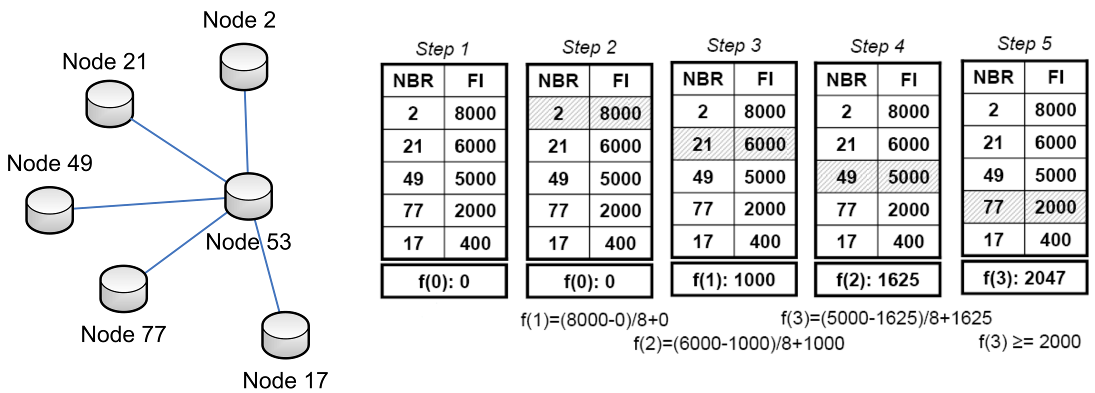


Figure 5.3: Example of Baumann's algorithm with a conductivity value of $\kappa = 1/8$. Step 1: Sort neighbours by field intensity; step 2-5 iterate down the table until the potential of the node gets higher or equal to the next neighbour; node 71 and 17 do not contribute to the potential of node 53. (Figure adopted from [5].)

5.3 Proposed FBR Algorithms: Enhancements

In the following, we will present enhancements which allow the two FBR algorithms presented in the previous section to integrate metrics. Subsequently, we will shortly analyse theses algorithms.

5.3.1 Enhancement of Lenders' Algorithm

In Lenders' algorithm, the quality of a link is completely left aside. Only the number of hops contributes to the potential value of a node. This can lead to non-optimal routing decisions, as the example in Figure 5.4 shows.

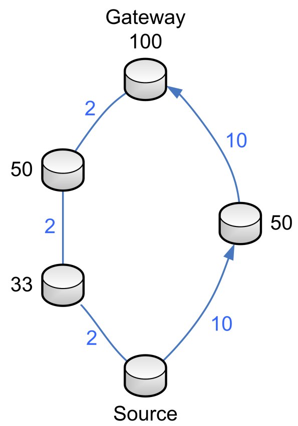


Figure 5.4: Example for a non-optimal path with Lenders' Algorithm: The path to the right is chosen, although the connections on the left are clearly better. (Link weights are written in blue, node potentials in black.)

A first approach to eliminate this problem is to include the metric into the computation of the field potential, as Lenders describes in his paper [6]. This is expressed in the formula

$$\varphi_2(n) = \sum_{\text{all gateways j}} \frac{Q_j}{weight_{sp,j}}$$

with $weight_{sp,j}$ being the cost to send a packet to the gateway j on the shortest path. In Figure 5.5, an example is given that shows how this enhancement leads to better routing decisions.

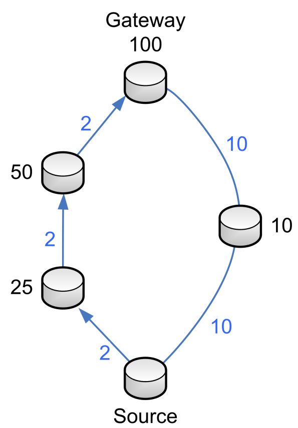


Figure 5.5: Enhanced Lenders' Algorithm: Now, the path to the left is chosen, because the field potential decreases more steeply on the low-quality links of the path to the right. (Link weights are written in blue, node potentials in black.)

A More Radical Approach

We estimated that the value of $\varphi_2(n)$ can be approximated roughly by the formula

$$\varphi_3(n) = \max_{\text{all neighbours(n)}} \frac{\varphi(m)}{d(n,m)}$$

where $\varphi(m)$ is the potential value of the neighbour and $d(n,m)$ stands for the metric value of the link between the neighbour and node n. Many variations of this algorithm are possible, e.g. not to take the maximum value of all neighbours, but taking a (weighted) sum of the potentials of the neighbours. However, we found that this is a fundamentally different class of algorithms. We will look deeper into this topic in Section 5.5.4.

5.3.2 Enhancement of Baumann's Algorithm

In order to incorporate metrics into Baumann's algorithm, we propose to replace the κ parameter by the metric of the corresponding link. Concretely, we propose to change line 5 of Baumann's algorithm as described in Algorithm 2. The algorithm requires that the metric values $d_{i,j}$ of the link between node j and its neighbour i are normalised to the interval $(0, 1)$.

Algorithm 2 Field calculation function of the enhanced Baumann algorithm

1: Sort x_i in ascending order
2: $l = 0$
3: $f(0) = 0$
4: **while** $f(l) < x_i$ **do**
5: $\quad f(l+1) = f(l) + (x_i - f(l)) \cdot d_{i,j}$
6: $\quad l = l + 1$
7: $\quad$ Go to next node in sorted list
8: **end while**
9: $f_{final} = f(l)$

Like this, the contribution to the node potential by a neighbour is weighted by the quality of the link that connects the node with this neighbour. The progress observed at the enhancement of Lenders' algorithm (see example in Figure 5.5) can be transferred analogically to the enhancement of Baumann's algorithm.

5.3.3 Analysis

While the enhanced versions of Baumann's and Lenders' show improvements in some situations, they still fail in others (as for example in the setting in Figure 5.6). This results from the fact that the algorithms only take into account the potential values of the neighbours. The links that are directly connected to a node, however, do not influence the routing decision. In the following sections, we will present algorithms that eliminate this challenge.

5.4 Proposed FBR Algorithms: A Steepest Gradient Approach

In current FBR algorithms, the route selection mechanism chooses the link which leads to the neighbour with the highest potential. In order to include links that are directly connected to a node, we go back to the notion of metrics as a distance. We propose that the forwarding function should select the link that has the steepest gradient. In analogy to plane geometry, where the gradient is defined as $\frac{\text{height}}{\text{distance}}$, we define the gradient of a link as $\frac{\text{higher potential of the two nodes}}{\text{metric value}}$ (see Figure 5.7).

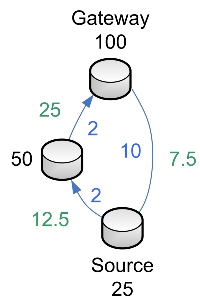


Figure 5.6: The enhanced versions of Baumann's and Lenders' algorithms both prefer the low-quality link on the right. (Link weights are written in blue, node potentials in black.)

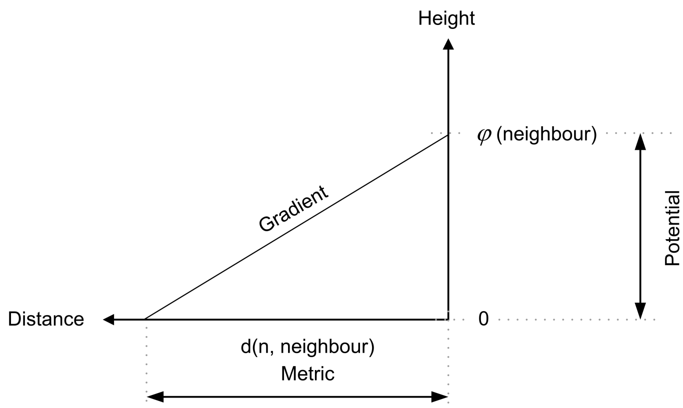


Figure 5.7: The gradient of a node n to one of its neighbours.

This forwarding function can be used in all possible FBR algorithms in principle. Figure 5.8 shows an example in which it is used in combination with Baumann's FBR algorithm. First large scale simulations indicate performance improvements of Baumann's algorithm, when it is combined with the steepest gradient approach [130].

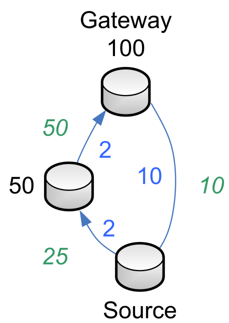


Figure 5.8: Example for the steepest forwarding function: Contrary to the previous algorithms, the path with more hops, but better quality links is chosen. (Link weights are written in blue, node potentials in black, gradient values in green/italics.)

It was considered to use the potential difference of the two nodes as height instead of the maximum of the two node potentials i.e. to use $\frac{\text{potential difference}}{\text{metric value}}$ as gradient formula. However, this idea fails in some situations: When there is an excellent link, the potential of the two adjoining nodes is similar by definition. This results in a flat gradient – although this link might be a good choice (see example in Figure 5.9).

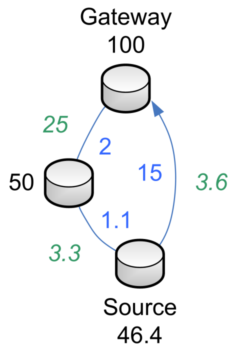


Figure 5.9: Using the potential difference as gradient function: Although the path to the left is clearly better, the low quality link to the right is chosen. (Link weights are written in blue, node potentials in black, gradient values in green/italics.)

5.5 Mathematical Models for the Integration of Different Types of Metrics

In this section, we propose another three algorithms; one for additive metrics, one for multiplicative metrics and one for metrics that use the minimum function as concatenation operator. The algorithms are based on the considerations presented in Section 5.5.4. The class of algorithms proposed in this section combines several features which – to the best of our knowledge – no other algorithm offers:

- The algorithms provide perfect scalability as only local information is needed.
- All kinds of metrics can be incorporated, independently from their concatenation operator.
- The redundancy of a path will influence the routing decisions. In fact, its influence is controlled by an adjustable parameter (see Section 5.5.3).
- Local maxima in the potential field cannot occur with any of the algorithms.

5.5.1 Proposed Algorithms for Different Types of Metrics

All three algorithms base on Baumann's algorithm (see Section 5.2.2). Even the parameter κ keeps its original role (see Section 5.5.3). The sole change concerns the input parameter x_i: While Baumann's algorithm uses the potential of its neighbours as input values, each of the proposed algorithm modifies this value in another fashion. Using the terminology described in Section 5.5.4, the three algorithms apply Baumann's algorithm as aggregation function and each has a peculiar concatenation function. Below, we describe how the algorithms modify the input parameter x_i. $d_{i,j}$ stands for the metric value of the link between node j and its neighbour i; $\varphi(\cdot)$ is the potential.

1. **Additive metrics:** $x_i = \varphi(\text{Neighbour i}) - d_{i,j}$[1]. When this algorithm is used, all metric values must comply with $d_{i,j} > 0$ in order to guarantee strict monotony of the potential field and, thus, convergence. An example situation with this algorithm is given in Figure 5.11.

[1] If the algorithm is implemented carefully, especially concerning initialisation values, it is no problem if the potential value drops below zero.

2. **Multiplicative metrics:** $x_i = \frac{\varphi(\text{Neighbour i})}{d_{i,j}}$. When this algorithm is used, all metric values must comply with $1 > d_{i,j} > 0$ in order to guarantee strict monotony of the potential field. Otherwise, the metrics must be normalised accordingly. An example situation with this algorithm is given in Figure 5.12.

3. **Concave metrics:** $x_i = \min(\varphi(\text{Neighbour i}), d_{i,j}) - \epsilon$. The constant ϵ is set to the smallest possible value. It guarantees strict monotony of the potential function. An example situation with this algorithm is given in Figure 5.13.

The entire algorithm for additive metrics is specified in Algorithm 3. Multiplicative and concave metrics are calculated analogically.

Algorithm 3 Proposed field calculation function for additive metrics

for all neighbours i **do**
    $x_i = \varphi(\text{i}) - d_{i,j}$
end for
Sort x_i in ascending order
$l = 0$
$f(0) = 0$
while $f(l) < x_i$ **do**
    $f(l+1) = f(l) + (x_i - f(l)) \cdot \kappa$
    $l = l + 1$
    Go to next node in sorted list
end while
$f_{final} = f(l)$

Route Selection Function

The route selection mechanism (see also Algorithm 4) is structured similarly for all three algorithms: For each neighbour i, node j calculates the potential value "as it observes it". These terms are defined as follows:

1. **Additive metrics:** $p_i = \varphi(\text{Neighbour i}) - d_{i,j}$

2. **Multiplicative metrics:** $p_i = \frac{\varphi(\text{Neighbour i})}{d_{i,j}}$.

3. **Concave metrics:** $p_i = \min(\varphi(\text{Neighbour i}), d_{i,j})$

The nodes forwards packets which are destined for the gateway to the neighbour with the highest p_i. We remark that these terms are the same as the concatenation function of the three algorithms. We also realise that the p_i of the multiplicative algorithm coincides exactly with the steepest gradient approach presented in Section 5.4.

Algorithm 4 Route selection function for additive metrics

for all neighbours i **do**
    $p_i = \varphi(\text{Neighbour i}) - d_{i,j}$
    if $p_i > \max$ **then**
        $\max = p_i$
        best_link = i
    end if
end for
return best_link

5.5.2 Relation of Potentials and Metric Values

In this section, we will shortly look at the problem of choosing appropriate potentials for gateways. Again, this depends on the type of metric that is in use.

1. **Additive metrics:** The absolute height of the gateway potentials is not relevant for the routing algorithm. However, the differences of the gateway potentials determine to which route a packet will be forwarded. In fact, potential differences are weighted equivalently as path metrics (see example in Figure 5.10).

2. **Multiplicative metrics:** The gateway potentials have a similar function as for additive metrics.

3. **Concave metrics:** When the minimum function is used, the potential of gateways is handled like an additional link. The potential is compared to the values of the links that are connected to them using $\min(\varphi(\text{Neighbour i}), d_{i,j})$. This means that the gateway potential has an effect on the field only if it is smaller than one of the links that are connected to it. This makes perfect sense e.g. for the bandwidth metric: The potential value can be regarded as the bottleneck bandwidth to the Internet.

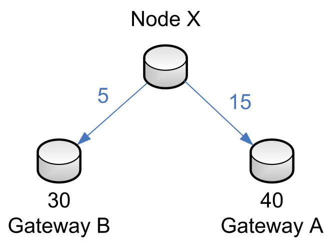


Figure 5.10: Relation of potential and metrics: Node X wants to send a packet to the Internet. Gateway A has the higher potential, but gateway B is connected through a better link. Both routes are considered equivalent. (Link weights are written in blue, node potentials in black.)

5.5.3 Analysis

Tests and simulations of these algorithms have been done on a limited scale. They showed that these algorithms can be implemented straightforwardly and produce correct results. Moreover, observations indicate that a κ of 1 causes the algorithms to act like an algorithm for finding optimal paths based on the link weights[2]. For additive metrics, the respective algorithm will find the same path like Dijkstra's algorithm - yet, instead of having to know the state of all nodes in the network, our algorithms only require knowledge of their direct neighbours. This is rendered possible due to the fact that we do not need many-to-many communication but an anycast routing algorithm. The algorithm for concave metrics will choose the path with the largest minimal bandwidth. In fact, it will even select the Shortest Widest Path (see Section 4.3.4). Similarly, the multiplicative algorithm finds an optimal path when κ is set to 1.

When κ is lowered, redundancy becomes – analogically to Baumann's algorithm – more important. When we compare a node that is linked to one gateway with a node with connections to two gateways, the redundancy of a node has the highest influence on the potential value of the node at a value of $\kappa = 0.5$ (see Figure 5.14). However, when we increase the number of neighbouring gateways, i.e. we boost redundancy, the highest influence of this redundancy onto the potential value is observed at values of κ significantly below 0.5 (see Figure 5.15).

Investigation on these algorithms should be continued, in our opinion (see Section 7.3, Future Work).

[2] This is valid if there is only one gateway in the network. When there is more than one gateway, potential differences have the same effect like additional path lengths (see Section 5.5.2).

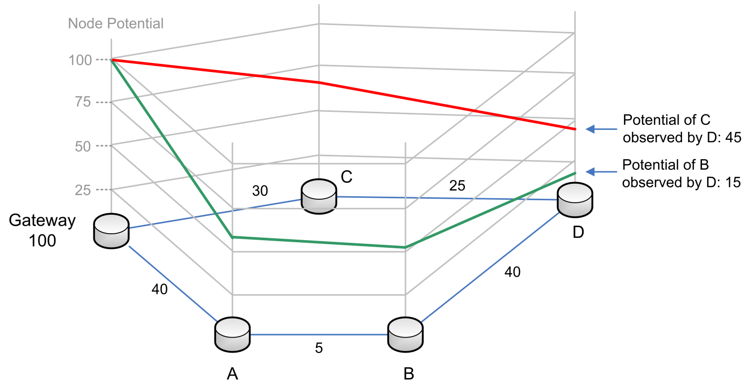


Figure 5.11: Behaviour of the potential function using the proposed algorithm for additive metrics with κ set to 1: D will send packets to the gateway via C; D's potential will be calculated as 45.

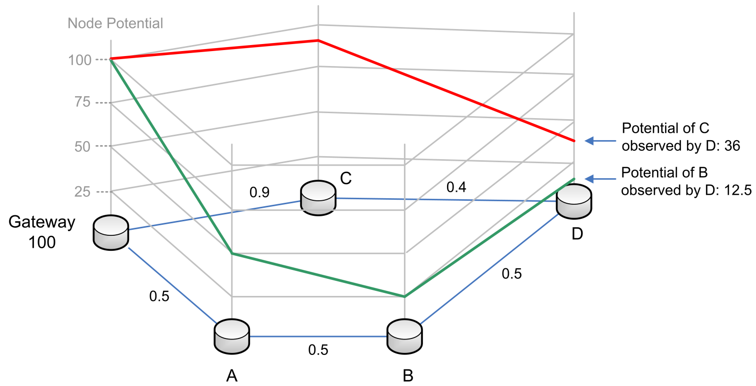


Figure 5.12: Behaviour of the potential function using the proposed algorithm for multiplicative metrics with κ set to 1: D will send packets to the gateway via C; D's potential will be calculated as 36.

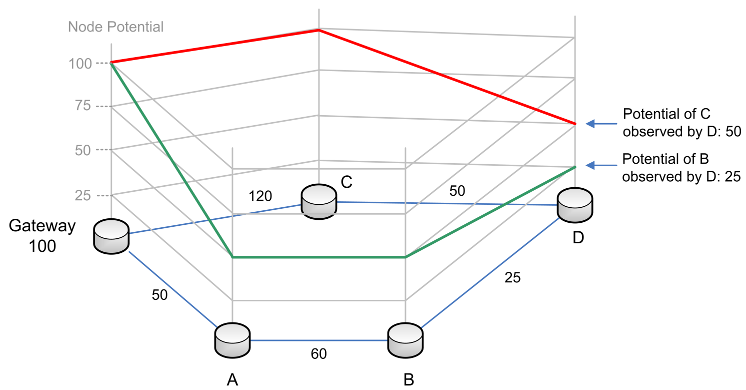


Figure 5.13: Behaviour of the potential function using the proposed algorithm for minimum metrics with κ set to 1: D will send packets to the gateway via C; D's potential will be calculated as 50.

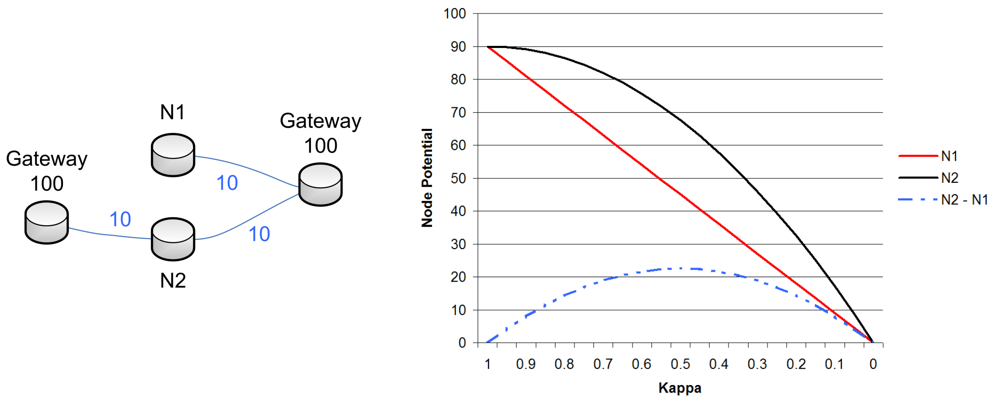


Figure 5.14: Additive FBR: Influence of κ on the potential of nodes, measured on the example on the left. With $\kappa = 1$, redundancy does not play any role. The highest impact of redundancy is observed with $\kappa = 0.5$.

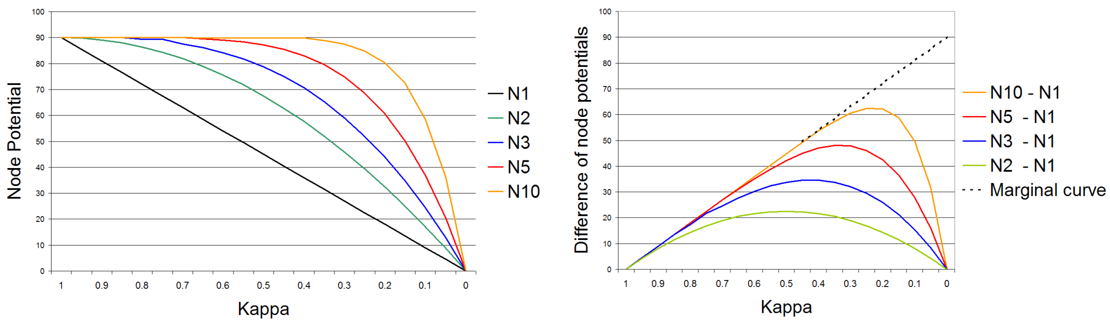


Figure 5.15: Additive FBR: Influence of κ on the potential of nodes, measured analogically to Figure 5.14, but with nodes that are connected to 1, 2, 3, 5 and 10 gateways. On the left: Node potentials with different κ values. The more neighbours a node has, the higher the κ must be set until its potential value is lowered. On the right: Difference of the node potentials of nodes with links to 2 / 3 / 5 / 10 gateways to a node that is connected to one gateway. When there are more gateways, redundancy influences the potential value more with lower κ's. When the number of gateways tends to ∞, redundancy has most influence with $\kappa = 0$.

5.5.4 Derivation and General Reflections

In this section, we will show that many aspects of Field-Based Routing can be analysed using the algebraic description of metrics, which we discussed in Section 3.3.4, as a model. We did derive the three algorithms in the previous section basing on these considerations.

Before we generalise the concept of metrics in Field-Based Routing, we will have a look at a special situation: A linear concatenation of nodes with one gateway (Figure 5.16)

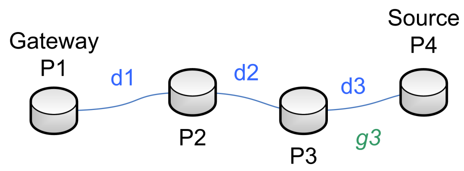


Figure 5.16: A special situation: Concatenation of nodes (Link weights are written in blue, node potentials in black, gradient values in green/italics.)

We suppose now a FBR algorithm that calculates the field potential of a link as $\frac{P_x}{d_x}$ when there is only one neighbour. This is the case for the adapted versions of Lenders' and Baumann's[3] algorithms. The analytical calculation of the gradient (according to the steepest gradient calculation in Section 5.4) of the link between P_3 and P_4 results in

$$P_2 = \frac{P_1}{d_1}$$

$$P_3 = \frac{P_2}{d_2} = \frac{P_1}{d_1 \cdot d_2}$$

$$g_3 = \frac{P_3}{d_3} = \frac{P_1}{d_1 \cdot d_2 \cdot d_3}$$

Concatenation

More generally, the gradient g at the end of a chain of links that starts at a gateway can be expressed as

$$g = Q_{GW} \cdot \prod_{\text{all links i}} \frac{1}{d_i}$$

with Q_{GW} being the capacity of the gateway. We can see that the gradient of the link is nothing else than the reciprocal of the multiplicative concatenation combination of the links, as it is described in Section 3.3.4. It is not difficult to devise this idea further with the additive and minimum/maximum operators: In the additive case, the routing algorithm does not calculate the potential by a division, but by a subtraction. The value upon which the routing decision based – the steepest gradient – does not need to be calculated as steepest gradient of the adjacent links, i.e. as division, but again as subtraction. In the same situation as described above, the "gradient" is calculated as

$$g = Q_{GW} + \sum_{\text{all links i}} -d_i$$

An example situation with an additive concatenation operator is shown in Figure 5.17. Similar considerations can be done for minimum and maximum operators.

We realise that what we called "gradient" before should rather be interpreted as the quality of a link as it is observed by a node, or more precisely: The quality of a link and the paths to which it is connected.

[3] This is valid if we consider the normalised metrics in the interval $(0, 1)$ as reciprocals of the real metric values.

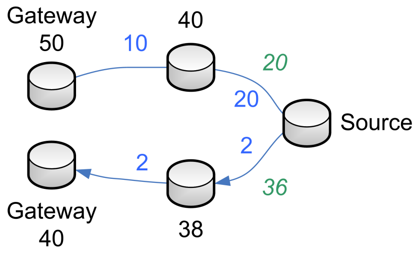


Figure 5.17: Additive version of FBR. (Link weights are written in blue, node potentials in black, "gradient" values in green/italics.)

Aggregation

When more than one neighbour is present, the combination of potential values from different neighbours plays the same role as the aggregation operator of metrics. For example, the radically adapted version of Lenders' algorithm in Section 5.3.1 uses the maximum function as aggregation operator. Baumann's algorithm uses a more sophisticated function which is described in Section 5.2.2. Lenders' algorithm, however, does not fit into this model because the potential is not calculated only using the potential of the neighbours, but with the previous knowledge of the shortest paths to the gateways in its vicinity.

The problem of local maxima can be approached by choosing an appropriate aggregation operator. One option is to use a maximum[4] or minimum function as operator, another option is used in Baumann's algorithm.

As example for the meaning of concatenation and and aggregation: The "gradient" in Figure 5.18 can be expressed in the algebraic form proposed in Section 3.3.4:

$$g_6 = d_5^{-1} + (P_1 \cdot (d_1 + d_2)^{-1} \parallel P_3 \cdot (d_3 + d_4)^{-1})$$

where $+$ is an addition, a multiplication, or the maximum; $\parallel$ is any aggregation operator, e.g. Baumann's algorithm.

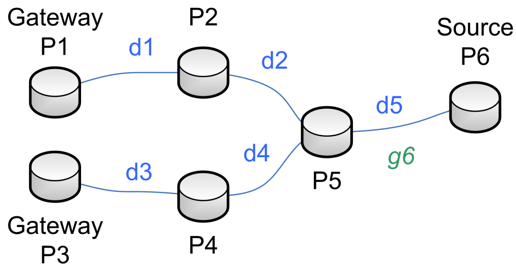


Figure 5.18: Analytical determination of the "gradient": Concatenation and aggregation of nodes (Link weights are written in blue, node potentials in black, gradient values in green/italics.)

Summary

We propose a class of Field-Based Routing algorithms which is characterised by three properties: (1) The potential of each node depends only on the potential of its neighbours and the links that

[4] We claim without prove that when the maximum is taken as aggregation operator, FBR is a distributed algorithm for finding optimal paths (be it under additive, multiplicative or minimum/maximum operators) from each node to a gateway. If there is more than one gateway, but all having the same capacity, the path to the gateway is chosen which carries the least path cost. If gateways have different capacities, these values can be thought of as weights that are given to the paths starting at these gateways.

connect them. (2) The algorithm can be subdivided into two components: One calculates how single links are added (concatenation), the other describes how these single links are aggregated (aggregation). (3) The aggregation operator provides that no local maxima can originate. The three algorithms proposed in Section 5.5.1 are representatives of this class.

Chapter 6

An FBR Simulation Tool

When prototyping, analysing and comparing different routing algorithms, a simulation tool is an indispensable aid. In the case of Field-Based Routing (FBR), it is central to study the behaviour of the field calculation and the route selection functions. As no program suitable to our needs had been available, we designed and implemented such a tool.

We will first describe the requirements that we defined. In Section 6.2, we will briefly describe the programming framework we used. Later in the same section, we will present the features that we have implemented as well as a number of screen shots of the tool.

6.1 Requirements

The goals of the tool are the following: It should provide means to create, manipulate, save and load abstract network graphs. It should calculate the field values of all nodes in a given network when different field calculation and route selection functions are applied. In addition, it should be easily extendible to other algorithms. And it should visualise the networks and results of the simulation.

6.2 Design and Implementation

In this section, the basic features of our FBR Simulation Tool are described. A user guide and a more extensive implementation overview can be found in Appendices C and D, respectively.

6.2.1 Programming Framework

Several options for an implementation framework for the tool were considered, among them full-scale network simulator environments such as ns-2, OpNet and GloMoSim. As we estimated that these applications overshot our needs, we decided to use the *Java Universal Network/Graph Framework JUNG* [131] as basis for the simulation tool. This JAVA software library provides an API for the modelling and visualization of a graphs and networks. It showed to be suited excellently for the visualisation of the simulation results.

6.2.2 Algorithms

Currently, six different FBR algorithms are implemented in the tool: (1) Baumann's original FBR algorithm (see Section 5.2.2); (2) the adopted Lenders algorithm (see Section 5.3.1); (3) an algorithm that estimates the node failure likelihood based on link failure ratios; (4) a first algorithm which enhances Baumann's FBR algorithm (see Section 5.3.2); (5) Baumann's algorithm in combination with the steepest gradient approach (see Section 5.4); and (6) the additive version of the algorithms proposed in Section 5.5.1. Screen shots of three examples are shown in Figures 6.3 to 6.5.

6.2.3 Random Graphs

Besides the option to create and alter networks by hand, the FBR Simulation Tool offers the opportunity to generate random networks. The user specifies the number of ordinary nodes and gateway nodes as well as two parameters which determine how links between the nodes are generated (see Figure 6.1). Link weights are chosen randomly. Two methods to distribute the nodes are available:

True Random The nodes and gateways are distributed randomly on the whole simulation area.

Uniform Random The area is divided into rectangles of equal size. One node or gateway is placed at a random location within each of these rectangles (see Figure 6.2).

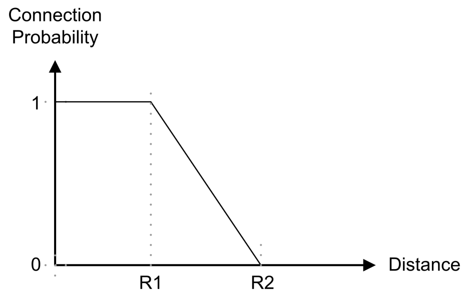


Figure 6.1: Graph generation: If two nodes are located at a distance less than R1, there will always be a link between them. At a distance greater than R2, no link will be created. At distances between R1 and R2, a link will be created with a likelihood that depends from the distance.

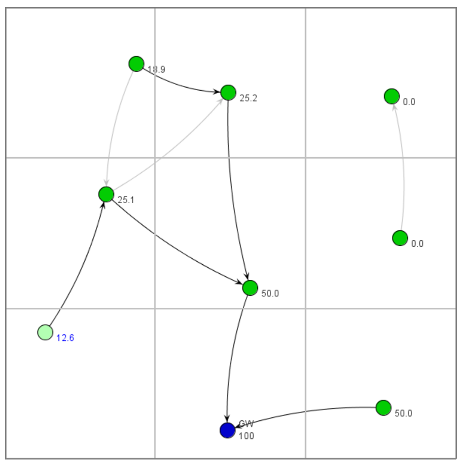


Figure 6.2: Pseudo-Random Distribution: Graph with 9 nodes in 3 rows and 3 columns.

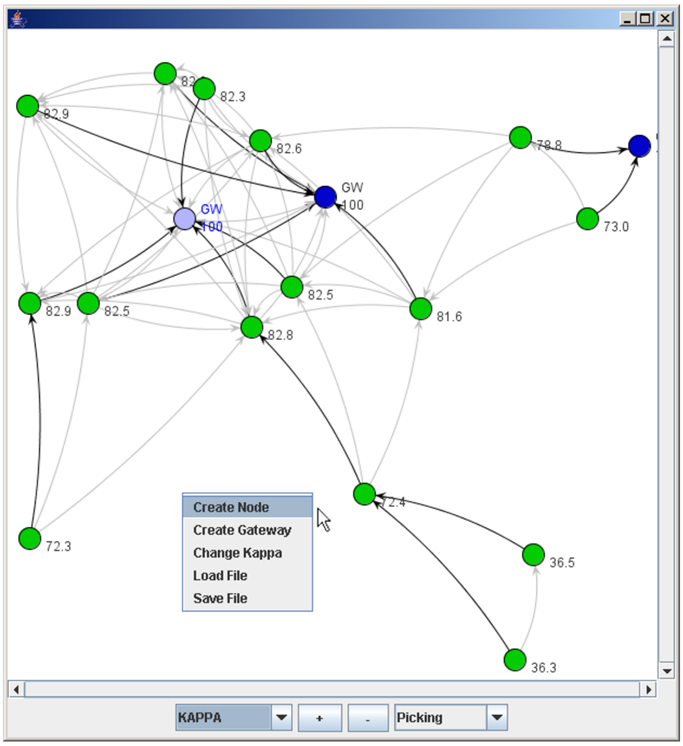


Figure 6.3: Screen shot: Baumann algorithm on a random graph

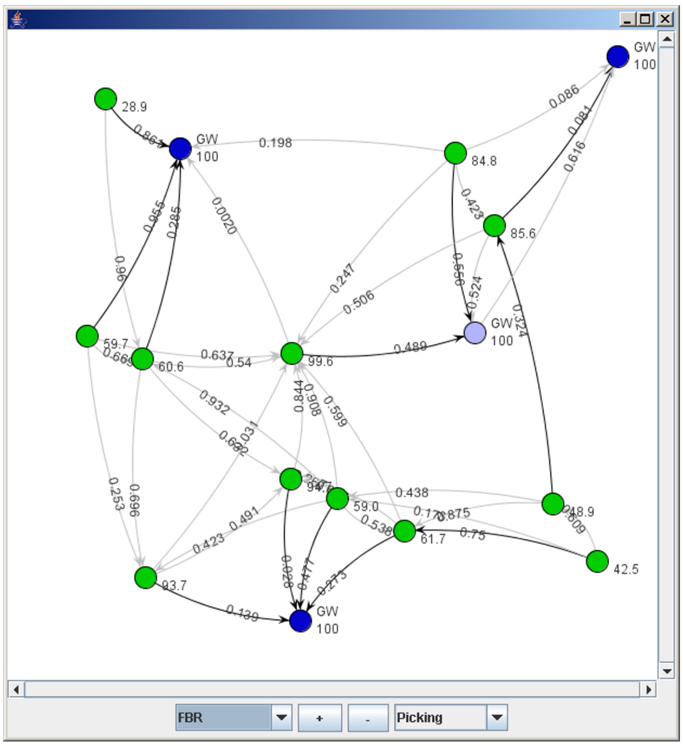


Figure 6.4: Screen shot: Adopted Lenders algorithm on a pseudo-random graph.

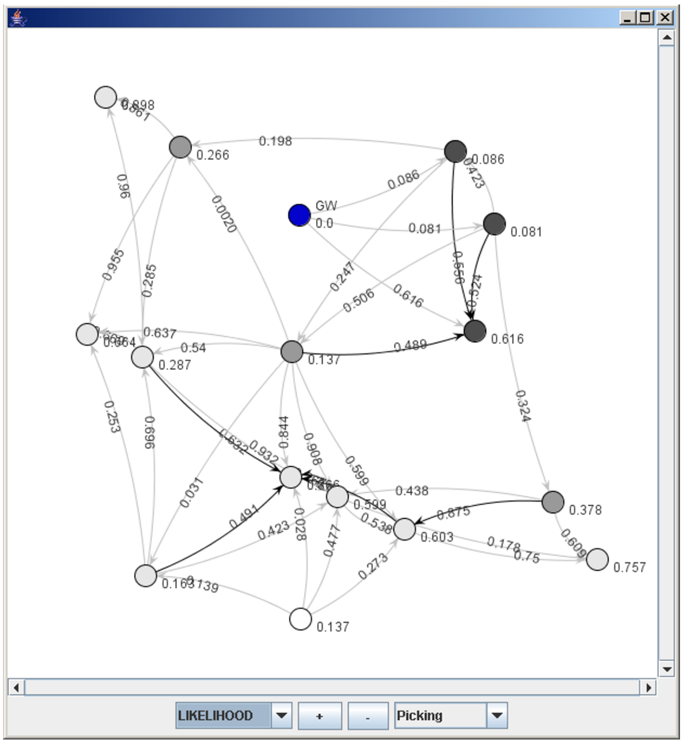


Figure 6.5: Screen shot: Node failure likelihood prediction algorithm on a pseudo-random graph.

Chapter 7

Conclusion

This chapter reports the conclusions we reached as results of this thesis. First, the main findings of the precedent chapters are shortly addressed. In the second section, we present two metrics that we consider suited best for certain standard situations. Section 7.3, finally, contains our recommendations for the next steps in the study of the topics treated in this thesis.

7.1 Results

The main findings and contributions of this thesis are the following:

1. **Mathematical Definition:** In Section 3.3.4, we propose an algebraic definition of routing metrics. We studied some existing approaches and enhanced them by features such as the possibility to aggregate links.
2. **Taxonomy:** In order to categorise and analyse metrics, we formulated a taxonomy which classifies routing metrics in five aspects: Factors that influence the value of the metric, mathematical properties, design perspective, implementation characteristics, and qualitative evaluation. The taxonomy is described in Section 3.4.
3. **Survey:** The most voluminous part of this thesis is an extensive survey of all routing metrics that we could find in literature. The survey in Chapter 4 contains around 61 metrics. These metrics are subsumed in 19 groups that are characterised according to the taxonomy mentioned above.
4. **Incorporation into Field-Based Routing:** In Chapter 5, we propose and analyse a number of methods how metrics can be incorporated into Field-Based Routing (FBR) algorithms.
5. **Simulation Tool:** In addition, we present a tool which allows to simulate and study the behaviour of different field calculation and route selection functions for FBR algorithms (see Chapter 6).

7.2 Recommended Metrics

It is not possible to define a "perfect metric". There are too many factors that influence the quality of a metric: The features and restrictions of the hardware platform in use; the environment into which a metric is deployed; the routing algorithm in which it is embedded; the characteristics of the traffic we assume; and, what is more, the design goals may be diverse. Additionally, in some cases, performance effects of a metric depend more on the quality of the implementation than on the metric itself.

Nevertheless, we dare to give two recommendations for two scenarios in the following section: One for the use in systems with restricted battery capacity, the other for general networks, where the goal is high throughput performance. We highlight that these recommendations are to be treated with prudence. They are proposed for good reasons, but they are neither tested in simulation nor in practise.

7.2.1 Energy-Restricted Networks

For energy restricted networks, we propose an adopted version of the metric proposed by Chang and Tassulias in their paper [116]:

$$d_{i,j} = (E\left[RTM_{i,j}\right] \cdot e_{i,j})^{x_1} E_j{}^{-x_2} E_i{}^{x_3}$$

where $e_{i,j}$ is the energy that the hardware will spend to transmit a packet from node i to node j as described in Section 4.5.1. E_i and E_j are the remaining battery capacities of i and j, respectively. The term $E\left[RTM_{i,j}\right])$ denotes the expected number of retransmissions necessary to transmit a packet successfully from i to j. It can be deduced e.g. from the packet loss ratio. x_1, x_2 and x_3 are adjustable parameters. In there work, Chang and Tassulias propose suitable values for these variables.

We chose this metric, because it reduces overall energy consumption and, at the same time, avoids that single nodes are stressed with exceedingly much traffic by incorporating battery power. We added the term for expected retransmissions in order to make the metric more apt for networks with fluctuating connection quality.

7.2.2 General Networks

In this case, we assume that there are no energy restrictions, or that there is an additional mechanism in the routing protocol that excludes nodes with critical battery load from relaying packets. Furthermore, we suppose that the design goal for the networks in this case is to maximise throughput.

Of course, the most straightforward approach to achieve high throughput would be to create a metric that predicts the available bandwidth for each link. Thus, the routing algorithm would be able to pick the route that promises the best available bandwidth. However, measuring the currently available bandwidth of a link represents a major problem, which no known technique can tackle satisfyingly.

A large number of effects influence the available bandwidth, among them mobility, interference, competing traffic and the properties of the used radio technology (see Figure 4.6). When devising a suitable metric, we aim at covering the same effects, but on a better measurable way. If we combine the packet loss ratio metric with the nominal bandwidth of a link, the same influences are considered as with the available bandwidth metric. Therefore, we recommend a combination of these two metrics for general networks:

$$d_{i,j} = mETX_{i,j} \cdot NomBW_{i,j}$$

where $NomBW_{i,j}$ is the nominal bandwidth between nodes i and j. For example, IEEE 802.11 standard provides around a dozen distinct transmission rates which are chosen adaptively based on the current Signal-to-Noise Ratio. Depending on the currently effective transmission rate, $NomBW_{i,j}$ would be chosen. Additionally, the number of free channels and available interfaces should contribute to the calculation of this value.

$mETX_{i,j}$ is an adoption of the ETX metric, which itself is a bidirectional version of the packet loss ratio metric. mETX (see Section 4.1.9) not only takes into account the Packet Loss Ratio of a link, but also its variance. Thus, it becomes more stable against self-interference effects. In order to include the congestion of nodes into the metric, we propose to measure the Packet Loss Ratio on the network layer. Like this, retransmission due to full router queues will have an effect onto the metric.

Like the metric proposed for energy-restricted networks, this metric contains some cross layer elements. Being aware that such approaches need to be treated with precaution [13], we believe that the possible benefits of such methods outweigh the risks clearly.

7.3 Future Work

In this section, we describe which topics we propose to pursue in future.

7.3.1 Metric Evaluation

The main dissatisfying point in the classification of metrics in this thesis is the lack of quantitative performance comparisons. As described in Section 3.4.5, our taxonomy only contains qualitative evaluation criteria. If, in future, the intention would arise to create a quantitative ranking of routing metrics, this would be only possible by extensive simulation or even real life tests.

We have abstained from conducting such simulations and tests for several reasons: (1) It is not possible simply to compare two metrics with each other. Simulation or test results will highly depend on the choice of various parameters. As mentioned in Section 7.2, among these parameters are the used routing protocol, the supposed environment and traffic characteristics as well as the chosen hardware platform. (2) The performance of metrics highly depends on the quality of the implementation. It would be difficult to eliminate such effects completely. (3) Simulations of Mobile Ad Hoc Networks (MANETs) always are tainted with the supposition of being imprecise or being transferable to practise only with reservations. Real life tests, on the other hand, are extremely laborious and costly. (4) Only a rather small number of metrics could be analysed thoroughly in a test or simulation. Already the selection of the metrics to be tested would necessarily have been subjective.

Therefore, we suggest evaluation comparisons by simulation or real life tests only if the usage of the designed system is already clearly defined. However, we would not recommend to try to compare metrics quantitatively in general.

7.3.2 EWMA Parameters

A minor topic for possible future work is mentioned in Section 4.1.2. It concerns finding good parameters for Exponential Weighting Moving Average (EWMA) calculation in MANETs or WMNs. This could be of interest e.g. for the packet loss ratio utilised in the metric proposed in Section 7.2.2.

7.3.3 Field-Based Routing

First large scale simulation simulations, which were conducted outside of this thesis [130], indicate that the performance of Baumann's algorithm can be improved when it is combined with a steepest gradient route selection function (Section 5.4). We recommend that similar simulations are undertaken with the algorithms proposed in Section 5.5.

Furthermore, we propose to study the influence of different κ values on the performance of networks with varying density patterns. We suspect that low κ's are a good choice in relatively mobile networks with a high variation in node quality, whereas κ's near 1 are suited for networks with differing link qualities but nodes that all have a similar number of neighbours.

7.3.4 Tool

Currently, the FBR Simulation Tool is programmed in a well understandable, but inefficient way. The recursive coding limits the number of nodes to a few dozens due to stack overflows. If the tool should be used for larger networks, we recommend to restructure the program code so that it calculates the node potentials iteratively.

Appendix A

Combination Operators

Metrics can be characterised by the mathematical operator that is used to compute the metric value of the concatenation or aggregation of links (see Section 3.3.4). In the following, an overview on the combination operators of various metrics is given.

A.1 Concatenation Operators

Metric	**Proposed Concatenation Operator**
Delay, RTT	Addition
Delay Variation	Addition or Addition and Covariance
Queue Length	Addition or Minimum
Bandwidth	Minimum
Packet Loss Ratio	Multiplication
Link Quality Metric (LQM)	Addition
Packet Reordering Ratio	Multiplication
ETX	Addition
mETX, ENT, EDR	Addition
ETT	Addition
WCETT, MCR	Addition and Maximum
MIC	Addition
PARMA	Addition
Signal Strength, SNR, Interference	Addition
MTM, ETT	Addition
Number of Neighbours	Addition
Number of Paths	Addition
Hop Count	Addition
Widest Shortest Path / Shortest Widest Path	Addition and Minimum
Geographical Distance	Addition
Speed / Relative Speed	Addition or Maximum
Link Lifetime	Addition or Multiplication or Minimum
Connectivity	Minimum
Map Information	Addition
Energy per Packet / MTPR	Addition
MBCR	Addition

Table A.1: Concatenation operators, Part I

Metric	Proposed Concatenation Operator
MMBCR	Minimum
CMMBCR	Addition and Minimum
MREP	Addition
Battery Prevention	Addition
MTPR with Packet Loss Ratio	Addition
MTPR with Interference	Addition
Security	undefined
Billing	Addition
Application-Specific	Addition
Node Resources	Addition or Minimum
Learning Networks	Addition

Table A.2: Concatenation operators, Part II

A.2 Aggregation Operators

Unlike the concatenation operator, the aggregation operator is explicitly defined only for a small number of metrics. For all other metrics, the aggregation operator has to be chosen according to the requirements of the routing algorithm.

Metric	Proposed Aggregation Operator
Delay, RTT	Weighted Average
Bandwidth	Addition
Packet Loss Ratio	Addition and Multiplication
Energy per Packet	Weighted Average

Table A.3: Aggregation operators

Appendix B

An Alternative Field Function

Each Field-Based Routing algorithm defines a strategy for calculating the value of the field potentials. Baumann's algorithm, for example, rewards high redundancy. In other words, a node with a large number of neighbours will get a high potential (see Section 5.2.2). Baumann's field calculation function is displayed in Algorithm 1. In Figure B.2, we examine this formula at the example depicted in Figure B.1.

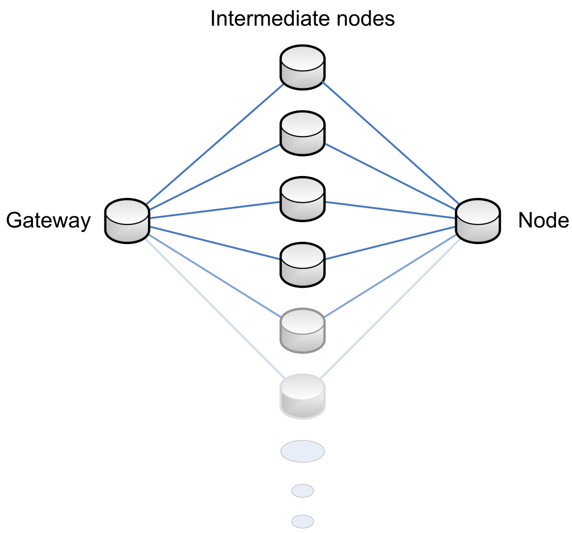


Figure B.1: Example scenario: A node with a high potential – e.g. a gateway – is connected to a node over a varying number of intermediate nodes.

In some situations, however, it can be advantageous not to foster nodes that are strongly connected to other nodes. For example, in certain networks, central nodes can become bottlenecks if all traffic is routed towards them. Therefore, we present a function that privileges nodes with a limited number of links (see Algorithm 5 and Figure B.3).

Algorithm 5 Field calculation function of Baumann's algorithm

Sort x_i in ascending order
$l = 0$
$f(0) = 0$
while $f(l) < x_i$ **do**
    $f(l+1) = f(l) + (x_i - f(l)) \cdot \kappa$
    $l = l + 1$
    Go to next node in sorted list
end while
$f_{final} = (f(l) - \delta) * signum(f(l) - \delta) - \delta$

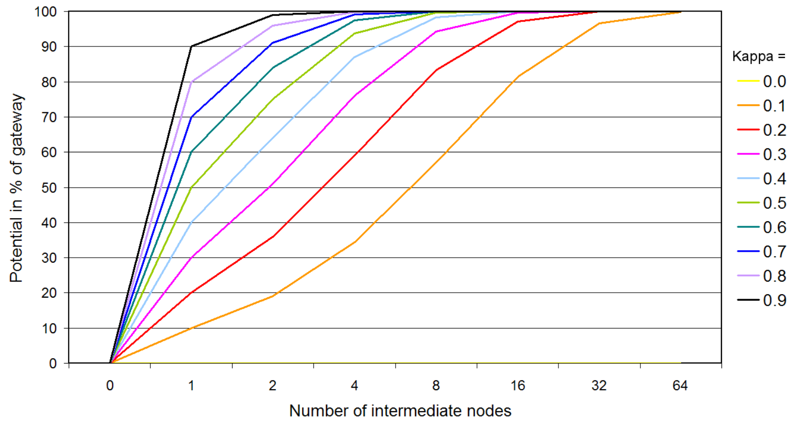


Figure B.2: Baumann's field calculation function at the example in Figure B.1. When the number of intermediate nodes increases, the potential of the examined node converges to the value of the gateway potential. The convergence rate increases with higher κ parameters.

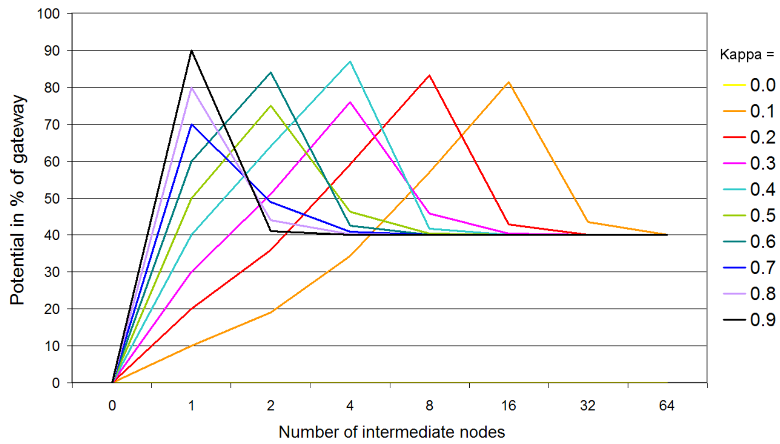


Figure B.3: Alternative field calculation function at the example in Figure B.1. δ is set to 70. When the number of intermediate nodes increases, this is only rewarded up to a certain level (gateway potential$-2 \cdot \delta$). Beyond this level, the potential of the examined node converges to a lower value.

Appendix C

FBR Simulation Tool: User Guide

C.1 Starting the Tool

The FBR Simulation Tool can be started in two ways: One producing a graph with randomly distributed nodes, the other will produce a graph with pseudo-randomly distributed nodes (see Section 6.2.3).

```
java -jar FieldBasedSim2.jar random <NumOfNodes> <NumOfGWs> <R1> <R2>
java -jar FieldBasedSim2.jar pseudo <NumOfRows> <NumOfColumns> <GWRatio> <R1> <R2>
```

The command line parameters are defined as follows:

`NumOfNodes`	Number of ordinary nodes
`NumOfGWs`	Number of gateway nodes
`R1`	Radius1 (see Figure 6.1)
`R2`	Radius2 (see Figure 6.1)
`NumOfRows`	Number of node rows (see Figure 6.2)
`NumOfColumns`	Number of node columns (see Figure 6.2)
`GWRatio`	Likelihood that a node is set as gateway

C.2 Calculation Modes

Currently, six different algorithms are implemented in the tool:

FBR An adopted version of the Lenders FBR algorithm (see Section 5.3.1). Gateways are displayed in blue, ordinary nodes in green. Link weights can have any value greater than zero.

KAPPA The Baumann FBR algorithm (see Section 5.2.2). Gateways are displayed in blue, ordinary nodes in green. The κ can be changed by right-clicking on an empty place in the window and clicking onto `Change Kappa`. The value must lie within the range [0, 1]. Link weights do not have any function.

LIKELIHOOD This algorithm estimates the node failure likelihood based on link failure probabilities. The weight of the links represents the failure probability of each link. The algorithm is restricted to handle only one gateway, which by default has failure ratio 0. The brighter a node is coloured, the farther away from the gateway it is located in terms of hops.

EXPERIMENT1 An adopted version of the Baumann FBR algorithm, see Section 5.3.2. Link weights must lie within the range $[1, \infty]$.

EXPERIMENT2 The Steepest Gradient version of the Baumann FBR algorithm, see Section 5.5.4. Link weights must lie within the range $[1, \infty]$. They are generated randomly every time this mode is chosen. On the links, the metric value of the link is displayed as well as, after the $\|$-sign, the gradient of the link i.e. the potential divided by the metric value of the link.

EXPERIMENT3 An additive version of the Baumann FBR algorithm, see Section 5.5.1. Link weights must lie within the range [1, ∞]. The κ can be changed by right-clicking on an empty place in the window and clicking onto `Change Kappa`. The value must lie within the range [0, 1].

Links always point to the node with higher potential. The outgoing link of each node that is chosen by the route selection mechanism is coloured black, all others grey.

C.3 Changing the Graph

The following functions for manipulation of the graph are available:

Create a node	Right-click on an empty place → `Create Node`
Move a node	Choose `Picking` mode, drag and drop node
Create a gateway	Right-click on an empty place → `Create Gateway`
Change the weight of a gateway	Mark the gateway, Right-click → `Set Gateway Weight`
Create a link	Choose `Editing` mode, drag a line between nodes to be connected.
Change the metric value of a link	Right-click on the link → `Change Link Weight`
Delete a node / gateway	Right-click on the node → `Delete Node`
Delete a link	Right-click on the link → `Delete Link`
Move the entire graph	Choose `Transforming` mode, drag and drop graph
Save a graph	Right-click on an empty place → `Save File`
Load a graph	Right-click on an empty place → `Load File`
Change κ	Right-click on an empty place → `Change Kappa`. This option only has an effect in KAPPA and EXPERIMENT3 mode.

Appendix D

FBR Simulation Tool: Implementation

In the following, a short overview over the source code of the FBR Simulation Tool is given. This appendix is intended to aid users and developers who wish to enhance the tool with more functionality or other algorithms. The source code of the tool is available from the CD handed in together with the thesis or it can be obtained from the author of the thesis under the GNU Public License.

Main.java

Main	The main initialisation constructor. Generates an initial random graph.
calculateAll	Calculates the potentials of all nodes. (Calls calculatePotential or calculateLikelihoodPotential.)
calculatePotential	Calculates the potential of the nodes that are linked with one given node. (Calls calculateLikelihoodPotential, or calculate-FBRPotential, or calculateKappaPotential.)
calculateLikelihoodPotential	Calculates the failure probability of each vertex. Arbitrarily chooses one single node as gateway.
calculateFBRPotential	Recursively calculates the potential with the adopted Lenders' formula (see Section 5.3.1).
calculateKappaPotential	Recursively calculates the potential with Baumann's formula (see Section 5.2.2).
calculateExperimentPotential	Recursively calculates the potential in the EXPERIMENT1, EXPERIMENT2 and EXPERIMENT3 modes.
main	The driver method
loadFile	Load a graph from file. Uses class PajekNetReader (see below).
saveFile	Save the currently displayed graph to file.
graph2file	Turns the graph into a string in NET format
calculateDirections	Turn each link such that it points in the direction of the vertex with higher potential.
markBestEdge	The path select function. For each vertex, chooses the link with the potential difference. This link will be coloured black, other links will be grey. EXPERIMENT2 and EXPERIMENT3 modes are treated specially.

Table D.1: Methods in Main.java

Popup	Defines the pop-up that is shown on right-click and the actions that are taken when a pop-up entry is chosen.
EdgeColor	Determines the colour given to edges.
VertexColor	Determines the colour given to vertices.
VertexColorLikelihood	Determines the colour given to edges if the LIKELIHOOD mode is active.
VertexLabel	Determines the text that is displayed next to a vertex (the potential).
EdgeLabel	Determines the text that is displayed next to an edge (value of the metric / none in KAPPA mode).

Table D.2: Classes in Main.java

RandomGraphGen.java

generateRadomGraph	Returns a graph with arbitrarily distributed nodes (true random). Vertices within distance radius1 are always linked, vertices within a distance between radius1 and radius2 are linked with a probability depending on the distance
generatePseudoRandomGraph	Returns a graph with one arbitrarily placed node for each square (uniform random). Vertices within a distance radius1 are always linked, vertices within a distance between radius1 and radius2 are linked with a probability depending on the distance.

Table D.3: Methods in RandomGraphGen.java

PajekNetReader.java

PajekNetReader	Read a graph from a file in NET format.

Table D.4: Classes in PajekNetReader.java

EditingModalGraphMouse.java

EditingGraphMousePlugin	Determines what happens when a new link is added using the mouse: Add a link with metric = 1.
EditingModalGraphMouse	Layout class for `Editing` mode. Uses EditingGraphMousePlugin.

Table D.5: Classes in EditingModalGraphMouse.java

Appendix E

CD Contents

The included CD-ROM contains the following directories:

- `papers`
  Most of the literature used for this thesis
- `presentation`
  Presentation slides
- `report`
  This report
- `report/src`
  The source files of this report, including the images
- `report/various`
  Different documents that were utilised in the report
- `tool/bin`
  Class files of the FBR Simulation Tool
- `tool/code`
  Source code of the FBR Simulation Tool
- `tool/doc`
  Small documentation of the code

Appendix F

Thesis Timetable

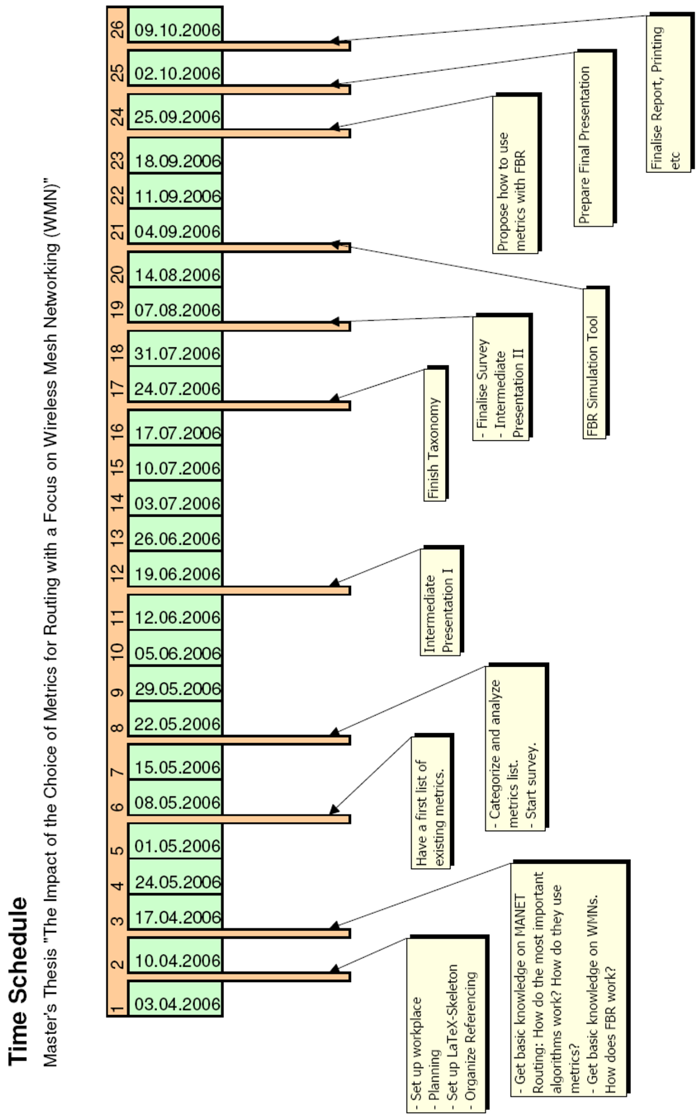


Figure F.1: Thesis timetable

References

[1] E. Moglen, "Freeing the mind: Free software and the death of proprietary culture," in *University of Maine Law School's Fourth Annual Technology and Law Conference*, (Porland, Maine), June 29 2003.

[2] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[3] D. S. J. D. Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: Shortest path is not enough," in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, (Princeton, New Jersey), ACM SIGCOMM, October 2002.

[4] R. Baumann, M. May, and B. Plattner, "Field based interconnection of hybrid wireless mesh networks," TIK Report 249, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, July 2006.

[5] R. Baumann, S. Heimlicher, M. May, V. Lenders, K. Farkas, and B. Plattner, "Field-based interconnection of hybrid wireless mesh networks," in *INFOCOM (submitted)*, 2007.

[6] V. Lenders, M. May, and B. Plattner, "Service discovery in mobile ad hoc networks: A field theoretic approach," *Journal on Pervasive and Mobile Computing (PMC)*, vol. 1, pp. 343–370, September 2005.

[7] V. Lenders and B. Plattner, "Magnetic routing: A content-based routing approach for ad hoc networks," TIK Report 184, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, December 2003.

[8] A. Khanna and J. Zinky, "The revised arpanet routing metric," *SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 4, pp. 45–56, 1989.

[9] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[10] J. Sobrinho, "Algebra and algorithms for qos path computation and hop-by-hop routing in the inernet," in *IEEE INFOCOM 2001*, 2001.

[11] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *SIGCOMM '03: Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 49–60, ACM Press, 2003.

[12] L. Iannone, R. Khalili, K. Salamatian, and S. Fdida, "Cross-layer routing in wireless mesh networks," in *1st International Symposium on Wireless Communication Systems*, pp. 319–323, 2004.

[13] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross layer design," *IEEE Wireless Communications*, pp. 3–11, February 2005.

[14] J. J.-N. Liu and I. Chlamtac, *Mobile Ad Hoc Networking*, ch. Mobile Ad Hoc Networking with a View of 4G Wireless: Imperatives and Challenges, pp. 1–45. IEEE Press, 2004.

[15] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. 23–30, 2005.

[16] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks." IEEE Communications Magazine, March 2005.

[17] P. Ferguson and G. Huston, *Quality of Service - Delivering QoS on the Internet in Corporate Networks.* Robert Ipsen / John Wiley & Sons, Inc., 1998.

[18] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the ACM/IEEE MobiCom*, October 1998.

[19] E. M. Belding-Royer, *Mobile Ad Hoc Networking*, ch. Routing Approaches in Mobile Ad Hoc Networks, pp. 275–300. IEEE Press, 2004.

[20] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, 2004.

[21] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks*, ch. 7, pp. 304–307. Bernard M. Goodwin, 2004.

[22] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *Proceedings of SIGCOMM*, October 1994.

[23] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, (New Orleans, LA), pp. 90–100, February 1999.

[24] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (T. Imielinski and H. Korth, eds.), (Dordrecht, The Netherlands), Kluwer Academic Publishers, 1996.

[25] P. Basu, W. Ke, and T. D. Little, "Metrics for performance evaluation of distributed application execution in ubiquitous computing environments," in *Proceedings of the ACM Workshop on Evaluation Methodologies for Ubiquitous Computing*, (Atlanta, GA), September 2001.

[26] V. Paxson, "Towards a framework for defining internet performance metrics," in *Proceedings of INET '96*, (Montreal), June 1996.

[27] H. Uijterwaal, "Overview of implementation reports relating to ietf work on ip performance measurements," IETF Internet Draft, draft-ietf-ippm-implement-02.txt, January 2006. (Work in Progress).

[28] S. Keshav, *An Engineering Approach to Computer Networking.* Addison-Wesley / AT&T, 1997.

[29] L. L. Peterson and B. S. Davie, *Computer Networks.* Morgan Kaufmann, third ed., 2003.

[30] C. Hendrick, "Routing information protocol," IETF RFC 1058, June 1988 http://www.ietf.org/rfc/rfc1058.txt.

[31] J. Moy, "Ospf version 2," IETF RFC 1247, July 1991 http://www.ietf.org/rfc/rfc1247.txt.

[32] Y. Rekhter and T. Li, "A border gateway protocol 4 (bgp-4)," IETF RFC 1771, March 1995 http://www.ietf.org/rfc/rfc1771.txt.

[33] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, (Banff, Canada), October 2001.

[34] A. Faragó, *Mobile Ad Hoc Networking*, ch. Algorithmic Callenges in Ad Hoc Networks, pp. 427–445. IEEE Press, 2004.

[35] J. Matousek and J. Nesetril, *Invitation to Discrete Mathematics.* Oxford University Press, 1998.

[36] J. Jost, *Riemannian Geometry and Geometric Analysis.* Berlin: Springer-Verlag, 2002.

[37] J. D. Lipson, *Elements of Algebra and Algebraic Computing.* Addison-Wesley Publishing, 1981.

[38] Y. Yang, J. Wang, and R. Kravets, “Designing routing metrics for mesh networks.” IEEE Workshop on Wireless Mesh Networks, WiMesh, 2005.

[39] E. Stephan, L. Liang, and A. Morton, “Ip performance metrics (ippm) for spatial and multicast,” IETF Internet Draft, draft-ietf-ippm-multimetrics-00, January 2006. (Work in Progress).

[40] J. Boleng, T. Camp, and W. Navidi, “Metrics to enable adaptive protocols for mobile ad hoc networks,” in *Proceedings of the International Conference on Wireless Networking*, 2002.

[41] Y. Yang, J. Wang, and R. Kravets, “Interference-aware load balancing for multihop wireless networks,” Tech. Rep. UIUCDCS-R-2005-2526, University of Illinois at Urbana-Champaign, March 2005.

[42] H. Lundgren, E. Nordström, and C. Tschudin, “Coping with communication gray zones in ieee 802.11b based ad hoc networks,” in *WOWMOM '02: Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, (New York, NY, USA), pp. 49–55, ACM Press, 2002.

[43] H. Zhang, A. Arora, and P. Sinha, “Learn on the fly: Data-driven link estimation and routing in sensor network backbones,” in *25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[44] V. Raisanen, G. Grotefeld, and A. Morton, “Network performance measurement with periodic streams,” IETF RFC 3432, November 2002
http://www.ietf.org/rfc/rfc3432.txt.

[45] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” in *SIGCOMM 2000*, (Stockholm, Sweden), pp. 43–56, August 2000.

[46] V. Jacobson, “Congestion avoidance and control,” in *ACM SIGCOMM '88*, (Stanford, CA), pp. 314–329, August 1988.

[47] R. Draves, J. Padhye, and B. Zill, “Comparison of routing metrics for static multi-hop wireless networks,” in *SIGCOMM '04: Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 133–144, ACM Press, 2004.

[48] P. Gupta and P. Kumar, “A system and traffic dependent adaptive routing algorithm for ad hoc networks,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, 1997.

[49] G. Almes, S. Kalidindi, and M. Zekauskas, “A one-way delay metric for ippm,” IETF RFC 2679, September 1999
http://www.ietf.org/rfc/rfc2679.txt.

[50] S. Shalunov, B. Teitelbaum, A. Karp, J. W. Boote, and M. J. Zekauskas, “A one-way active measurement protocol (owamp),” IETF Internet Draft, draft-ietf-ippm-owdp-16.txt, February 2006. (Work in Progress).

[51] V. Jacobson, R. Braden, and D. Borman, “Tcp extensions for high performance,” IETF RFC 1323, May 1992
http://www.ietf.org/rfc/rfc1323.txt.

[52] G. Almes, S. Kalidindi, and M. Zekauskas, “A round-trip delay metric for ippm,” IETF RFC 2681, September 1999
http://www.ietf.org/rfc/rfc2681.txt.

[53] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A multi-radio unification protocol for ieee 802.11 wireless networks,” in *Proceedings of the First International Conference on Broadband Networks, BroadNets*, pp. 344–354, 2004.

[54] C. Demichelis and P. Chimento, “Ip packet delay variation metric for ip performance metrics (ippm),” IETF RFC 3393, November 2002
http://www.ietf.org/rfc/rfc3393.txt.

[55] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[56] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.

[57] K. Lai and M. Baker, "Measuring bandwidth," in *Proceedings of IEEE INFOCOM '99*, (New York, NY), March 1999.

[58] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance Evaluation*, vol. 27-28, pp. 297–318, October 1996.

[59] P. Chimento and J. Ishac, "Defining network capacity," IETF Internet Draft, draft-ietf-ippm-bw-capacity-01, November 2005. (Work in Progress).

[60] V. Paxson, "End-to-end internet packet dynamics," in *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, vol. 27,4 of *Computer Communication Review*, (Cannes, France), pp. 139–154, ACM Press, September 1997.

[61] C. R. Lin and J.-S. Liu, "Qos routing in ad hoc wireless networks," *IEEE Journal on Selected Topics in Communication*, vol. 17, pp. 1426–1438, August 1999.

[62] M. Mathis and M. Allman, "A framework for defining empirical bulk transfer capacity metrics," IETF RFC 3148, July 2001
http://www.ietf.org/rfc/rfc3148.txt.

[63] T.-W. Chen, J. Tsai, and M. Gerla, "Qos routing performance in multihop, multimedia, wireless networks," in *6th IEEE International Conference on Universal Personal Communications Record, Conference Record.,*, vol. 2, pp. 557–561 vol.2, 1997.

[64] H. Chang, V. Misra, and D. Rubenstein, "A new routing metric for high throughput in dense ad hoc networks," Tech. Rep. CUCS-047-05, Columbia University, December 2005.

[65] V. Jacobson, "Pathchar - a tool to infer characteristics of internet paths." Slides:
ftp://ftp.ee.lbl.gov/pathchar/, April 1997.

[66] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *Proceedings of ACM SIGCOMM 1999*, (Cambridge, MA), pp. 241–250, September 1999.

[67] S. Keshav, "A control-theoretic approach to flow control," *SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 1, pp. 188–201, 1995.

[68] G. Almes, S. Kalidindi, and M. Zekauskas, "A one-way packet loss metric for ippm," IETF RFC 2680, September 1999
http://www.ietf.org/rfc/rfc2680.txt.

[69] R. Koodli and R. Ravikanth, "One-way loss pattern sample metrics," IETF RFC 3357, August 2002
http://www.ietf.org/rfc/rfc3357.txt.

[70] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communicatioins*, vol. 21, pp. 1615–1626, Decemeber 2004.

[71] M. Borella, D. Swider, S. Uludag, and G. Brewster, "Internet packet loss: Measurement and implications for end-to-end qos," in *Proceedings of the International Conference on Parallel Processing*, August 1998.

[72] S. Biaz and N. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *IEEE Symposium on Application - Specific Systems and Software Engineering and Technology*, 1999.

[73] M. Yarvis, W. Conner, L. Krishnamurthy, J. Chhabra, B. Elliott, and A. Mainwaring, "Real-world experiences with an interactive ad hoc sensor network," in *Proceedings of the International Workshop on Ad Hoc Networking*, August 2002.

[74] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser, "Packet reordering metric for ippm," IETF Internet Draft, draft-ietf-ippm-reordering-13.txt, May 2006. (Work in Progress).

[75] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of MOBICOM 2003*, (San Diego), 2003.

[76] C. E. Koksal and H. Balakrishnan, "Quality aware routing in time-varying wireless networks," *To appear in IEEE Journal on Selected Areas of Communication Special Issue on Multi-Hop Wireless Mesh Networks*, 2006.

[77] T. Lopatic, "olsrd link quality extensions." Internet page: http://www.olsr.org/docs/README-Link-Quality.html, December 2004.

[78] J. C. Park and S. K. Kasera, "Expected data rate: An accurate high-throughput path metric for multi-hop wireless routing," in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005.

[79] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 114–128, ACM Press, 2004.

[80] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, pp. 31–43, January 2006.

[81] S. Zhao, Z. Wu, A. Acharya, and D. Raychaudhuri, "Parma: A phy/mac aware routing metric for ad-hoc wireless networks with multi-rate radios," Tech. Rep. RC23469 (W0412-042), IBM Research Division, Yorktown Heights, NY, December 2004.

[82] R. Dube, C. Rais, K. Wang, and S. Tripathi, "Signal stability based adaptive routing (ssa) for ad hoc mobile networks." in *IEEE Personal Communication*, February 1997.

[83] B. Awerbuch, D. Holmer, and H. Rubens, "The medium time metric: High throughput route selection in multirate ad hoc wireless networks," tech. rep., Johns Hopkins University, Department of Computer Science, October 2004. Baltimore, Maryland.

[84] M. Lampe, H. Rohling, and W. Zirwas, "Misunderstandings about link adaptation for frequency selective fading channels." In *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 710-714, 2002.

[85] L. Iannone and S. Fdida, "Mrs: A simple cross-layer heuristic to improve throughput capacity in wireless mesh networks," in *Proceedings of Conext 2005 Conference*, (Toulouse, France), October 2005.

[86] B. Awerbuch, D. Holmer, and H. Rubens, "High throughput route selection in multi-rate ad hoc wireless networks." In *Technical report, Johns Hopkins University, Computer Science Department. Version 2.*, March 2003.

[87] D. Aguayo, J. Bicket, and R. Morris, "Srcrr: A high throughput routing protocol for 802.11 mesh networks (draft)." Internet Article: http://pdos.csail.mit.edu/ rtm/srcrr-draft.pdf, 2006.

[88] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, (New York, NY, USA), pp. 31–42, ACM Press, 2005.

[89] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, no. 61, 1967.

[90] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[91] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," IETF RFC 3626, October 2003; http://www.ietf.org/rfc/rfc3626.txt.

[92] V. R. Syrotiuk and A. Bikki, *Mobile Ad Hoc Networking*, ch. Modeling Cross-Layer Interaction Using Inverse Optimization, pp. 411–425. IEEE Press, 2004.

[93] R. Guerin, A. Orda, and D. Williams, "Qos routing mechanisms and ospf extensions.." Internet Draft: draft-guerin-qos-routing-ospf-00.txt, November 1996.

[94] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks in mobile ad hoc networks," vol. 15, pp. 30–39, November 2001.

[95] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on embedded networked sensor systems*, (New York, NY, USA), pp. 108–121, ACM Press, 2004.

[96] P. Basu, W. Ke, and T. D. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of the IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing*, (Phoenix, AZ), April 2001.

[97] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 195–206, ACM Press, 1999.

[98] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications*, vol. 4, pp. 103–139, Mar. 1997.

[99] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," in *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pp. 480–486, 1996.

[100] A. Mcdonald and T. Znati, "A path availability model for wireless ad-hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1998.

[101] M. Gerharz, C. de 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 (LCN'02)*, (Tampa, FL), pp. 30–39, November 2002.

[102] S. Jiang, D. He, and J. Rao, "A prediction-based link availability estimation for mobile ad hoc networks," *INFOCOM*, pp. 1745–52, 2001.

[103] J. Mahdavi and V. Paxson, "Ippm metrics for measuring connectivity," IETF RFC 2678, September 1999; http://www.ietf.org/rfc/rfc2678.txt.

[104] R. Punnoose, P. Nikitin, J. Broch, and D. Stancil, "Optimizing wireless network protocols using realtime predictive propagation modeling," in *Proceedings of the IEEE Radio and Wireless Conference 1999 (RAWCON'99)*, (Denver, CO), August 1999.

[105] H. Messer, A. Zinevich, and P. Alpert, "Environmental monitoring by wireless communication networks," *Science*, vol. 312, no. 5774, p. 713, 2006.

[106] L. M. Feeney, *Mobile Ad Hoc Networking*, ch. Energy-Efficient Communication in Ad Hoc Wireless Networks, pp. 301–327. IEEE Press, 2004.

[107] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks." In *SITE, University of Ottawa*, 2000.

[108] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2, 2000.

[109] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, (Tel Aviv, Israel), pp. 404–413, March 2000.

[110] T. Meng and V. Rodoplu, "Distributed network protocols for wireless communication," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems ISCAS*, vol. 4, pp. 600–603 vol.4, 1998.

[111] D. Kim, J. Garcia-Luna-Aceves, K. Obraczka, J. Cano, and P. Manzoni, "Performance analysis of power-aware route selection protocols in mobile ad hoc networks," *IEEE NETWORKS (submitted)*, 2002.

[112] Q. Li, J. A. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Mobile Computing and Networking*, pp. 97–107, 2001.

[113] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 181–190, 1998.

[114] K. Scott and N. Bambos, "Routing and channel assignment for low power transmission in pcs," in *ICUPC '96*, vol. vol. 2, pp. pp. 498–502, 1996.

[115] A. Michail and A. Ephremides, "Energy-efficient routing for connection-oriented traffic in wireless ad-hoc networks," *Mobile Networks and Applications*, vol. 8, pp. 517–533, October 2003.

[116] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks." *INFOCOM '2000*, March 2000.

[117] J.-P. Sheu, C.-T. Hu, and C.-M. Chao, *The Handbook of Ad Hoc Wireless Networks*, ch. Energy-Conserving Grid Routing Protocol in Mobile Ad Hoc Networks, pp. 24–1 – 24–15. RCR Press LLC, 2003.

[118] N. Gupta and S. R. Das, "Energy-aware on-demand routing for mobile ad hoc networks," in *IWDC*, vol. 2571, pp. 164–173, January 2002.

[119] C.-K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," *IEEE Communications Magazine*, pp. 2–11, June 2001.

[120] J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," in *Proceedings of Advanced Telecommunications and Information Distribution Research Program*, (College Park, MD), 2000.

[121] C. F. Chiasserini and R. R. Rao, "Routing protocols to maximize battery efficiency," in *Proceedings of the IEEE MILCOM*, (Los Angeles, CA), pp. 496–500, October 2000.

[122] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," Technical Report CS-TR 4315, Department of Computer Science, University of Maryland, College Park, MD, December 2001.

[123] A. Misra and S. Banerjee, "Maximizing network lifetime for reliable routing in wireless environments," in *Proceedings of IEEE Wireless Communications and Networking Conference*, March 2002.

[124] P. Michiardi and R. Molva, "Game theoretic analysis of security in mobile ad hoc networks," Research Report RR-02-070, Institut Eurecom, Sophia-Antipolis, April 2002.

[125] S. Giordano and A. Urpi, *Mobile Ad Hoc Networking*, ch. Self-Organized and Cooperative Ad Hoc Networking, pp. 355–371. IEEE Press, 2004.

[126] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks*, ch. 13, pp. 736–750. Bernard M. Goodwin, 2004.

[127] L. Barolli, A. Koyama, and N. Shiratori, "A qos routing method for ad-hoc networks based on genetic algorithm," *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, p. 175, 2003.

[128] M. Gunes, *Routing und Adressierung in mobilen multi-hop Ad-hoc-Netzen*. Dissertation, RWTH Aachen, 2004.

[129] A. Basu, A. Lin, and S. Ramanathan, "Routing using potentials: a dynamic traffic-aware routing algorithm," *SIGCOMM'03*, pp. 37–48, August 2003.

[130] R. Baumann, M. May, and B. Plattner, "A study on field based routing and extensions," TIK Report 261, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, October 2006.

[131] Java Universal Network/Graph Framework, http://jung.sourceforge.net, Internet Page, 13 September 2006.