TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

DIPARTIMENTO DI INGEGNERIA BIOMEDICA,
ELETTRONICA E DELLE TELECOMUNICAZIONI

DOTTORATO DI RICERCA IN
INGEGNERIA ELETTRONICA E DELLE TELECOMUNICAZIONI

MOBILE AD HOC NETWORKS: THE DHT PARADIGM

MARCELLO CALEFFI

Il Coordinatore del Corso di Dottorato	Il Tutore
Ch.mo Prof. Giovanni POGGI	Ch.mo Prof. Luigi PAURA

XXI ciclo

Computer games
don't affect kids,
I mean if Pac Man
affected us as kids,
we'd all be running
around in darkened
rooms, munching pills
and listening to
repetitive music.

Kristian Wilson, whoever he was!

Acknowledgments

I would like to thank my advisor, Prof. Luigi Paura, for his trust and unwavering support.

A special thanks to my parents, by the way, for getting me my first computer.

Finally, thanks to my wife Sara for her love and understanding.

Contents

List of Figures

List of Acronyms

AODV Ad Hoc On-Demand Distance Vector

AODV-BR Ad hoc On-Demand Distance Vector Routing with Backup Routes

AOMDV Ad hoc On-demand Multipath Distance Vector

ARPANET Advanced Research Projects Agency Network

ARQ automatic repeat request

ATR Augmented Tree-based Routing

BAN body area network

BSC base station controller

CBR constant bit rate

CCK complementary code keying

CSMA carrier sense multiple access

DARPA Defense Advanced Research Projects Agency

DTN delay tolerant network

DHT distribute hash table

DSDV Destination-Sequenced Distance Vector

DSR Dynamic Source Routing

DSSS direct sequence spread spectrum

DTN disruption tolerant network

DART Dynamic Address RouTing

DPSR Dynamic P2P Source Routing

ETX expected transmission count

FEC forward error control

GPS Global Positioning System

GSM Global System for Mobile communications

HiperLAN HIgh Performance Radio Local Area Network

ETSI European Telecommunication Standard Institute

IETF Internet Engineering Task Force

IEEE Institute of Electrical and Electronics Engineers

IRTF Internet Research Task Force

IP internet protocol

ITR Indirect Tree-based Routing

LAN local area network

LAR Location-Aided Routing

LOS line of sight

LPR Low-cost Packet Radio

MA moving average

MAC Media Access Control

MAN metropolitan area network

MANET mobile ad hoc network

MPR multipoint relay

MSC mobile switching center

ODR Opportunistic DHT-based Routing

OLSR Optimized Link State Routing

P2P peer-to-peer

PAN personal area network

PRNet Packet Radio Network

PDR packet delivery ratio

PDU protocol data unit

RDP route discovery process

SINR signal to interference plus noise ratio

SURAN Survivable Radio Networks

TCP Transmission Control Protocol

TDMA time division multiple access

TTL time to live

TPRR terminal-pair routing reliability

UDP User Datagram Protocol

VRR Virtual Ring Routing

WAN wide area network

WLAN wireless LAN

WiMAX Worldwide Interoperability for Microwave Access

ZRP Zone Routing Protocol

Introduction

The proliferation of mobile communication and computing devices makes wireless technology the only solution to provide end-to-end connectivity.

Although in the last two decades great attention has been devoted to the ad hoc networking paradigm, which tries to enhance the feasibility of wireless networking by allowing multi-hop communications in absence of pre-existent infrastructure or central administration, currently wireless networking is still limited to single-hop communications achieved via infrastructure-based topologies.

Despite its attractive features, the transition from "traditional" networking to ad hoc networking gives rise to several challenging problems. Ad hoc networks inherit all the traditional problems of wireless and mobile communications, such as bandwidth optimization, power control and transmission quality enhancement. In addition, the multihop nature and the lack of fixed infrastructure bring new issues such as network configuration, device discovery, topology maintenance and ad hoc addressing.

In the last years, many different approaches and protocols have been proposed to overcome these drawbacks, and there are multiple standardization efforts within the Internet Engineering Task Force and the Internet Research Task Force, as well as academic and industrial projects. These efforts have produced several routing protocols able to perform very well in small networks. However, it has been proved that the overhead needed by all of them to provide network connectivity increases so fast with the number of nodes that it eventually consumes all of the available bandwidth also in networks of moderate size.

A trivial solution to this problem is to arbitrarily consider only small networks. Since present wireless networks are composed by few dozens of nodes and large ones are merely a prof of concept, why one should worry about scalability issues?

In 1977, Leonard Kleinrock and Farouk Kamoun answered to the same question in the paper *"Hierarchical routing for large networks"* [1]:

«Present computer networks may be characterized as small to moderate in size (77 nodes for the ARPANET as of December 1975). Predictions indicate that, in fact, large networks of the order of hundreds (or even possibly thousands) of nodes are soon to come. In the course of developing the ARPANET, a design methodology has evolved which is quite suitable for the efficient design of small and moderate networks. Unfortunately the cost of conducting the design is prohibitive if the same techniques are extrapolated to the case of large networks. Indeed, not only the cost of design grow exponentially with the network size, but also the cost of a straightforward adaptive routing procedure becomes prohibitive. Other design and operational procedures (routing techniques) must be found which handle the large networks case. »

At that time, Internet was in its early stages and very few could predict its explosive growth. However, most researchers were involved in developing networking protocol for large store-and-forward packet-switched wired networks. Differently, currently ad hoc networks research seems to have downplayed the importance of scalability and, moreover, it seems to restrict itself to adapting wired networking techniques to ad hoc networks.

In this thesis, we focus on providing a scalable network layer for ad hoc networks by resorting to both traditional and innovative networking paradigms. The first traditional paradigm, referred to as *hierarchical routing*, has been proposed by Kleinrock and Kamoun in the above mentioned paper:

«The main idea ... is to keep, at any node, complete routing information about nodes which are close to it (in terms of a hop distance or some other nearness measure), and lesser information about nodes located further away from it. This can be realized by providing one entry per destination for the closer nodes, and one entry per set of destinations for the remote nodes. The size of this set may increase with the distance. »

Hierarchical routing is still the basic idea of current wired networking and its scalability has been substantiate throughout the ages, therefore in this thesis we propose a network layer based on such a paradigm. However, to face with the flat, transient nature of ad hoc topologies, new distributed procedures for node clustering have to be found. We solve this issue by resorting to location-aware network addresses, that is by defining a logical hierarchy over the network topology in the address space.

To provide a scalable mapping between transient addresses and node iden-

tifiers, we adopt a novel routing paradigm, the *indirect key-based* one, based on distribute hash table (DHT) systems. Such a routing, initially proposed for application layer peer-to-peer systems, requires an amount of sophistication to couple with ad hoc network features. In fact, it has been proved that simply deploying indirect key-based routing over ad hoc networks may cause poor performances due to the lack of cooperation and communication between the two layers, causing so significant message overhead and redundancy. In this thesis, we address to these issues by integrating both traditional direct routing and indirect key-based routing at the network layer.

Unlike traditional routing procedures that, at the best, single out a unique route, we explore also the feasibility of multi-path routing, which consists of proactively discovering several alternative routes towards the same destination. Moreover, since most studies in the area of multi-path routing focus on heuristic methods, and the performances of these strategies are commonly evaluated by numerical simulations, in this thesis we define an analytical framework to evaluate the performance gain achieved by multi-path routing resorting to the graph theory.

Finally, we deal with delay tolerant networks (DTNs), that is with ad hoc networks characterized by sparse topologies as well as hostile propagation conditions. Traditionally, ad hoc networking tries to fortify the environment so that it behave like a wired network. More in detail, the wireless channel is reinforced by means of automatic repeat request (ARQ) or forward error control (FEC) data-link techniques to counteract the time-variant impairment of the wireless propagation, while the transient network topology is fortified resorting to multi-path and/or flooding routing techniques. These approaches are based on two hypotheses. The former is that the network topology is quite dense to assure the presence of a persistent path between each pair of nodes and the latter assures that the wireless propagation conditions are enough stationary to allow a persistent communication among neighbor nodes. In this thesis, rather than counteracts, we try to take advantages by the time-variant nature of the environment to provide end-to-end connectivity in scenarios where traditional networking fails resorting to the *opportunistic networking* paradigm.

The outline of the thesis is the following:

Chapter 1 presents the general framework. The ad hoc paradigm is introduced along with the main design constraints of ad hoc networking. The latest research activities related with routing in mobile ad hoc networks (MANETs) are presented and discussed as well.

Chapter 2 addresses the design of a scalable network layer for mobile ad hoc networks, referred to as Augmented Tree-based Routing (ATR) protocol. In such a design, we resort to both hierarchical and indirect key-based routing to achieve a scalable network layer. A number of optimization has been used to face with fading channels and, according to our knowledge, ATR is the first protocol for ad hoc networking able to exploit a multi-path approach without introducing any communication overhead. To effectively assess the scalability property of the proposed protocol, we accomplish a performance comparison among some widely adopted routing protocols with a data load which accounts for the theoretical capacity scaling bounds of wireless multi-hop networks.

In Chapter 3 we describe the terminal-pair routing reliability (TPRR), a metric for evaluating the tolerance of multi-path routing against route failures. Such a metric allows one to evaluate the robustness against the link failures, as a function of the number of the discovered routes as well as their reliability. Moreover, an upper bound on the terminal-pair routing reliability of any shortest-path routing protocol it is introduced, allowing so an easily comparison among multi-path and shortest-path routing protocols. The effectiveness of the proposed framework has been evaluated by means of a widely used routing performance metric, the packet delivery ratio (PDR).

In Chapter 4 we resort to some features of Augmented Tree-based Routing for providing a peer-to-peer (P2P) system over a mobile ad hoc network. It has been proved that simply deploying P2P systems over MANETs may cause poor performances. By coupling both the direct and the indirect key-based routing at the network layer and by resorting to the same hierarchical address space structure of ATR, we are able to build a P2P overlay network in which the logical proximity agrees with the physical one, limiting so the message overhead and avoiding the redundancy.

Finally, in Chapter 5 we propose a routing protocol based on opportunistic networking, namely the Opportunistic DHT-based Routing (ODR) protocol. By exploiting both the temporal diversity and the broadcast nature of the wireless propagation, opportunistic networking can enable connectivity in ad hoc environments characterized by non stationary wireless propagation as well as sparse topologies. In this chapter, we extend the location-aware addressing schema proposed for ATR to match with opportunistic forwarding, building so an distribute procedure for candidate selection able to exploit all the opportunities offered by the wireless propagation.

Chapter 1

Ad Hoc Networks

Ad hoc networks represent complex distributed systems comprised by wireless nodes that can freely and dynamically self-organize into arbitrary and temporary (ad hoc) network topologies, allowing so communications in areas with no pre-existing infrastructure. The ad hoc network paradigm is not a new concept, since it has been proposed 20 years ago mainly for tactical networks. Recently, the introduction of enabling technologies, such as Bluetooth and Wi-Fi, has allowed the deployment of commercial ad hoc networks outside the military domain, generating so a renewed and growing interest in the research and development of such networks. This chapter provides an overview of the ad hoc paradigm by presenting its main characteristics and design constraints. The latest research activities related with networking for ad hoc networks will be presented and discussed as well.

1.1 Introduction

The proliferation of mobile communication and computing devices is driving a revolutionary change in the Information and Communication Technology (ICT) domain. We are moving from the Personal Computer age, i.e. one computing device per user, to the Ubiquitous Computing age, in which several devices are utilized by the same user to access the required information whatever and wherever needed [2]. Since wireless communications are the easiest solution to interconnect devices in the ubiquitous computing, they have been experiencing an exponential growth in the past decade [3].
Nowadays, the most of the connections among wireless devices are still achieved via infrastructure-based networks: the connections between cell

phones are setup by BSC and MSC in cellular networks, while laptops are connected to Internet via wireless access points. Although infrastructure-based networks provide a great way for mobile devices to get network services, their deployment requires time and resources. Furthermore, there are scenarios where communication infrastructures are not available in a given geographic area, as it happens in tactical or emergence networks. In such cases, providing the needed connectivity becomes a real challenge.
The ad hoc paradigm has been proposed to overcome such issues by providing a self-organizing network infrastructure, that is by allowing a device to connect with others through automatic configuration as soon as they come in its transimission range. The ad hoc networks not only provide *spontaneous connectivity* [4] among devices in absence of communication infrastructure, but also extend the Internet services beyond the infrastructured areas.

1.2 Ad hoc paradigm

An ad hoc network is a transient network able to provide connectivity among a collection of arbitrarily located wireless devices, namely *nodes*, without relying on pre-existent network infrastructures (i.e. routers, switches, servers, ecc.) or centralized administration. In such a network, the nodes represent the infrastructure, since they cooperate to provide the connectivity functionalities resorting to the multi-hop paradigm, i.e. by acting as relays for neighbors' communications.
In the following, we will provide first an historical overview of the ad hoc paradigm and then we will focus on the ad hoc issues from a network-layer perspective.

1.2.1 Ad hoc networks evolution

Historically, the ad hoc paradigm has been primarily proposed to improve battlefield communications in tactical networks. In fact, the dynamic nature of military operations forbids to rely on fixed infrastructures. Moreover, high data-rate wireless communications rarely propagate beyond line of sight (LOS) [5]. In such a scenario, mobile ad hoc networks (MANETs) represent a suitable framework to address these issues by providing connectivity beyond LOS without a pre-placed infrastructure.
In 1973, the Defense Advanced Research Projects Agency (DARPA) initiated a research program, namely the Packet Radio Network (PRNet) project, on

the feasibility of using packet-switched wireless communications to provide reliable tactical networks. PRNet featured a distributed architecture consisting of a network of wireless devices with minimal central control. A combination of Aloha and carrier sense multiple access (CSMA) protocols were used to support the dynamic sharing of the wireless channel, along with multi-hop store-and-forward routing techniques to enable communications beyond the device coverage.

A following program, namely the Survivable Radio Networks (SURAN), was developed by DARPA in 1983 to address some issues of PRNet and to develop more sophisticated algorithms for large networks (thousands of nodes) composed by small, low-cost, low-power devices. This effort resulted in the design of the Low-cost Packet Radio (LPR) technology [6], which featured a digitally controlled direct sequence spread spectrum (DSSS) radio with an integrated Intel 8086 microprocessor-based packet switch.

Towards late 1980s and early 1990s, the growth of the Internet infrastructure and the micro-processor revolution made the concept of packet wireless network more applicable and feasible.

In 1994, DARPA initiated the Global Mobile (GloMo) Information Systems program [7], which aimed to support Ethernet-type multimedia connectivity any time, anywhere among wireless devices by exploiting both flat and hierarchical network architectures.

In 1997, the US Army implemented the Tactical Internet (TI) network, which has been one of the largest-scale implementation of mobile wireless network. A DSSS-based modulation along with modified commercial Internet protocols for networking enabled data rates in the order of tens kilobits per second. However, the experimental results reinforced the perception that commercial wireline protocols were not suitable to cope with topology changes, low data rate and high bit error rate, commonly in wireless communications.

In 1999, the Extending the Littoral Battle-space Advanced Concept Technology Demonstration (ELB ACTD) was another test bed to demonstrate the feasibility of over-the-horizon communications with aerial relay.

However, in the middle of 1990, with the definition of civil standards (as the IEEE 802.11 [8] one) commercial wireless technologies began to appear on the market, and the research community became aware of the potential and the advantages of mobile ad hoc networking outside the military domain. Most of the existing civil ad hoc networks have been developed in the academic environment, but recently some commercially oriented solutions start to appear [9].

1.2.2 Issues

Generally speaking, an ad hoc network is a transient collection of arbitrarily located wireless devices which are able to self-organize themselves in a network to support communications among nodes. Such a network may operate in a standalone fashion providing spontaneous connectivity, or it may be connected to the larger Internet. If the nodes are mobile, they are free to move randomly and, in such a case, the network topology may change rapidly and unpredictably. Each node is able to communicate directly with any other node that resides within its transmission range and it can use its neighbor nodes as relays to communicate beyond its transmission range whitout relying on a pre-based infrastructure.
However, despite these interesting features, ad hoc networks inherit all the traditional problems of wireless communications and wireless networking:

- the wireless medium has neither absolute nor readily observable boundaries outside of which nodes are always unable to communicate;
- the wireless medium is unprotected from outside signals;
- the wireless medium has time-varying and asymmetric propagation properties;
- hidden-terminal and exposed-terminal phenomena may occur.

Beside these issues, the ad hoc networking adds a number of specific characteristics and design constraints [10]:

Multi-hop routing. Every node acts as a relay and forwards neighbors' packets to enable communications beyond the coverage area.

Self-organization and infrastructure-less. Each node operates in distributed peer-to-peer mode, acts as an independent router and generates independently data. All the network services have to be distributed across different nodes.

Heterogeneity. Each node may be equipped with one or more wireless interfaces with different communication capabilities, resulting in possible asymmetric links. In addition, each node might have a different software/hardware configuration, resulting in variability in processing capabilities.

Network scalability. Ad hoc network applications can involve large networks, as it happens in sensor and tactical networks [5]. Although scalability is critical to the successful deployment of these networks, many challenges have still to be solved [11].
Further complexities rise in presence of node mobility:

Transient network topology. Since nodes can move arbitrarily, the network topology may change frequently and unpredictably, resulting in route failures

and frequent network partitions.

Energy constrained operation. Because batteries carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node.

1.2.3 Enabling technologies

As shown in Fig. 1.1, we overview the enabling technologies for as hoc networks by resorting to the same taxonomy adopted for computer networks, i.e. basing on their coverage area:

- body area networks (BANs);
- personal area networks (PANs);
- local area networks (LANs);
- metropolitan area networks (MANs);
- wide area networks (WANs);

While metropolitan and wide-area enabling technologies for ad hoc networks are not yet available, ad-hoc single-hop BAN, PAN and LAN wireless technologies are common on the market [12].

Since these technologies can be used as building blocks for constructing multihop ad hoc networks [10], we consider them the enabling technologies for ad hoc networking. A detailed discussion of Body, Personal, and Local Ad hoc Wireless Networks can be found in [12]. Hereafter, the characteristics of these networks, and the technologies available to implement them, are in the following summarized.

A body area network has to provide the connectivity among wearable devices, that is computing devices placed on the user body, therefore the typical communicating range of a BAN corresponds to the human body range, i.e. 1-2m.

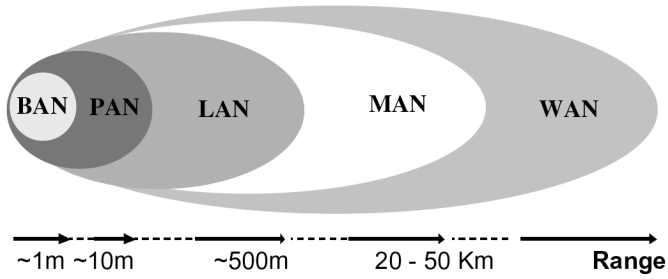


Figure 1.1: Network taxonomy based on the coverage area

Clearly, wireless technologies constitute the best solution for interconnecting wearable devices.
Personal area networks connect mobile devices carried by users to other mobile and static devices. While a BAN must assure the interconnection of one-person wearable devices, a PAN is a network composed by devices of several persons along with some environmental devices. Therefore, the communicating range is typically up to 10 m.
wireless LANs (WLANs) have a communication range typical of a single building, or a cluster of buildings, that is 100-500 m. A WLAN should satisfy the same requirements typical of any LAN, including high capacity, full connectivity among attached stations, and broadcast capability.
The success of a network technology is related with the development of networking products at a competitive price, which requires in turn the availability of appropriate standards. Currently, two main standards have emerged for ad hoc wireless networks: the Bluetooth specifications [13] for BANs/PANs and the IEEE 802.11 standard for WLANs [8].
In addition to these standards, the European Telecommunication Standard Institute (ETSI) has promoted the HIgh Performance Radio Local Area Network (HiperLAN) [14] family of standards for WLANs. Among these, the most interesting standard for WLAN is HiperLAN/2, which achieves data rates ranging from 6 to 54 Mbit/s and supports both infrastructure-based and ad hoc configurations. Along with HiperLAN, different standards have been proposed in the last years, i.e. ZigBee [15] and WiMAX [16].

1.3 Ad hoc network layer

The aim of the network layer is to provide (reliable) end-to-end connectivity among nodes. In wired networks, the internet protocol (IP) [17] assigns to nodes location-aware addresses, that is the address identify the location of the end device within the network topology, and the routing protocols exploit the topology meaning of network addresses to provide connectivity among the network.
Differently, several routing protocols for ad hoc networks resort to flat address spaces, i.e. the addresses are simply used to identify the node within the network, since assigning location-aware addresses in presence of node mobility and in absence of network infrastructure gives rise to several issues, although it simplify notably the routing process. Other routing protocols discount the use of a static addressing schema: in such a case, both an address

allocation and an address discovery procedures have to be provided to allow communications among nodes.
However, all the routing protocols for ad hoc networks exploit multi-hop communications, and thus all need to discovery all the available neighbors that can be used as relays. This task is provided by the *neighbor discovery* service, which is also in charge for estimating the available link qualities used in path selection. In fact, experimental results [18] have shown that the route metric commonly adopted in wired networks, namely the hop count, is unable to assure reliable paths in ad hoc networks.
Finally, the location discovery service, whose purpose is to dynamically map a node to its current location inside the network topology, plays a major role in the development of geographic-aware routing and multicasting protocols. However, location discovery is a broad topic and its significance stems from its wide spectrum of applications ranging from context-aware applications in ubiquitous computing to information retrieval in peer-to-peer networks.
In the following, we provide an overall overview of these network layer services.

1.3.1 Routing and forwarding

As mentioned before, the wireless propagation, the node mobility and the lack of infrastructure all introduce frequent and unpredictable changes of network topology, as shown in Fig. 1.2. Therefore, an ad hoc routing protocol should be able to maintain reliable routes despite the transient nature of the network topology. Clearly, this goal is shared by all the routing protocols, but the un-

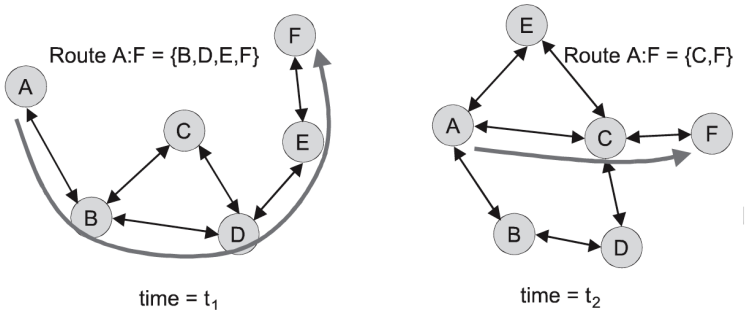


Figure 1.2: Transient network topology

derlying design assumptions of wireless communications and node mobility increase the technical challenges.
Several routing protocols and algorithms for ad hoc networks have been proposed, and their performance under various network environments and traffic conditions have been studied and compared [19, 20]. A preliminary classification of the routing protocols can be done basing on the cast mechanism, that is whether they use a Unicast, Geocast, Multicast, or Broadcast forwarding.
Broadcast is the basic mode of operation over a wireless channel, since each message transmitted on a wireless channel is generally received by all the neighbor nodes. Thus, adopting the broadcast as cast mechanism in all the nodes, namely resorting to flooding, is the simplest way of routing packets, although it causes very poor performances even in relatively small networks due to the broadcast storm problem [21].
Unicast forwarding means a one-to-one communication, i.e. each packet is transmitted to a single node. This is the largest class of routing protocols available in ad hoc networks, mainly composed by routing protocols for wired networks modified to cope with ad hoc environments.
Multicast routing protocols come into play when a node needs to send the same message, or stream of data, to multiple destinations, while geocast forwarding is a special case of multicast adopted to deliver data packets to a group of nodes situated inside a specified geographical area. From an implementation standpoint, multicast and geocasting are a form of *limited* broadcasting: messages are delivered to all the nodes that are inside a given set or region.

1.3.2 Neighbor discovery

The neighbor discovery service enables nodes to detect neighbors, that is nodes which reside within the transmission range, and to determine their link quality. The neighbor discovery service is often an integral part of MAC or address allocation protocols [22], and its procedure must be repeated from time to time to accommodate changes in the topology.
The importance of such a service has been proved by experimental measurements [18, 23], which have shown that:

- for a given transmit power, there is no deterministic relationship between distance and link quality: nodes at the same distance from the transmitter can experience widely varying packet loss rates and, in extreme cases, nearby neighbors cannot hear a node's packets but far away nodes occasionally can;
- the region around a node having a certain packet loss rate is irregularly

shaped, as illustrated by Fig. 1.3;
- there is a significant degree of asymmetric links in a network topology, which grows with the distance between nodes;
- the packet loss rate is time variable even when in the case of static node and it can experience significant short-term variations.

1.3.3 Location discovery

In legacy mobile networks [24] (GSM, Mobile IP), the presence of a fixed infrastructure led to the diffusion of two-tier schemes to track the position of mobile nodes. Examples are the Home Location Register/Visitor Location Register approach used in GSM networks, and the Home Agent/Foreign Agent approach for Mobile IP networks. Efficient implementations of these approaches use centralized servers. Nevertheless, these solutions are not suitable for ad hoc networks, and new approaches have to be found for mobility management [25].
A simple solution to node location is based on flooding the location query through the network. Of course, flooding does not scale, and hence this approach is only suitable for limited size networks, where frequently flooded packets have only a limited impact on network performance. Controlling the

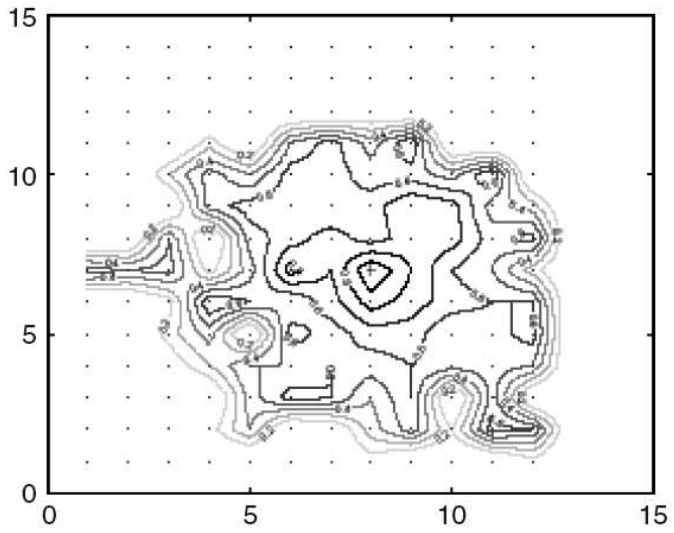


Figure 1.3: Contour plot of packet loss rates

flooding area can help to refine the technique. This can be achieved by gradually increasing, until the node is located, the number of hops involved in the flooding propagation. In this way, the location information accuracy decreases with the distance from the node but this shortcoming is balanced by the distance effect: "the greater the distance separating two nodes, the slower they appear to be moving with respect to each other" [26].
The flooding approach constitutes a reactive location service in which no location information is maintained inside the network. The location service maintenance cost is negligible, and all the complexity is associated with query operations. On the other hand, proactive location services construct and maintain inside the network data structures that store the location information of each node. By exploiting the data structures, the query operations are highly simplified.
A different approach consists of defining for each node a subset of nodes that are designed to store its location, basing on the node location or resorting to globally known hash function applied to the node identifier [27, 28, 29, 30].

1.4 Routing in mobile ad hoc networks

Dating back to the early 1980s, a large number of routing protocols designed for mobile ad hoc networks have been proposed, covering a wide range of design choices and approaches, from simple modifications of wired protocols to complex multilevel hierarchical schemes. However, many of them resort to a similar set of assumptions derived from wired networks.
For instance, most routing protocols assume that all nodes have homogeneous resources and capabilities, including also the transmission ranges. As mentioned in Sec. 1.3.2, this assumption is of course not yet valid in ad hoc networks. Moreover, although the ultimate end goal of a protocol may be operation in large networks, most protocols are typically designed for moderately sized networks [20].
Despite the different classes in which the several routing protocols can be grouped, in the following we underline some common design issues [31]:

Multi-hop routing capability To extend the limited transmission range of wireless communications, the routing protocol must be able to exploit store-and-forward techniques.

Dynamic topology maintenance Since route discovery is a expensive service, the routing protocol should deal with topology changes without wasting the resources already spent for route discovery, namely the topology changes

should have only local effects.

Loop avoidance Routing loops occur when the routing protocol select as next hop a node already occurred in the path and, since they useless waste the resources, an efficient loop avoiance mechanism should be implemented.

Minimal control overhead Control messaging consumes bandwidth, processing resources, and battery power to both transmitter and receiver side, therefore the routing protocol should be designed to minimize the number of control packets needed to operate.

Minimal processing overhead By minimizing the processing cycles, both the computational and power resources can be employed to accomplish the user tasks.

With these goals in mind, numerous routing protocols have been proposed in the last years, which can be classified in five major categories: proactive, reactive, hybrid, hierarchical and geographic based. Reactive protocols perform route discovery on-demand by flooding the network and they delay packets until the routes are set up. Proactive protocols maintain routes between all pairs of nodes, flooding information across the network whenever the topology changes, but they do not incur the delays experienced by the reactive ones. Hybrid protocols combine both reactive and proactive approaches by dividing the network in to zones: intra-zone routes are proactively mantained while the inter-zone ones are discovered on demand. Hierarchical and geographic-based protocols do not flood the network, but introduce more complexity exploiting position systems or clustering techniques.

In the following, for each class we first describe the main characteristics and then present some illustrative examples along with an high level description of each example. Further details about each protocol can be found in its respective citation and further detail about the performances achieved by the discussed protocols can be found in [20, 32, 33].

1.4.1 Proactive protocols

The proactive routing protocols for ad hoc networks are derived from the traditional distance vector [34] and link state [35] ones developed for wired networks.

The main characteristic of the proactive approach is that each node in the network maintains an update route to each other node by exchanging both periodic and event-triggered routing updates (usually referred to as *hello* packets). The periodic updates occur at specific intervals, while the event-triggered ones are transmitted whenever a change in the topology occurs and, therefore,

they introduce time-variable overhead.
The main advantage of proactive routing is that the routes are available when needed by simply looking for the destination in the routing table. Moreover, proactive routing performances do not suffer from high data session rate.
On the other hand, the main issue of proactive routing is related with the routing overhead, which can be excessive in presence of frequent topology changes and for large networks. In fact, the amount of routing state keep at each node scales as $O(n)$ where n is the number of nodes in the networks, and thus, neglecting the event-triggered updates, the overall routing overhead scales as $O(n^2)$.

Destination-Sequnced Distance Vector

The Destination-Sequenced Distance Vector (DSDV) routing protocol [36] is an implementation of distance vector routing customized for mobile ad hoc networks. DSDV utilizes node sequence numbers which are incremented at each topology change event to avoid the counting to infinity problem. The sequence numbers are also used in route selection to pick up the most recent information. More in detail, if a node learns two different paths to the same destination, it selects the one with the larger sequence number. If both have the same sequence number, the node picks up the one with the shortest hop count. If both the metrics are the same, the choice is arbitrary.
Each node maintains a route to each other node in the network, composed by the destination IP address, the destination sequence number, the next hop IP address, the hop count and the update time. Periodically, each node broadcasts a routing update composed by the destination IP addresses, the destination sequence numbers and the hop counts to neighbors, which utilize it for routing table updating by resorting to an iterative distance vector algorithm [34]. Along with the periodic updates, DSDV resorts to event-triggered updates to ensure timely discovery of topology changes.
To reduce the processing overhead and the bandwidth consumption, DSDV exploits two different types of updating: incremental and full. The former includes only the entries changed from the last full update. The latter, which requires the transmission of the whole routing table, is used only when the number of changed entries exceed the space available in a single PDU.
A mechanism for routing fluctuation damping is another improvement of DSDV with respect to wired distance vector protocols. Since routing updates are not synchronized, they can propagate along different paths at dif-

ferent rates, producing marginal route updates. To avoid such a issue, DSDV requires nodes to wait a settling time before announcing their route updates.

Optimized Link State Routing

The Optimized Link State Routing (OLSR) protocol [37] resorts to a link state routing modified to cope with the characteristic of an ad hoc network. The key feature of OLSR is the introduction of the multipoint relay (MPR) concept to limit the routing update overhead.
The MPR of a node is the minimal set of neighbors whose are able to communicate with all the two-hop neighbors, as shown in Fig. 1.4. The MPR set can be calculated according to the following algorithm [38]: each node starts with an empty MPR set and two support sets: the set N1 contains the one-hop bidirectional neighbors and the set N2 stores the two-hop bidirectional neighbors. The MPR is first populated by the nodes in N1 which are the only neighbors of some nodes in N2, and then by the remaining nodes in N1 until all the nodes in N2 have at least one neighbor in the MPR. The metric to select whose remaining nodes should be added first is based on the two-hop neighbor degree, that is on the number of neighbor nodes in N2 not yet covered by the MPR. To populate the support sets, each node periodically broadcasts an hello packet containing a list of neighbors along with their link directionality (i.e asymmetric/symmetric). By receiving a hello packet from all its neighbor, a node is able to populate both the sets N1 and N2.

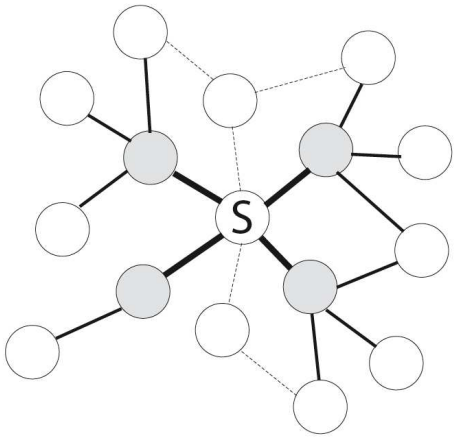


Figure 1.4: OLSR multipoint relay set

To limit the routing overhead, whenever a node broadcasts a packet only the nodes stored in its MPR are allowed to re-broadcast the packet. Further, in exchanging link state routing information, a node only lists its connections to those neighbors that have selected it as an MPR.

1.4.2 Reactive protocols

On demand routing resorts to a very different approach with respect to proactive routing.
In wired networks, connectivity patterns change relatively infrequently and resource constraints are relaxed, thus maintaining full connectivity graphs is a worthwhile expense since they assure no latency in packet forwarding. Differently, in an ad hoc network, the topology change events are frequent and control overhead is costly. For such reasons, reactive routing gives up to maintain a route between all pairs of nodes and it discovers the routes when needed (on demand), commonly by flooding the network with a route request. Clearly, a number of optimizations to reduce the overhead by limiting the search area have been proposed [39, 40].
The benefit of reactive routing is a reduction of the signaling overhead, with respect to proactive routing, particularly in network with moderate data session rate or high level of node mobility. The drawbacks are the presence of a route acquisition latency and a notable overhead in the case of high data session rates.
In the following we present two reactive routing protocols, namely the AODV and the DSR, and the reader is referred to [41] for a performance comparison analysis of both these protocols.

Ad Hoc On-Demand Distance Vector Routing

The Ad Hoc On-Demand Distance Vector (AODV) protocol [42] is a widely adopted reactive protocol for mobile ad hoc networks. Like most reactive protocols, AODV route discovery bases on a broadcast network search and a unicast reply containing the discovered path. Like DSDV, AODV relies on node sequence numbers for loop avoidance and for selecting the most recent path.
More in detail, each node maintains a routing table which stores for each route the next hop and its lifetime. If a route has not been used during its lifetime, the node discards it.
To route a packet, a node first checks if a route is available in the routing table. If so, that route can be used, otherwise the node has to start a route discov-

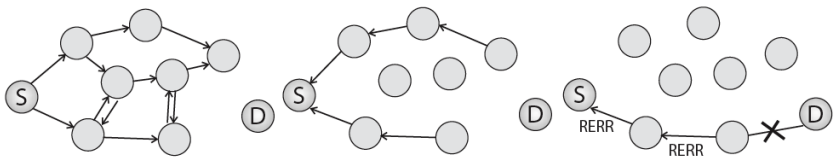
ery procedure by broadcasting a route request (RREQ) packet. The RREQ packet header stores the destination's IP address, the last known destination's sequence number (if any), the source's IP address and its sequence number. Moreover, it stores a hop counter, initialized to zero, and a RREQ identifier for duplicate detection.

When a node receives a RREQ, it first creates a *reverse route* to the source node setting as next hop the node which has broadcasted the packet. If a valid route, namely a route with a sequence number greater than the one stored in the RREQ, is available, the receiver sends the information to the RREQ source with a reply packet (RREP), otherwise it simply re-broadcasts the RREQ, as shown in Fig. 1.5-a. The sequence number condition ensures that only routes most recent than the source requests are exchanged and , moreover, it represents a loop avoidance mechanism [42].

Unlike the RREQ packets, the RREP ones are unicast forwarded hop by hop toward the destination (Fig. 1.5-b), since the needed routing information is available at each intermediate node in the reverse routes. Moreover, each receiving node creates a *forward route* entry using the node from which it received the RREP as the next hop toward the destination. Such an entry will be eventually used for data forwarding.

Once a route has been established, it is maintained as long as needed by means of route error messages (RERRs). When a link failure occurs between two nodes belonging to an active path, the node closest to the source unicast sends a RERR packet to all its neighbors that were using that link, whose in turn propagate the REER along the reverse routes, as illustrated in Fig. 1.5-c.

Beside the operational details, AODV implements several optimizations and optional features [43]. To improve the protocol performance and reduce overhead, source nodes can utilize an expanding ring search to route discovery, by modifying the TTL field of the RREQ packet. In such a way, if a route toward the destination is available in the neighborhood of the source node, a network



(a) RREQ Broadcast (b) RREP Propagation (c) RERR Message

Figure 1.5: AODV route discovery and maintance

flooding can be avoided. Another optimization regards the route maintaining: in the case of link failure, a node tries to local repair the failure finding another route. Only in the case of local repair failure, the node sends a RERR packet.

Dynamic Source Routing

The Dynamic Source Routing (DSR) protocol [44] is a reactive protocol based on the source routing approach: each packet stores the whole path in the header allowing so a simpler forwarding process with respect to the hop-by-hop forwarding exploited by AODV. As illustrated by Fig. 1.6, both the route request (RREQ) and the route reply (RREP) packets accumulate the forwarders' IP addresses at each hop so that, once a route has been discovered, the source knows the entire route.
DSR shares with AODV some common mechanisms: the RREQ packets are broadcasted by each receiving node until a route have been discovered, while the RREP packets are unicast forwarded resorting to the reverse route information collected by the RREQs. Moreover, both maintain the routes resorting to RERR packets.
However, unlike AODV, each node maintains several routes toward the same destination which can be used in the case of link failures. In other words, DSR exploits a multi-path routing strategy. Moreover, the routes have no lifetime: once a route has been discovered, it remains valid until it breaks. Finally, DSR enables nodes to promiscuously listen to control packets not addressed to themselves. In such a way, nodes can utilize the source routes carried in both DSR control messages and data packets to gratuitously learn routing information for other network destinations.

(a) RREQ Broadcast (b) RREP Propagation

Figure 1.6: DSR route discovery

1.4.3 Hybrid approaches

The characteristics of proactive and reactive routing protocols can be integrated in various ways to form hybrid networking protocols, which exhibit proactive behavior given a certain set of circumstances, while resort to reactive routing given a different set of circumstances. Several protocols and techniques belong to this class, and in the following we present the Zone Routing Protocol, which is the most notable one.

Zone Routing Protocol

The Zone Routing Protocol (ZRP) [45] has been proposed to reduce the control overhead of proactive routing protocols as well as to decrease the latency caused by routing discover in reactive ones by resorting to the *zone* concept. A zone of a node is defined as its k-neighbourhood, that is a zone is the set of nodes within k hops (typically $k = 3$), and Fig. 1.7 shows an example. Each node utilizes proactive routing within its zone and reactive routing outside of it.

For intra-zone routing, ZRP defines the Intrazone Routing Protocol (IARP), a link-state protocol that maintains up-to-date information about all nodes within the zone, while the Interzone Routing Protocol (IERP) is used for discovering routes toward destinations locate outside of the zone. The IARP resort to the peripheral node concept, namely the one whose minimum hop distance from the considered node is the zone radius k. With reference to Fig. 1.7, A, B, C,

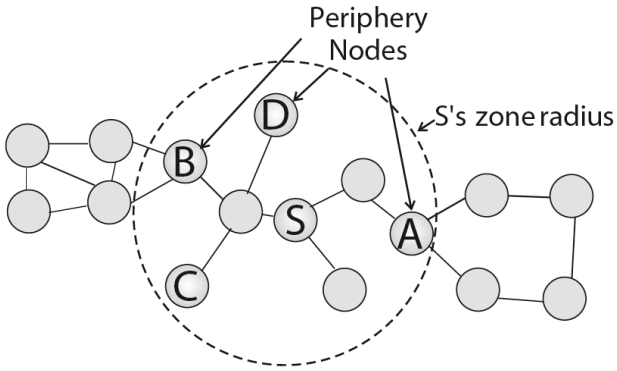


Figure 1.7: ZRP zone

and D are the peripheral nodes of S.
For inter-zone route discovery, the notion of *bordercasting* is introduced: once a source determines that the destination is not located inside its zone, it sends a query message to its peripheral nodes exploiting the intra-zone knowledge. After receiving the query, the peripheral nodes, in turn, check whether the destination lies within their zone and the procedure iteratively continues until either the destination is located or the entire network is searched.

Fig. 1.8 illustrates an example of the bordercast discovery procedure: by resorting to IARP, S learns that X is not located within its zone. Thus, it bordercasts the query message to its peripheral nodes which bordercast the query message to their peripheral nodes as well. The solid circles in the figure represent the forward propagation of the query messages to peripheral nodes. Eventually, node G discovers X within its zone, and then it unicast sends a reply back to node S.
To improve query efficiency, ZRP exploits a random query processing delay between query reception and query forwarding to reduce chance of collisions during forwarding. Other optimizations are used by ZRP to reduce the messaging and processing overhead [46] and other ones are introduced in a subsequent version of ZRP, namely ZRPv2 [47].

1.4.4 Clustering protocols

As illustrated before, traditional ad hoc protocols exploits *flat* routing, achieving low scalability properties. In fact, in the worst case where a node must track every other node in the network, the amount of routing information exchanged by nodes grows as $O(n^2)$, independently of the routing approach (proactive,

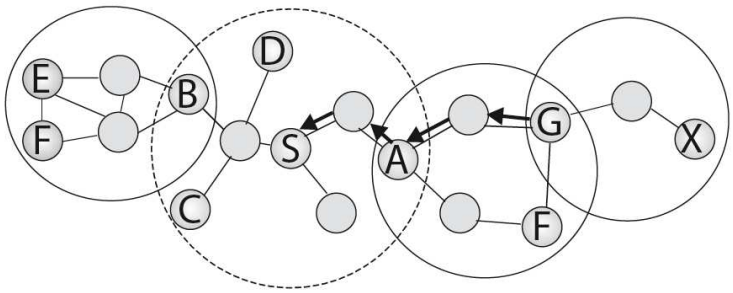


Figure 1.8: ZRP route discovery

reactive or hybrid one).

To overcome such a issue, a number of routing protocols which groups nodes in sets, often called clusters, have been proposed. The nodes can be clustered according several criteria, which are commonly based on either location [48, 25] or functionality [49, 50] and in this section we present a survey of the characteristics and algorithms for clustering routing without examining individual protocols.

The physical properties of clusters vary among the protocols as shown in Fig. 1.9, namely they can be either overlapping or completely disjoint. Usually, the cluster boundaries are based on the transmission range or on the neighborhood of the cluster leaders and the nodes located within the boundaries of multiple clusters, namely the gateways, are in charge of inter-cluster routing.

Further, an one-level hierarchy can be created, or recursive multilevel hierarchies are also possible. Finally, the control within a cluster can be held by a cluster leader, which typically processes control packets on behalf of its member nodes, or the procedures can be completely among the cluster nodes. It is also possible for cluster leaders to form a routing backbone within the network [50].

Cluster leaders are typically initialized through some distributed algorithm based, for instance, on node properties like the node ID, the number of neighbors, the transmission range, or resorting to “first come, first elected” approach [51]. Along with the leader election, there must be leader revocation algorithm, differently the number of leaders will continual grow due to topology changes.

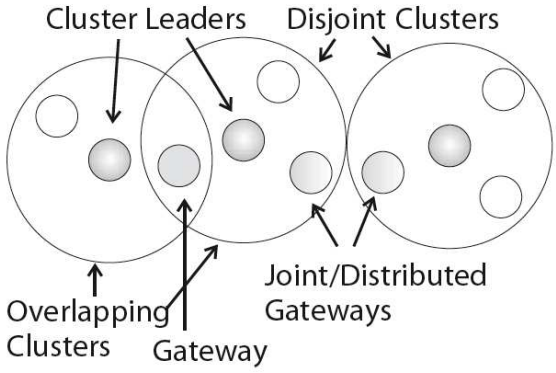


Figure 1.9: Cluster topologies

Most commonly, when two leaders come within direct transmission range of each other, one of the leaders must give up its leader status according to the same metric used for leader election.

The cluster-based routing has two key benefits. The former is that it enables the hierarchical routing, achieving so more resilience to link failures. As example, let us consider the network shown in Fig. 1.10 and let us suppose that the node S has to forward a packet to node D. One of the previously discussed flat routing protocols might find the following path:

$$S \rightarrow C_1 \rightarrow G_1 \rightarrow C_2 \rightarrow D \tag{1.1}$$

while a hierarchical one will discover this:

$$S \rightarrow C_1 \rightarrow C_2 \rightarrow D \tag{1.2}$$

Since a clustering protocol tracks paths at the cluster level, without specifying the intermediate node, a higher freedom degree is available to face against link failures. Further, decreasing the number of route repairs decreases as well the amount of control overhead generated in the network.

The latter benefit of clustering routing is that the hierarchy can be used to implement hierarchical addressing schemes, based on cluster membership. For instance, in the network shown in Fig. 1.11, a node z is a member of a cluster y. In that case, its address may be y.z. If the hierarchy consists of multiple levels, then cluster y might in turn be within a larger cluster x. In this case, the node's address would be x.y.z.

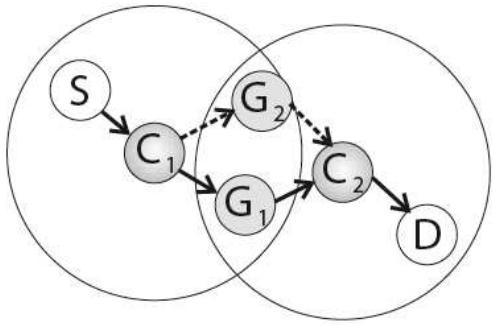


Figure 1.10: Cluster routing

Hierarchical routing protocols have many clear advantages. They improve route robustness by increasing routing flexibility; routes that are recorded between clusters, unlike the ones recorded between nodes, have more routing options (a higher freedom degree), and, hence, can be repaired more easily. Increasing route robustness leads to an increase in route lifetimes, thereby resulting in fewer route reconstructions, less control traffic from route repairs, and increased data delivery.
However, there are also disadvantages that many hierarchical routing protocols suffer from. To create and maintain the clusters, many clustering protocols require periodic overhead to maintain current information about cluster memberships and gateway availability and, moreover, the centralization of routes through cluster leaders results in network congestion and longer routes.

1.4.5 Geographical approaches

Geographical approaches resort to geographical information to simplify the routing process [26][3, 25, 33, 58, 59], usually resorting to coordinates either absolute, as the ones provided by GPSs, or relative with respect to some reference points. The use of geolocation information can prevent network-wide searches for destinations, as either control packets or data packets can be sent in the general direction of the destination if the recent geographical coordinates for that destination are known. However, all nodes must have continual access to their geographical coordinates as well to the destinations' coordinates, moving thus the complexity from the routing to the coordinates dissemination.

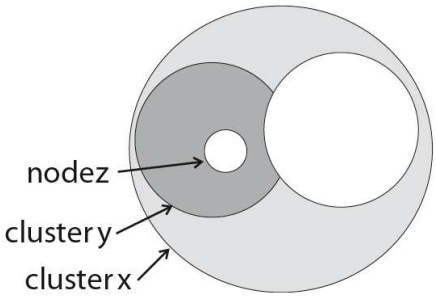


Figure 1.11: Hierarchical addressing

Location-Aided Routing

The Location-Aided Routing (LAR) protocol [39] exploits absolute geographical coordinates and resorts to reactive routing for forwarding the route requests to the previously known location of the destination.
The protocol defines two areas: the expected zone and the request zone. The former is the area in which the destination is most likely to be discovered and the latter is the area in which the route request for the destination should propagate.
By knowing both the location (x_d, y_d) of the destination at the time t_0 and an estimate of its velocity $v(t_0)$ at the same time, the expected zone at the time t_1 is estimated as the circle of radius $r = v(t_0) * (t_1 - t_0)$ centered at (x_d, y_d) and the request zone is defined as the the smallest rectangle that contains both the expected zone and the source node (Fig. 1.12).
The basic route discovery procedure of LAR resorts to broadcasted route request (RREQ) packets (like reactive routing), which are allowed to propagate only within the request zone if the destination position is known. Differently, the algorithm defaults to basic flooding.
The size of the request zone is a trade-off between control overhead and probability of finding the destination. A small request zone runs the risk of not including the area in which the destination is currently located or not including the whole path between the source and the destination. On the other hand, if the request zone is too large, the control overhead reduction will be minimal.

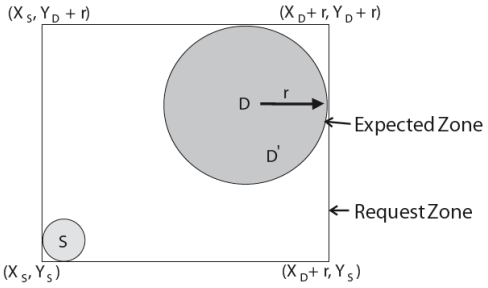


Figure 1.12: LAR zones

Chapter 2

Augmented Tree-based Routing

This chapter presents a new routing protocol for ad-hoc networks, which resorts to both a distribute hash table (DHT) and a location-based addressing schema in order to assure a scalable routing service. The protocol, referred to as Augmented Tree-based Routing (ATR), can be used with any link layer technology but in the following we consider an implementation based on IEEE 802.11 technology operating on a hostile channel, namely presence of long-term fading, additive thermal noise and interferences. The performances have been evaluated by means of numerical simulations across a wide range of environments and workloads. The results show that ATR outperforms traditional routing protocols whenever the number of nodes grows, assuring satisfactory performances also for large networks operating in presence of hostile channels and moderate node mobility.

2.1 Introduction

In the last ten years, ad hoc technologies have tremendously grown. Most of the research has mainly regarded relatively small networks and has been focused on performances and power consumption related issues. More recently, due to the importance of ad hoc paradigm in applications involving a large population of mobile stations interconnected by a multi-hop wireless network [11], great attention has been devoted to self-organizing routing protocols with satisfactory scalability requirements.
However, most of the proposed protocols, regardless of the belonging class (reactive, proactive, and hybrid), do not scale efficiently when the number of nodes grows [20, 21] mainly since they have been proposed for wired net-

works and modified to cope with ad hoc scenarios [3]. More specifically, they are based on the assumption that node identity equals routing address, that is, they exploit static addressing which of course is not yet valid in ad hoc scenarios (Sec. 1.3).
Recently, some routing protocols have exploited the idea of decoupling identification from location, by resorting to distribute hash table services, which are used to distribute the node's location information throughout the network. Several proposals based on this approach have been recently presented, and they can be classified according to the lookup model in two main groups. The former requires the knowledge of the geographical node's position which can be provided by a central infrastructure such as the GPS (a survey can be found in [52]), and clearly this solution is not suitable in the case of self-organizing networks. In the latter group, the information stored in the DHT is the node network address, which reflects the node topological position inside the network. In few words, the proposals belonging to this group introduce a logical and mathematical structure on the address space based on connectivity between nodes. After that the dynamic address of a node has been retrieved by the lookup procedure in the DHT, the routing is performed using the topological mean of the addresses, resembling the routing procedure performed for wired networks [49, 50, 53, 54, 55, 56]. All the above cited schemes are hierarchically organized and exploit a tree structure for both the address space management and routing. Although this structure offers a simple and manageable procedure, it lacks for robustness against mobility and/or link failure and exhibits unsatisfactory route selection flexibility [52]. It is worthwhile to underline that some of them [54, 55, 56] do not deal with the implementation of the DHT service, which is a key process of the whole routing protocol. In order to improve the performance, more complex structures can be used, like ring [57, 58]. However, in such a case the increased complexity in the address allocation mechanism could discourage their use in presence of channel hostility and very large networks.
In this chapter, following the work developed in [59, 60], we give a contribution toward such approach by focusing our attention on the problem of implementing a scalable routing protocol, namely the Augmented Tree-based Routing protocol, whose performances are competitive with those of other widely adopted protocols [42, 44, 36].
ATR, according to [55], resorts to a location-aware addressing schema: each

node has a permanent unique *identifier*[1], that identifies the node in the network, and a transient *network address*, which reflects the node's location inside the topology.
ATR organizes the nodes with a tree-based address structure, defining a xor-like *overlay distance* between nodes based on their network addresses. Each node stores routes toward sets of nodes, and the cardinality of the sets depend on the overlay distance between the source and the destination addresses. Thus, ATR adopts a hierarchical approach [1], which allows one to reduce the routing state information stored by each node with respect to a flat approach from $\Theta(n)$ to $\Theta(\log(n))$, where n is the overall number of nodes in the network. Differently by previous work [49, 50, 53, 54, 55, 56], ATR resorts to a multi-path strategy: the address space structure is *augmented* by storing multiple routes toward each set of nodes. With regards to the address space overlay, the multi-path approach improves the tolerance of the tree structure against mobility as well as channel impairments while, with reference to the packet forwarding, it improves the performance by means of route diversity.
As mentioned before, the mapping between node identifiers and network addresses is provided by a DHT system. Differently from traditional application-layer DHTs which assume the presence of an underlying network routing protocol providing connectivity among nodes, the DHT of ATR is implemented directly on top of the layer-2 in order to provide its services to the routing procedure. Moreover, while in application-layer DHTs the communications are established independently of the physical node position and the redundancy allows one to provide reliable services, a routing-layer DHT service has to take into account the network topology and to avoid redundant transmissions to minimize the overhead. Consequently, the DHT system of ATR relies on the physical neighbors and the information, namely the network addresses, is distributed across the network without redundancy.
To test the routing scalability of ATR, numerical simulations on 802.11 technology have been carried out and ATR performances have been compared with those of a representative set of routing protocols. It is worthwhile to underline that ATR can be accommodated with slight modifications to operate over any link layer technology and, moreover, it does not require any change in the upper layers. Differently from most of the traditional performance comparisons [55, 56, 61, 62] that adopt a deterministic channel model, we have evaluated the performances by resorting to a more realistic channel model, namely a

[1]The assumption of uniqueness has been made only for the sake of simplicity: ATR can be easily generalized whenever multiple network cards are available at the same node.

model which accounts for long-term fading effects, additive thermal noise and interferences.

2.2 System architecture

ATR resorts to a network-layer architecture in which each node has a permanent unique *identifier* (as an IP address one), which identifies the node in the network, and a transient *network address* that reflects the node's topological location inside the network. Nodes acquire network addresses by listening for the routing update packets exchanged by neighbors.
The network addresses are strings of l bits, thus the ATR's address-space structure can be represented as a *complete binary tree* of $l+1$ levels, that is a binary tree in which every vertex has zero or two children and all leaves are at the same level (Fig. 2.1-a). In the tree structure, each leaf is associated with a network address, and a inner vertex of level k, namely a *level-k subtree*, represents a set of leaves (that is a set of network addresses) sharing an address prefix of $l-k$ bits. For example, with reference to Fig. 2.1-a, the vertex with the label *01x* is a level-1 subtree and represents the leaves *010* and *011*. Let us define as *level-k sibling* of a leaf as the level-k subtree which shares the same parent with the level-k subtree the leaf belongs to. Each address has l siblings at all and each other address belongs to one and only one of these siblings. Referring to the previous example, the vertex with the label *1xx* is the level-2 sibling of the address *000*, and the address *100* belongs only to this sibling.

In Fig. 2.1-b, the address space is alternatively represented as an *overlay network* built upon the underlying physical topology. Its tree-based structure offers simple and manageable procedures for address allocation, avoiding to rely on inefficient mechanisms like flooding. Usually, these attractive proper-

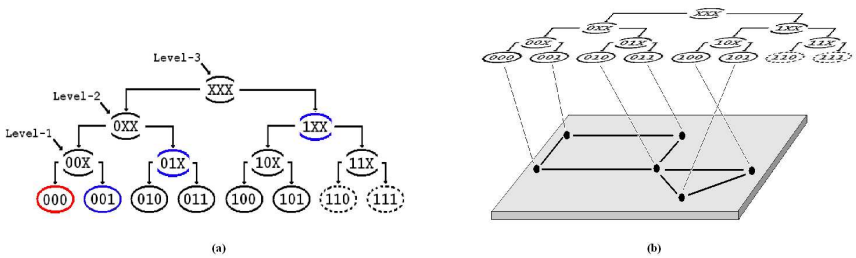


Figure 2.1: ATR address space overlay

ties are obtained at the price of low fault-tolerance as well as traffic congestion vulnerability since there exists only one path between any pair of nodes [52]. Moreover, the address overlay embeds only a partial knowledge about the physical network topology, since only a subset of the available communication links is used for the routing [63]. For such reasons, we propose to *augment* the tree structure by storing in the routing tables multiple next hops towards the same sibling, that is by resorting to multi-path routing, with no impact on the routing overhead since the routing update packet (e.g. hello packet) sizes do not depend on the number of multiple paths stored in the routing table (See Section 2.3.3).

The ATR routing procedure is an iterative one through the address tree, based on a hierarchical form of multi-path proactive distance-vector routing. ATR routing tables have l sections, one for each sibling. The k-th section stores the available routes, namely the next hops, towards a node belonging to the level-k sibling. According to the Fig. 2.1-a, the node *000* has three sections in its routing table. The first stores the routes towards the node *001*, the second towards a node belonging to the sibling *01x* and the last towards nodes belonging to the sibling *1xx*. To route a packet, a node compares its network address with the destination one, one bit at time starting with the most significant (left-side) bit, say the l-th. If the i-th bit is different, the node forwards the packet towards one of the routes stored in the i-th section. With reference to Fig. 2.1-a, if the node *000* has to send a packet to the node with the address *101*, then it will forward the packet to one of the next hops (if any) stored in the third section. The hierarchical feature of ATR is so based on the concept of sibling and it allows nodes to reduce the routing state information, as well as the routing update size. Moreover, it assures that routes toward far nodes remain valid despite local topology changes occurred in the vicinity of these nodes.

Since the routing process is based on the network addresses, they have to be efficiently distributed across the network. We implement this service resorting to a distribute hash table. The core of a DHT service is a globally known *hash* function $h(\cdot)$: if the node *s* with identifier ID_s has to communicate with the node *d* with identifier ID_d, *s* has to request ADD_d to the node with the address $ADD_p = hash(ID_d)$, which is in charge of storing the mapping $< ID_d, ADD_d >$.

2.3 Augmented Tree-based Routing

The architecture of ATR is represented in Fig. 2.2, where the *address allocation process* allows nodes to acquire a valid network address, while the *route discovery process* is responsible of both routing-table building and updating. The services provided by these processes are exploited by the *packet forwarding process*, which is in charge of both choosing the best route and forwarding the packets through. The *address discovery process* supplies the mapping between identifiers and network addresses, by resorting to the *packet forwarding* services. Finally, the *link quality estimation* process assesses the quality of the available links, supporting so the other processes.
ATR uses five types of control packets. The first type, say the *hello* packet, is locally broadcasted and it is used by the address discovery, the route discovery and the link quality estimation processes. The *Dynamic Address UPdate* (DAUP), the *Dynamic Address ReQuest* (DARQ) and the *Dynamic Address RePly* (DARP) ones are unicast packets used by the address discovery process. Finally the *Dynamic Address BRoadcast* (DABR) packet is locally broadcasted by the address discovery process.

2.3.1 Address Allocation Process

The address allocation process provides distributed assignment of network addresses to nodes and the whole procedure is based on the locally broadcasted hello packets.
Node mobility, concurrency and fading contribute to address duplication as well as prefix constraint violation (defined in the following). Differently from the stateless approach [64], which usually resorts to flooding mechanisms in

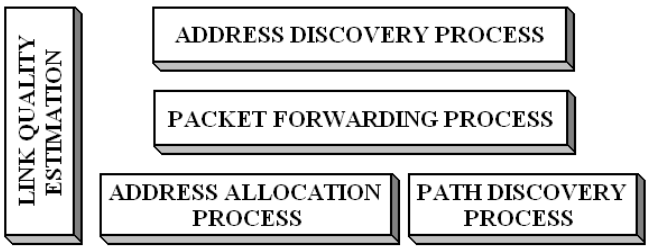


Figure 2.2: ATR functional structure

order to perform the duplicate address detection, ATR exploits a *stateful* approach based on *multiple disjoint allocation tables* [65, 66]. Each node is responsible for a quota of the address space, namely a subtree, and different nodes manage disjoint quotas. When a new node joins the network, it listens for the hello packets exchanged by neighbors in order to acquire a valid and available address. Supposing that the selected neighbor was managing a level-k subtree, which is composed by two level-$k-1$ subtrees, then, along with the address, the new node takes the control over the level-$k-1$ subtree to which the selected address belongs to.

The proposed procedure guarantees that nodes, which share the same address prefix, form a connected sub-graph in the network topology; we refer to this property as the *prefix constraint* one. This procedure was first proposed in [55] and, in [59], two convergence issues have been recognized and solved as shown in the following.

The detection of duplicate addresses resorts to the subtree identifier concept: we define as *subtree id* the lowest node identifier of all the nodes whose network addresses belong to that subtree. The subtree ids allow ATR to detect the presence of the same address prefix in two disconnected parts of the network. If this occurs, then the prefix constraint has been violated and a duplicate address can be present in the network.

More in detail, when a node switches on, it chooses a default network address and it periodically executes the procedure shown in Algorithm 1. It first

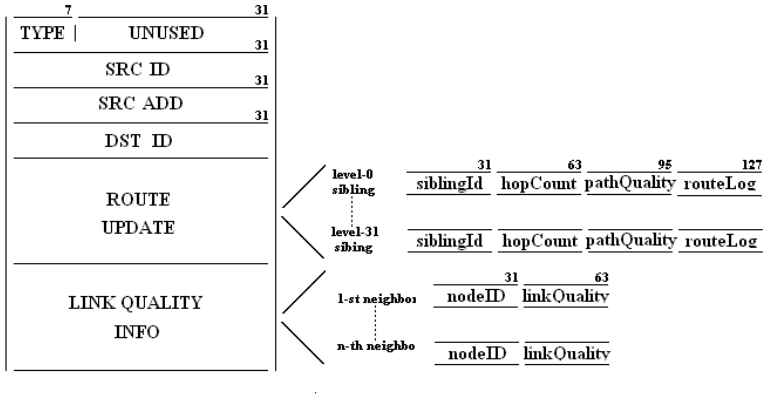


Figure 2.3: ATR hello packet

Algorithm 1 periodicCheck()

```
neighborListPurge()
routingTableClear()
if not validateAddress() then
  address = selectAddress()
end if
for each neighbor neigh do
  routeUpdate(neigh)
end for
```

Algorithm 2 validateAddress()

```
for each neighbor neigh do
  if my.add = neigh.add then
    if my.id > neig.id then
      return FALSE
    end if
  else
    i = overlayDistance(my.add, neigh.add)
    if my.siblingId(i) > neigh.siblingId(i) then
      return FALSE
    end if
  end if
end for
return true
```

purges the neighbor list from the neighbors whose link quality is below a certain threshold according to the metric described in Section 2.3.2. Then, ATR clears the routing table before selecting a new address, avoiding so that the address selection procedure resort to outdated information. This is the first convergence issue.

After that, the node checks if its current network address is valid by means of the procedure shown in Algorithm 2. For each neighbor, the node first checks if its address and the neighbor's one are the same. If so, an address duplication has been detected and only the node with the higher[2] id has to change its address. Then, the node computes the sibling to which the neighbor address belongs to, by recognizing the most significant bit that differs between the node

[2]The *highest-id* metric is chosen for the sake of simplicity, although the ATR could be improved with more sophisticated metrics.

Algorithm 3 selectAddress()

```
oldAdd = my.add
for each neighbor neigh ordered by free address space quotas do
  i = freeSection(neigh.routingUpdate)
  for each free section k, with k ≤ i do
    add = neigh.add
    my.add = add.flip(k)
    routeUpdate(neigh. routingUpdate);
    if validateAddress then
      break
    else
      my.add = oldAdd
    end if
  end for
end for
```

address and the neighbor one. If the related sibling id, stored in node routing table, is higher then the one stored in the neighbor's route update, the presence of the same address prefix in two disconnected parts of the network has been detected and the node has to change its address. The way of computing the sibling id is the second convergence issue recognized by [59] as detailed in Section 2.3.3.

If the address validation process fails, the node acquires a new address as illustrated in Algorithm 3. First, the node sorts its neighbor set according to the cardinality of the address subtree managed by each neighbor. This is practically done by computing the level of the highest empty section in their route updates, since an empty section represents a subtree available for address allocation. Starting from the neighbor with the highest cardinality, the node selects an address in the highest available section. Then, the node updates its routing table with the neighbor route update and it tries to validate the address as detailed above. Once a valid address has been acquired, the procedure ends.

Let's make an example of the address selection mechanism. Supposing the node A switches on: it chooses the default network address *000*. When node B switches one, the node with the highest id, says B, has to change its address. Since A's routing table is empty, B picks up an address in the highest available entry in A's route update. Since this entry is related to the level-*2* sibling of address *000*, namely the subtree *1xx*, B selects the address *100*, according to our implementation, and it tries to validate it.

2.3.2 Link Quality Estimation Process

The link quality estimation process provides two services. The former allows the packet forwarding process to choose the routes assuring the highest throughput. The latter enables the address allocation process to converge to a steady state also in presence of time-variant channels. More specifically, since the node address has to be validated against the neighbor ones, the presence of propagation instability forces a node to change continuously its network address also when the network topology does not change (namely in presence of static nodes). Let us note that the last issue, together with the one due to the presence of link asymmetry [67], have not been recognized in the previous works [55, 59].

To estimate the link quality, ATR resorts to the hello packets and to a moving average (MA) filtering. Each node locally broadcasts the hellos with an average period τ (one second in our implementation) jittered up to $\pm\tau/k$ for each period; thus we can model the hello reception events as binary independent random variable $x(n) \in \{0, 1\}$. Since the channel is time-variant, the probability that the node j receives an hello from the node i depends on the time, namely $P(x_{i\to j}(n) = 1) = p_{i\to j}(n)$. At the time n, the node j evaluates by a MA filtering the link quality $q_{i\to j}(n)$ for the packets received by the neighbor i, according to:

$$q_{i\to j}(n) = \sum_{m=0}^{M-1} b(m) x_{i\to j}(n-m) \tag{2.1}$$

where $b(m)$ is the weighting factor.

Each node j broadcasts its estimated link qualities $q(i \to j)$ with the hello packets (Fig. 2.3). This allows neighbor i to retrieve the link quality $q_{i\to j}(n)$ and thus to compute the bi-directional link quality $q_{i,j}(n)$ as[3]:

$$q_{i,j}(n) = q_{i\to j}(n) \times q_{j\to i}(n) \tag{2.2}$$

By means of a link quality threshold, we assure that each node acquires a steady address, since neighbors, whose link quality $q_{i,j}(n)$ does not exceed the threshold, do not take part in the address validation process. In the same way, neighbors with insufficient link quality are not used to update the routing tables, solving so the problems due to the presence of asymmetric links.

As mentioned before, the link quality is also used in the routing process to

[3]This link quality metric is adopted because a 802.11 link layer with ARQ mechanism is considered in the protocol implementation. Clearly, the estimation mechanism can be easily adapted to different link layer technologies.

compute the path cost, by means of the expected transmission count (ETX), first proposed in [68]. This path-cost metric estimates the expected number of packet transmissions (included retransmissions) required to successfully deliver a packet to the ultimate destination. If nodes i and j are neighbors, the estimate $c_{i,j}$ of ETX needed to delivery a packet through the link $l(i,j)$ at the time n is:

$$c_{i,j}(n) = \frac{1}{q_{i,j}(n)} \tag{2.3}$$

whereas the estimate of ETX on whole route $R(s,d)$is:

$$c_{s,d}(n) = \sum_{l(i,j) \in R(s,d)} c_{i,j}(n) \tag{2.4}$$

2.3.3 Route Discovery Process

The route discovery process maintains a consistent routing state through the network by updating the routing tables with the information broadcasted by nodes with the hellos.
A routing table is made up by l sections (where l is the network address length) and the k-th section contains several (if available) routes, namely entries, toward nodes which addresses belong to the level-k sibling. Each entry contains four fields: the network address of the next hop, the sibling id, the path cost (computed according Section 2.3.2) and the route log (defined in the following). Differently, a routing update contains no more than l entries, namely one entry for each sibling, and each entry contains only three fields: the sibling id, the path cost and the route log. If a node stores multiple routes toward the same sibling, it will only record in the routing update the information concerning the best route, according to the path cost. In other words, a routing update notifies neighbors about the presence of routes towards a sibling, regardless the paths that the packets will be forwarded through, allowing so ATR to adopt the multi-path approach with no communication overhead with respect to the traditional shortest-path one.
Every route discovery process requires a loop detection mechanism to avoid that the information stored in a route update visits the same node more times. ATR exploits the address space overlay in order to implement an efficient and scalable loop avoidance mechanism. As mentioned before, each entry of a routing update contains a field referred to as route log, that is a bit array with the same length of a routing address. The i-th bit of the route log indicates that the routing update reaches the node via the level-i sibling. The loop avoidance

Algorithm 4 routeUpdate(neigh.routingUpdate)

```
i = overlayDistance(my.add, neigh.add)
if neigh.siblingId(i-1) ≤ my.siblingId(i) AND 1/neigh.linkQuality < maxCost then
    routeLog.reset() //clearing all the bits
    routeLog.set(i) //setting the i-th bit
    table.section(i).addEntry(neigh.add, neigh.siblingId(i-1),
    1/neigh.linkQuality, routeLog)
end if
for each k-th entry in the route update, with k ≥ i do
    if neigh.siblingId(k) ≤ my.siblingId(i) AND 1/neigh.linkQuality +
    ngh.entry(k).routeCost < maxCost then
        routeLog = neigh.entry(k).routeLog
        routeLog.clear(1,i-1) //clearing the first i − 1 bits
        routeLog.set(i)
        table.section(k).addEntry(neigh.add, neigh.siblingId(k),
        1/neigh.linkQuality + ngh.entry(k).routeCost, routeLog)
    end if
end for
```

mechanism blocks an entry to re-enter in a level-i sibling if the i-th bit is set, as illustrated by Algorithm 4.

More in details, a node, after the validation of its address, updates its routing table with the information stored in the hellos received by the neighbors according to Algorithm 1. For each neighbor, at first the *routeUpdate* procedure (Algorithm 4) computes the sibling to which the neighbor address belongs to, say the i-th. Then the neighbor is used as next hop for for each level-k sibling, with $k \geq i$, if the following two conditions are satisfied. The former is that the routing update agrees with the prefix constraints, and the latter is that the cost associated with the route does not exceed a certain threshold.

Let us suppose, for example, that the node *000* receives from the neighbor *010* a route update, which has three entries related with the siblings *011*, *00x* and *1xx* respectively. The information stored in the first entry of the routing update is useless to the node owing to the hierarchy present in the routing table. Moreover, also the information stored in the second entry is useless, since in this case such information is already owned by the node *000*. Differently, the information stored in the third entry can be used to set up a route toward the level-3 sibling *1xx* through the neighbor *010*. Finally, the neighbor itself is

used to set up a route toward its sibling, i.e. the level-2 sibling *01x*.
As mentioned in Section 2.3.1, the way of computing the sibling ids affects the address allocation convergence. We propose to compute the neighbor's sibling id for the i-th sibling id by choosing the lowest node identifier among the first $i-1$ entries of its routing update and the neighbor's identifier. According to the previous example, this means that the node *000* singles out the sibling id of the neighbor *01x* for the second sibling by computing the lowest identifier among the neighbor identifier (associated with the neighbor address *010*) and the sibling id stored in the first entry (associated with the address *011*). Therefore in this example the sibling id is the lowest identifier for the sibling *01x*. Differently, the procedure stated in [55] computes the sibling id by considering also the second entry of the routing update, which is related to the addresses belonging to the sibling *00x*, that is by computing the lowest identifier for the sibling *0xx*, although the routing update involves only the sibling *01x*. In [59] it has been shown by numerical simulations that the procedure [55] does not converge.
In contrast, the neighbor's sibling id for the k-th sibling, with $k > i$, is simply the lowest identifier among the neighbor identifier and the sibling ids stored in the first k-th entries of the routing update.

2.3.4 Packet Forwarding Process

The proposed procedure for choosing the path to forward the data packets is described by Algorithm 5.
According to such a procedure, the route is singled out by taking into account the hierarchical feature of ATR, that is by choosing, as next hop, the neighbor which shares the longest address prefix with the destination. If there are multiple neighbors sharing the longest address prefix, the node will select the one with the lowest route cost.
As example, let us suppose that the node *000* has to forward a packet towards the node *110*. Since the destination belongs to the level-3 sibling, namely the *1xx*, the node looks for routes in the third section of its routing table. Moreover, we suppose that this section stores two entries: the former through the next hop *010* and the latter through *100*. The node selects, as next hop, the node *100*, regardless of the costs associated with the routes. We recall that the address prefix rule is due to the hierarchical architecture of ATR routing tables: the closer a neighbor is to the destination in terms of address prefix, the more the routing information owned by the neighbor is thorough. According to the previous example, the neighbor *010* has just a section for all the four nodes

Algorithm 5 forwarding(dst.add)

```
//dst.add is the destination routing address
i = overlayDistance(my.add, dst.add)
nextHop = NULL
distance = l //l is the bit length of a network address
cost = maxCost
for each k-th section, with k ≥ i do
  for each entry in routing table towards the i-th sibling do
    if (overlayDistance(dst, entry.nextHop) < level OR (sibling(dst, entry.nextHop) == level AND entry.routeCost < cost)) AND entry.notFailed then
      nextHop = entry.nextHop
      level = (dstAdd, entry.nextHop)
      cost = entry.routeCost
    end if
  end for
end for
return nextHop //Returning the next hop towards the peer
```

belonging to the sibling *1xx*, while the neighbor *100* has a section for the node *101* and another one for the two nodes in *11x*.
Differently, if we assume that the two entries stored by the node be through the next hop *010* and *011* respectively, and thus both share the same address prefix, the node will select the one with the lowest route cost, i.e. the lowest ETX value.
Thanks to its multi-path feature, ATR can exploit the route diversity in packet forwarding: since multiple routes are available, when one fails due to node mobility and/or link local network congestion, such a route is checked off as failed and the node forwards the packet through another next hop, if available. Clearly, although such a strategy can lead to significant delays in packet delivery, it avoids to waste communication resources [20].

2.3.5 Address Discovery Process

This sub-section presents the Address Discovery Process, which supplies the mapping between node identifiers and network addresses resorting to a distributed hash table (DHT). This is a key process of the ATR framework and it requires a moderate amount of sophistication, since bringing the DHT concept

from the application-level down to the network one arises new issues.
Application-level DHTs assume the presence of an underlying network protocol which assures reliable communications between nodes [57, 58, 69, 70]. Moreover, most of the proposed works build a logical space, namely an *overlay network*, in which the proximity concept has no relation with the physical neighborly. Finally, data replication is commonly adopted to face against node failure and often to distribute the load through the network [71].
Differently, network-level DHTs need an overlay network which relies on physical connectivity for communication in order to reduce the average path length of query forwarding [52]. Moreover, they should avoid data replication, since it introduces overhead due to multiple copies of the same information traveling around the network. Finally, network-level DHTs have to implement fault-tolerant strategies to face against instabilities due to wireless propagation conditions and/or node mobility.
Our proposal exploits again the hierarchical nature of ATR to address the challenges related to the design of both the two services provided by a DHT system, namely:

- association of information to peers;
- query forwarding to responsible peers.

More specifically, let us define as information the network address ADD_d of a node d, which is identified by a key, namely its identifier ID_d. To associate the network address with a peer, ATR resorts to a globally known hash function $h(\cdot)$, which accepts as argument the key ID_d and returns the peer location $ADD_p = h(ID_d)$, i.e. the network address of the node responsible for storing the mapping $< ID_d, ADD_d >$. The operation of network address updating resorts to Dynamic Address UPdate (DAUP) packets, periodically sent by each node to its peer location. When a node has to send a data packet, it requires (with a Dynamic Address ReQuest (DARQ) packet) the network address of the destination to the peer location associated with destination, which replies to the node with a Dynamic Address RePly (DARP) packet as illustrated in Fig. 2.4. Only after the reception of the DARP packet, the source can forward the data packet towards the destination according to the procedure illustrated in Section 2.3.4.
Since network addresses are assigned to nodes according to the network topology, there is no assurance that the peer location (which is a network address) computed with the hash function is valid, i.e. it has been assigned to a node. To overcome such a drawback, we propose a distributed mechanism (referred

Algorithm 6 peerSelecting()

```
i = 0
while peerLocation is invalid AND i <= l do
   peerLocation.reset(i) //set the i-th bit to zero
   i += 1
end while
```

in the following as *indirect referencing*), characterized by low communication overhead and absence of node coordination. More specifically, to assess the validity of the peer location, each forwarder of a DAUP or DARQ packet[4] checks if the section of its routing table related to the sibling to which the destination belongs to is empty. If so, an invalid peer location has been recognized[5]. As example let us suppose that the node 010 has to forward a DAUP to the invalid peer location 110, which belongs to the level-3 sibling $1xx$. If the forwarder stores at least one route in its 3-th route section, the packet can be forwarded along that route. Differently, if the peer location is invalid, the

[4]The DARP packets do not resort to indirect referencing, since they are certainly sent toward a valid network address: the DARQ source.

[5]We have deliberately neglected the transient effects on the routing table due to mobility or channel propagation instability to simplify the presentation.

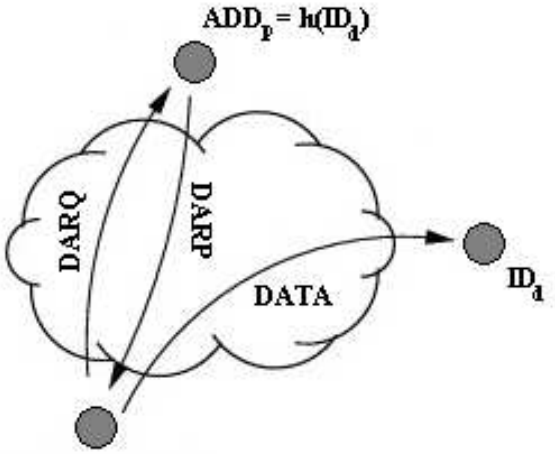


Figure 2.4: Address Discovery Process

forwarder singles out a new peer location according to Algorithm 6, that is it resets the network address one bit at time, starting from the less significant.
The indirect referencing exhibits two characteristics particularly feasible for a network-layer DHT:

i. the peer validation resorts only on the topological information stored in the routing table, without the need of explicit node coordination;

ii. the peer selection allows us to univocally redirect the DAUP and the DARQ packets to a valid peer by exploiting physical neighborly.

As mentioned before, ATR never resorts to data replication, but it makes extensively use of caching techniques to reduce the overhead due to the address discovery process, by storing at each forwarder all the available mapping <node identifier, network address> from unicast packets (data, DAUP, DARQ and DARP). Moreover, when a node changes its routing address, it locally broadcasts a Dynamic Address BRoadcasted (DABR) packet which contains all the mapping stored by the node. This mechanism allows us to face against node mobility, since the neighbor nodes can go on providing the mapping in the transient time. Clearly, a purge mechanism allows nodes to delete expired information.

2.4 Performance analysis

In this section, we present a numerical performance analysis of the proposed protocol by resorting to *ns-2* (version 2.29) network simulator [72].
At this end, for the sake of performance comparison, we consider three commonly adopted routing protocols, namely Ad Hoc On-Demand Distance Vector (AODV) [42], Dynamic Source Routing (DSR) [44] and Destination-Sequenced Distance Vector (DSDV) [36]. Since the Dynamic Address RouTing (DART) protocol [55] copes only with the address allocation and routing aspects, neglecting the address discovery process, it can not be considered in this performance comparison. However, in [59, 73] it has been shown that, with reference to both the mentioned aspects, ATR outperforms DART.
We ran a large set of experiments to explore the impact of several workloads and environmental parameters on the protocol performances by adopting the following three metrics:

i. packet delivery ratio (PDR): the ratio between the number of data packets successfully received and those generated, both by the application layer in the case of UDP transport protocol;

ii. hop count: the number of hops a data packet took to reach its destination; this metric accounts only for the data packets successfully received;

iii. routing overhead: the ratio between the number of generated data packets and the total number of generated routing packets;

Each experiment ran ten times, and for each metric we estimated both its average value and the standard deviation.

2.4.1 Channel model

Usually, routing performance analysis for ad-hoc networks adopts as radio propagation model the *Two-Ray Ground* one [55, 56, 61, 62], based on the following assumptions:

i. the radio's transmission area is circular and all the radios have equal range;

ii. communications are bidirectional (if a node receive a packet from a neighbor, then that neighbor will receive its packets too);

iii. the channel model is time-invariant (if a node can send a packet to a neighbor once, it will be possible until the topology does not change).

To remove these optimistic assumptions [74], we consider a propagation model, the *Shadowing* one, which accounts for the long-term fading effects by means of a zero-mean Gaussian variable $N(0, \sigma)$. Therefore, the received mean power $P_{dB}(d)$ at distance d is:

$$P_{dB}(d) = P_{dB}(d_0) - \log \beta(d/d_0) + N(0, \sigma) \tag{2.5}$$

where $P_{dB}(d_0)$ is the received mean power at the first meter, β is the path-loss exponent and σ is the shadow deviation, both empirically determined for a certain environment.
Moreover, unlike most routing performance analysis [67, 75], we take into account the effects of both the additive thermal noise and the interferences, by assessing the signal-to-interference-plus-noise (SINR) ratio at the receiver side:

$$SINR = 10log\frac{P}{\sigma_n^2 + \sum_i P_i} \tag{2.6}$$

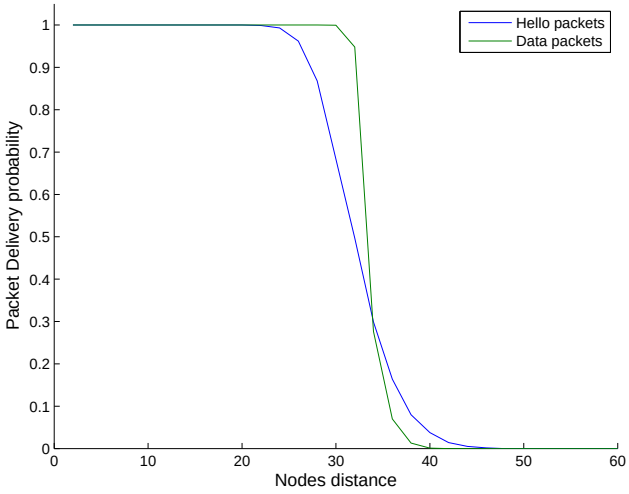


Figure 2.5: Channel characterization

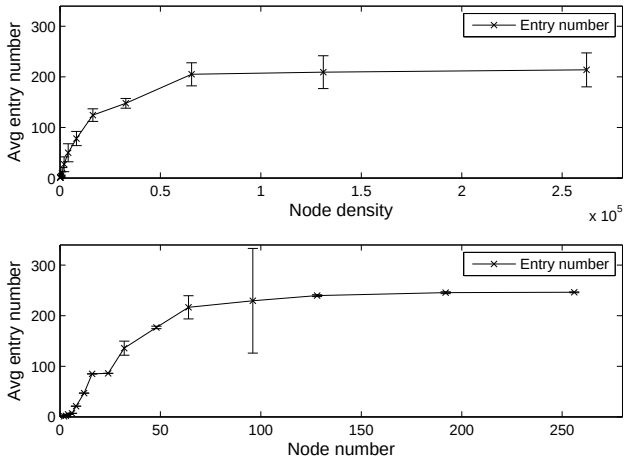


Figure 2.6: ATR memory requirements

where P is the received useful mean power, σ_n^2 is the additive noise mean power and finally, P_i is the received interference mean power. The signal to interference plus noise ratio (SINR) ratio is thus used to state if the received packet has been correctly received according to [76].
We set the path-loss exponent to 3.8, the shadow deviation to 2.0 and the mean noise power to -82dBm to simulate an IEEE 802.11b Orinoco network interface [77] with long preamble, CCK11 modulation and two-handshake mechanism, resulting in a transmission range of roughly 35 meters as shown by Fig. 2.5.

2.4.2 Experimental setup

Static network topologies have been generated by placing the nodes uniformly in the scenario area, while mobile ones resort to *Random Way-point* [20] as mobility model. The mobility parameters have been set to simulate pedestrian mobility, since ATR is not suitable for networks with higher levels of mobility due to its proactive characteristic. More specifically, the speed and the pause values are uniformly taken in the [0.5m/s; 1.5m/s] and in the [1s; 100s] ranges respectively, according to [78] to avoid the speed decay problem.
The node density has been set to 4096 $nodes/Km^2$. This value corresponds to a mean node connectivity degree of 12, which is a reasonable value to avoid the presence of network partitions [79], and the size of the scenario area was chosen according to this connectivity degree.
The duration of each run is 2060 seconds, longer then de facto standard value (900 seconds) to increase the accuracy of the measurements. All the measurements are taken during the interval [1000s; 2000s], since the initial 1000 seconds are used to ensure that the routing protocols reach a steady state.
The well-known *random traffic model* [20] is adopted as data pattern: every node singles out randomly a destination according to a uniform distribution among the remaining nodes. The workload is modeled as a constant bit rate (CBR) flow over UDP protocol with 1000 byte as packet size , and each flow starts at 1000 seconds and ends at 2000 seconds.
To effectively assess the scalability property of the analyzed protocols, instead of resorting to the capacity scaling bounds [80] for static scenarios (namely $O(\sqrt{n})$), we set the data throughput λ generated by each source to:

$$\lambda = \frac{W}{n\sqrt{n}} \tag{2.7}$$

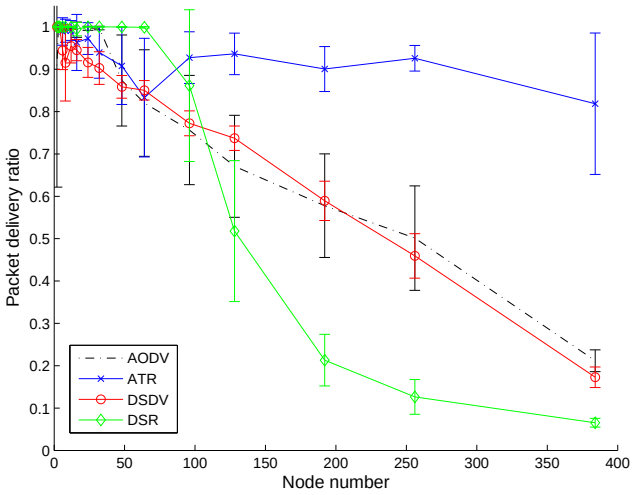


Figure 2.7: Packet delivery ratio vs node number

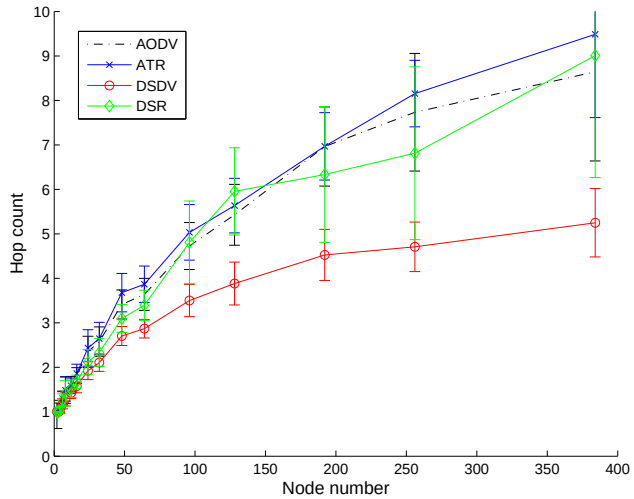


Figure 2.8: Hop count vs node number

where W is the link data throughput for a 802.11b channel with CCK11 modulation (about 5.4Mb/s) and n is the number of nodes in the network.
Such a choice is justified by the need to take into account the throughput reduction effects due to the routing service, since the scaling factor n accounts for the routing overhead generated by the periodic signaling of proactive protocols. It is worthwhile to underline that the adopted data load is in any case heavier than those usually adopted in routing performance analysis [20, 61, 56, 81].
We do not present the results regarding TCP flows since it offers a conforming load to the network, meaning that it changes the packet rate according to its perception of the network congestion. As a result, since any protocol is characterized by different time at which each data packet is originated by its sender and different position of the source node, a fair comparison among them is not possible.
However, our results have shown that the aggregate data throughput delivered on TCP flows by both ATR and AODV is unaffected by the number of nodes, whereas both DSDV and DSR performances decreases as the number of nodes grows. Moreover, TCP favors shorter connections, that is, it exhibits flow elasticity, as confirmed by the results in terms of hop number (all the protocols deliver packet on routes shorter than 3 hops in a network with 384 nodes).

2.4.3 Memory requirements

The first set of experiments aims to evaluate the memory requirements of ATR in terms of average number of routing table entries, which represents the overall cost due to the multi-path approach, since no communication overhead is introduced by ATR (Section 2.3.3) with respect to shortest-path protocols like DART.
We have run twenty trials for each experiment to measure the average number of routing table entries of all participating nodes and its standard deviation in two different scenarios (Fig. 2.6): in the former the node density increases whereas the node number is set to 64, and in the latter the node number increases while the node density is set to 4096 $nodes/Km^2$.
The results show the presence of a saturation effect for both the scenarios, which assures that the overhead is bounded in terms of memory space. Such a behavior is due to the choice of adopting a threshold based on the link quality in order to accept the routing updates from neighbors (Section 2.3.2).

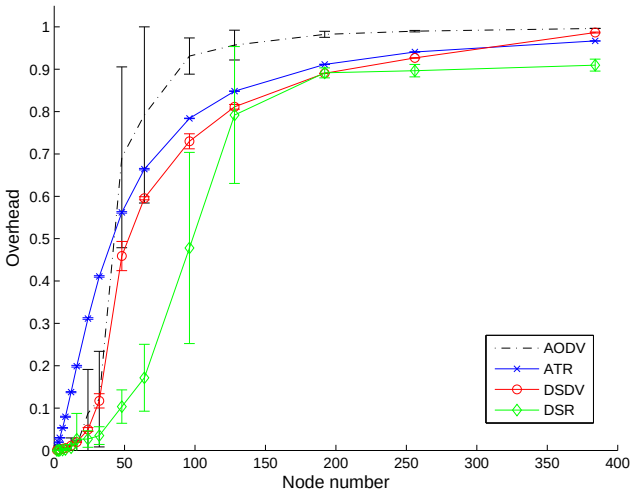


Figure 2.9: Routing overhead vs node number

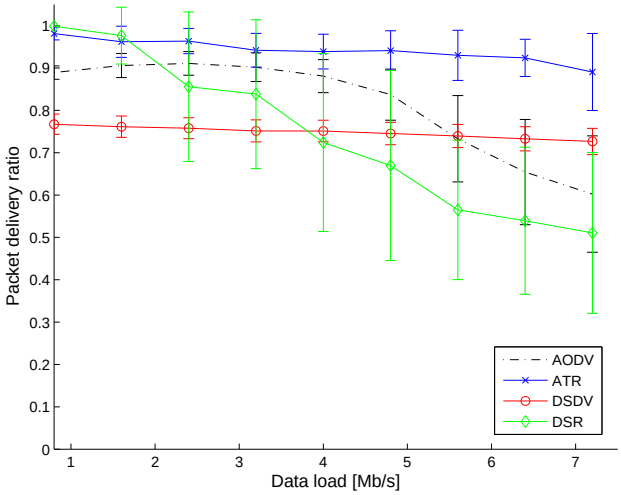


Figure 2.10: Packet delivery ratio vs data load

2.4.4 Performance comparison

Since we are primarily concerned with scalable networks, the second set of experiments aims to compare the protocol performances for a static scenario as the number of nodes increases (Fig. 2.7-2.9).
More specifically, as regards the packet delivery ratio (Fig. 2.7), ATR remains largely unaffected as the number of nodes increases. On the other hand, DSDV and AODV performances decrease rough linearly with the number of nodes. Finally, DSR outperforms all the remaining protocols only for small networks whereas, as the number of nodes increases, its performances become the worst and, with reference to largest networks, nearly an order of magnitude separates them from those of ATR. Such a behavior lies in the source routing nature of DSR since, as the network size grows, the complete ordered list of nodes through which the packet must pass stored in the packet's header becomes out-of-date.
Fig. 2.8 shows the hop count for the delivery ratios presented by Fig. 2.7. ATR has been designed to prefer reliable paths, despite of the hop number. Moreover, its hierarchical nature is a potential source of path length inefficiency. However, its performances are comparable with those of AODV and DSR, which experience a path stretch, defined as the ratio between the discovered path length and the shortest path length, of roughly two. In fact, by bounding the average shortest path length $\overline{h}$ measured in hop number as [73] (further details in Sec. 3.3.2):

$$\overline{h} = \left\lceil \frac{2\sqrt{\frac{n}{\delta}}}{3\sqrt{\pi}r} \right\rceil \tag{2.8}$$

where n is the number of nodes, δ is the node density, r is the transmissions range and $\lceil\rceil$ rounds to the higher integer, we have that $\overline{h} = 5$ for a network with 384 nodes. Therefore, DSDV is able to discovery routes very close to the shortest ones. Moreover, if we account for both the delivery ratio and the hop count performances, DSDV performs better than AODV since, by delivering the same number of packets on shorter routes, it uses more efficiently the network resources.
Finally, the results reported in Fig. 2.9 show that DSR outperforms all the other protocols in terms of routing overhead due to its aggressive route caching policy. Again, DSDV and AODV perform similarly in small networks but, when the number of nodes grows, AODV performs worst due to its reactive nature. In small networks, ATR exhibits the highest overhead, since its routing update packets have fixed size, regardless of the node number. However, when the number of nodes grows, its behavior becomes comparable with those of the

other proactive protocol, i.e. the DSDV.
Numerical results not here reported show that, if we account for the ratio between the total number of bytes sent at the routing layer over the total number of data bytes received, ATR outperforms all the other protocols thanks to its hierarchical approach. In fact, in largest networks, ATR ratio is about 15, AODV one is 66, DSDV one is 102 and DSR one is 58.

The third set of experiments aims to state a performance comparison for a static scenario with 128 nodes as the data load increases, namely as the value of the link data throughput W in Eq. 2.7 grows (Fig. 2.10-2.12).
The results in terms of packet delivery ratio (Fig. 2.10) show that the proactive protocols are able to scale well in terms of data load, whereas both DSR and AODV performances are affected by this parameter. Among all the protocols, ATR outperforms for nearly each data load. Moreover numerical results, not here reported, show that ATR outperforms all the other protocols in terms of delivery ratios for rough every data load when the number of nodes exceeds 64, whereas in small networks DSR reaches the best performances, confirming so the previous results (Fig. 2.7).
Regarding the hop count metric results (Fig. 2.11), unlike reactive protocols, the path lengths of proactive protocols are unaffected by the data load. DSDV routes have length closer to shortest ones ($\overline{h} = 3$ according to Eq. 2.8), confirming so the previous considerations (Fig. 2.8).
Finally, Fig. 2.12 illustrates the performances in terms of routing overhead: the proactive routing traffic does not depend on the data load, since the routing overhead decreases linearly with the data load, whereas the reactive routing traffic increases linearly with the data load. This behavior agrees with the one exhibited by the results concerning the delivery ratios.
The fourth set of experiments (Fig. 2.13-2.15) aims to assess the performances for mobile scenario with 64 nodes as the number of mobile node increases, according to the mobility model illustrated in Section 2.4.2.
ATR delivery ratios are slightly affected by the node mobility (Fig. 2.13), since its routing process exploits the topological meaning of the network addresses. However, the augmented structure build upon the address space by means of the multi-path approach allows ATR to perform satisfactorily in the case of moderate mobility. In this scenario, both the DSDV and the DSR delivery ratios are nearly independent of the node mobility. However, this behavior is exhibited only in small networks and both perform poorly for largest networks. Like ATR, also AODV performances depend on the node mobility.
Regarding the hop count metric performances (Fig. 2.14), DSDV and AODV

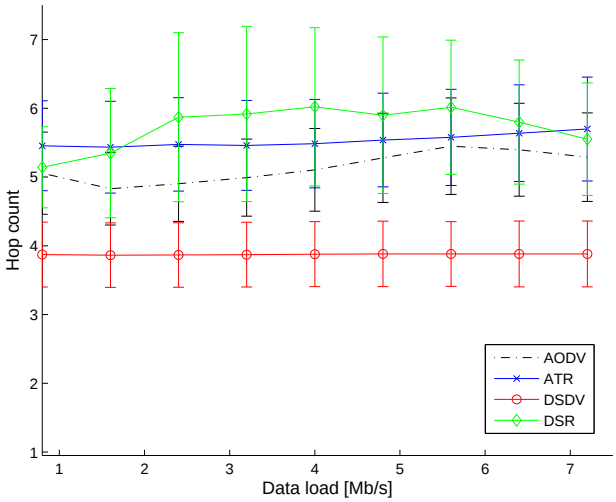


Figure 2.11: Hop count vs data load

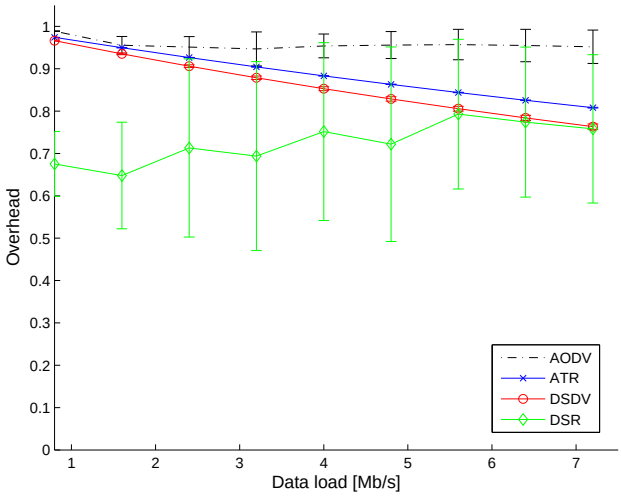


Figure 2.12: Routing overhead vs data load

take advantage by the route diversity introduced by node mobility. Differently, both ATR and DSR performances are affected by this parameter: the former since it uses source routing, and the latter because it resorts to hierarchical

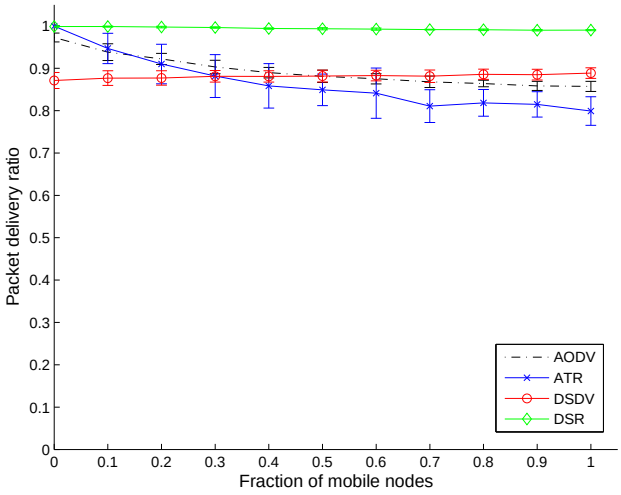


Figure 2.13: Packet delivery ratio vs fraction of mobile nodes

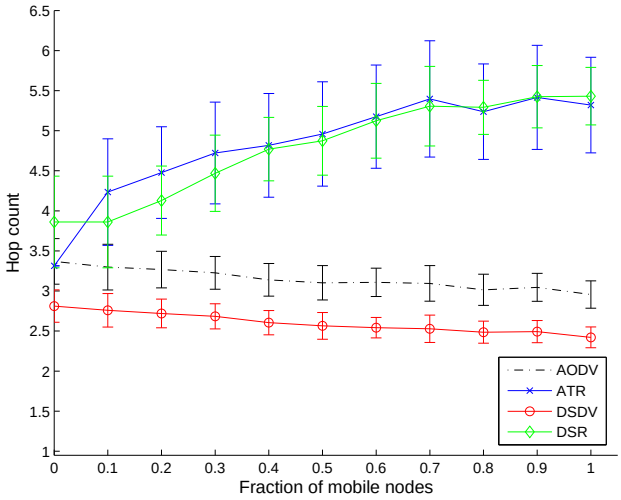


Figure 2.14: Hop count vs fraction of mobile nodes

routing.
Finally, the results regarding the routing overhead (Fig. 2.15) shows that, unlike reactive protocols, ATR and DSDV exhibit constant mobility-indipendent

overhead. The last set of experiments aims to evaluate the performances for a static scenario with 64 nodes as the hostility of the channel, namely the shadow deviation, increases (Fig. 2.16-2.18).
The shadow deviation affects in different ways the delivery ratios of all the protocols. DSR performance exhibits a non-linear behavior: the delivery ratio is nearly one in the case of line-of-sight communications ($sigma \leq 4$) but, as the shadow deviation increases, it becomes unable to deliver packets. Both ATR and AODV delivery ratios have an approximately linear relationship with the shadow deviation, but ATR performance remains largely although $\sigma = 6$, outperforming so the other protocols for a large set of propagation conditions. DSDV performance initially decreases as the shadow deviation grows, but it outperforms the other protocols in absence of line-of-sight communications, namely for the highest values of σ. The previous considerations are confirmed by the hop count metric (Fig. 2.17). DSDV is the only one whose hop count performances are unaffected by the channel hostility, whereas AODV, ATR and DSR path lengths increase rough linearly with the shadow deviation.
Finally, the considerations regarding the overhead metric as the hostility increases (Fig. 2.18) are the same of those made for node mobility (Fig. 2.15): the proactive overhead, unlike the reactive one, is independent of shadow fading.

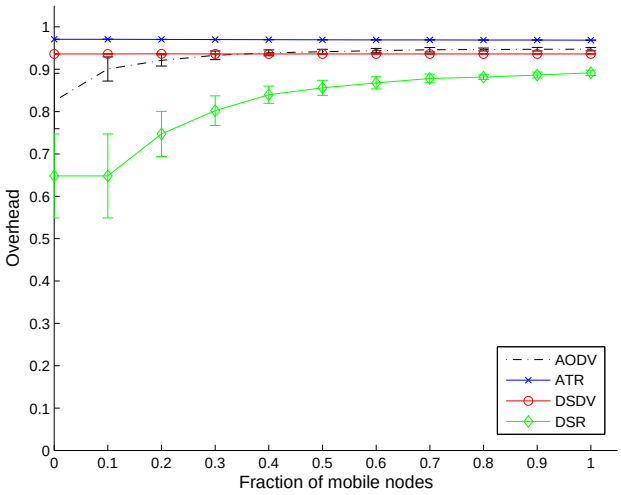


Figure 2.15: Routing overhead vs fraction of mobile nodes

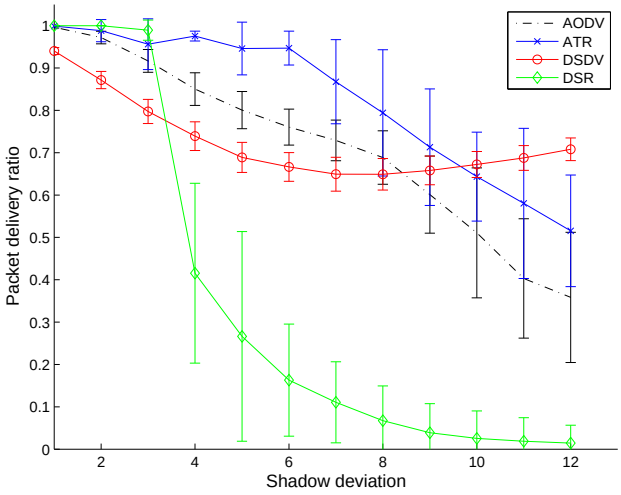


Figure 2.16: Packet delivery ratio vs shadow deviation

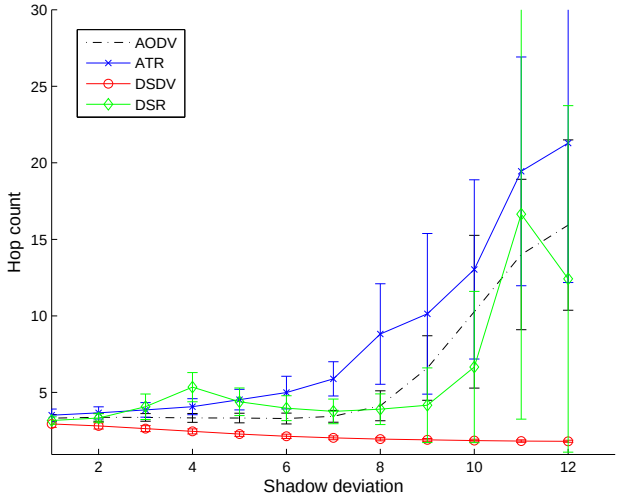


Figure 2.17: Hop count vs shadow deviation

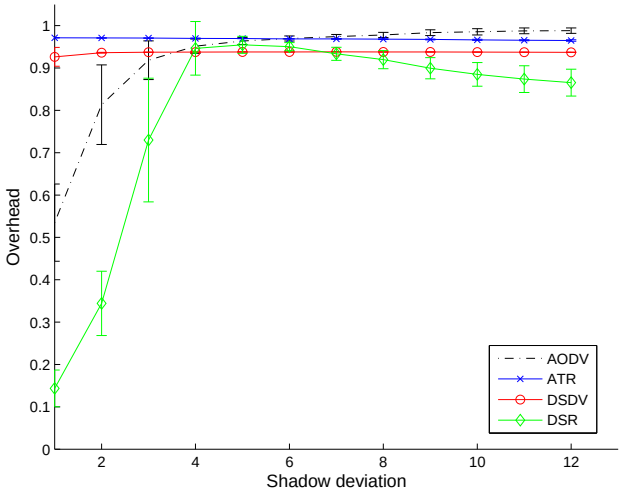


Figure 2.18: Routing overhead vs shadow deviation

Chapter 3

Reliability analysis

Unlike traditional routing procedures that, at the best, single out a unique route, multi-path routing protocols discover proactively several alternative routes. It has been recognized that multi-path routing can be more efficient than traditional one mainly for mobile ad hoc networks, where route failure events are frequent. Most of the studies in the area of multi-path routing focus on heuristic methods, and the performances of these strategies are commonly evaluated by numerical simulations. The need of a theoretical analysis motivates us to resort to the terminal-pair routing reliability as performance metric. This metric allows one to assess the performance improvement gained by the availability of route diversity. More specifically, resorting to graph theory, we propose an analytical framework to evaluate the tolerance of multi-path route discovery processes against route failures for mobile ad hoc networks. Moreover, we derive a useful bound to easily estimate the performance improvements achieved by multi-path routing with respect to any traditional routing protocol. By means of numerical simulation, we have assess the effectiveness of the proposed framework.

3.1 Introduction

In the last ten years, mobile ad hoc network (MANET) technologies have tremendously grown. A MANET is an autonomous system of mobile nodes connected by wireless links, without any static infrastructure such as access points. Such kind of networks was introduced manly for military and emergency applications, but recently, thanks to the mesh paradigm, it can guarantee ubiquitous communication services, and it is mandatory when no

cellular or other fixed network infrastructures are available.
To reach a destination node located out of the coverage range of the sender node, a multi-hop communication strategy must be exploited; in such a case, each node has to cooperate with the other ones and acts as relay for packet transmission. In this scenario, the instability of the topology (link and node failures) due to node mobility and/or changes in wireless propagation conditions can frequently give rise to disconnected routes.
For such reasons, the design of an effective routing protocol for ad hoc scenarios is a challenging problem, and much research activity had been carried on in the last years, producing a plethora of different approaches and solutions. The proposals in [33] focus on discovering the shortest available route, according to some metrics, and all the traffic is routed over that path. This approach exhibits low tolerance against route failure events, since in such case it is necessary to stop the data transmissions until a new route will be discovered [82].
An interesting approach to gain tolerance against unreliable wireless links and node mobility is based on *multi-path routing*, in which multiple routes are proactively found. In order to effectively exploit the advantages of multi-path approaches, it is necessary to assess the performance gain reached by these strategies and, moreover, to evaluate the trade off between advantages and costs in adopting more complex multi-path solutions.
Different studies and proposals on multi-path routing have focused on heuristic methods to establish how many routes are needed and how to select them. The on-demand multi-path routing protocol in [83], which is an extension of the well-known Dynamic Source Routing (DSR) protocol [44], takes advantages of maintaining alternative disjoint routes to be utilized when the primary one fails. However, the performance benefits are evaluated only in few particular cases, regardless the tolerance against route failures. The Ad hoc On-Demand Distance Vector Routing with Backup Routes (AODV-BR) protocol [84], which is an extension of Ad Hoc On-Demand Distance Vector (AODV) one [42], is analyzed by a numerical simulation analysis, which adopts the packet delivery ratio as performance metric. The same approach for performance evaluation is adopted in several works on multi-path routing, as in [85, 86, 87, 88, 89].
Some works have addressed the problem to analytically assess the multi-path benefits by resorting to graph theory, for both wireless sensor networks and MANETs. More specifically, in [90, 91] the study is focused on a particular routing protocol, whereas in [92, 93] the tolerance against route failures is

evaluated with reference to the physical layer, namely in terms of network connectivity. In [94, 95] the evaluation is performed for wireless sensor networks, assuming a hierarchical structure and the presence of a sink node. Finally, in [96, 97, 98] an analytical evaluation of multi-path routing is carried out by resorting to diversity coding.
This chapter, based on the work in [63, 73] proposes an analytical framework to evaluate the tolerance of multi-path route discovery processes against route failures, rather than to single out new multi-path routing discover processes. More specifically, with reference to MANET paradigm, we propose to resort to a theoretical approach based on graph theory. We first introduce an analytical framework based on the terminal-pair routing reliability (TPRR) as measure of the tolerance of routing protocols against route failures. Unlike the packet delivery ratio, such a metric allows one to evaluate the robustness against the link failures, as a function of the number of the discovered routes as well as their reliability. In order to derive the analytical expression of the TPRR, we resort to the concept of *overlay graph*, namely the logical structure built by the route discovery process (RDP) of a routing protocol upon the physical network. In this way, the incomplete knowledge about the network topology that each node possesses is taken into account. Then, it is introduced an upper bound on the TPRR of any shortest-path RDP. This allows one to easily compare the performances improvement gained by a multi-path RDP with respect to whatever shortest-path one. An algorithm for exact evaluation of routing reliability, both in numerical and symbolic form, is also provided.

3.2 Network model and assumptions

In the following we introduce the network representation by resorting to the graph theory and present the main assumptions utilized in our analysis.
The nodes in the network are assumed to be reliable, while the links are failure-prone [99]. This assumption is reasonable for both static and mobile networks. In fact, in a static network, as in a sensor one, the failure of a link is due to the instability of wireless propagation conditions and to the capacity constraints, whereas in a mobile network, as in a MANET, the failure of a link is also due to the node mobility. In the following, we assume that the node mobility does not affect the reliability performance. Clearly, this assumption is realistic only when the node mobility is relatively low, since in such a case the packet delivery times are commonly smaller than those associated with topology changes

[100]. The results of numerical simulations reported in Sec. 3.4.3 confirm the validity of such assumption for scenarios with moderate node mobility.
We model the network with a probabilistic direct graph:

$$G = (V, E, P) \tag{3.1}$$

in which a vertex $v_i \in V$ denotes a node belonging to the network and an edge $e_{ij} \in E$ represents a communication link from node v_i to node v_j. Each link is characterized by a failure probability p_{ij} (the *i*-th element of the link-failure probability matrix *P*) , which represents the probability that, at the transmission attempt time, the link is down. The edge failure events are assumed statistically independent of each other.
Given a probabilistic graph G , we define an overlay graph as:

$$G_o = (V, E_o, P_o) \tag{3.2}$$

where $E_o \subseteq E$ and P_o is the link-failure probability matrix associated with E_o.
Since a node s discovers (by means of the RDP) only a subset $E_{s,t} \subseteq E$ of the available links to reach a destination t, we can define the overlay graph built by the RDP upon the physical network topology as:

$$G_{s,t} = (V, E_{s,t}, P_{s,t}) \tag{3.3}$$

In the following, we refer to the graph defined in (3.1) as the *physical graph*, which is a representation of the physical topology, while we refer to the graph defined in (3.3) as the *overlay graph*, to which we resort to evaluate the tolerance of a routing protocol against path failures.

As example, in Fig. 3.1 both the physical graph of a network and a related overlay graph for the flow (2,8) are depicted. Clearly, for each routing protocol and for each flow, the RDP defines a different overlay graph, which accounts for the features of the particular RDP as well as the network topology. Then, the overlay graph allows us to evaluate the effectiveness of the RDP adopted by any table based routing protocol. In fact, it allows one to assess the number of multiple paths for each flow, and moreover their disjointness degree (i.e. the number of disjoint links among a set of routes), enabling so to analytically evaluate the tolerance against path failures.

3.3 Performance analysis framework

In this section, we present the proposed analytical framework for assessing the tolerance of RDP schemes to link failures, as well as the bound on the

reliability for shortest-path RDP strategies.

3.3.1 Preliminaries

With reference to a unicast routing scenario, let us adopt as RDP performance measure the terminal-pair routing reliability, namely the probability that at least one route from the node s to the node t exists.
Considering the flow (s,t) from the node s to the node t and denoting with $\Re_{s,t}$ the set of routes found by the RDP, we define the TPRR as:

$$R_{s,t}(G_{s,t}) = P(\Re_{s,t} \neq \emptyset) \tag{3.4}$$

where $G_{s,t}$ is the overlay graph built by the RDP for the flow (s,t).
The TPRR (3.4) can be re-written as:

$$R_{s,t}(G_{s,t}) = 1 - \sum_{i=c}^{m} C_{s,t}(i) p^i (1-p)^{m-i} \tag{3.5}$$

where $m = |E_{s,t}|$ is the cardinality of the edge set $E_{s,t}$, $p \equiv p_{i,j}$ is the link-failure probability (assumed for simplicity the same for each pair of nodes), c is the minimum edge cut set[1] dimension of the overlay graph between s and

[1] An edge cut set for the flow (s,t) is a set of edges whose removal disconnects s and t.

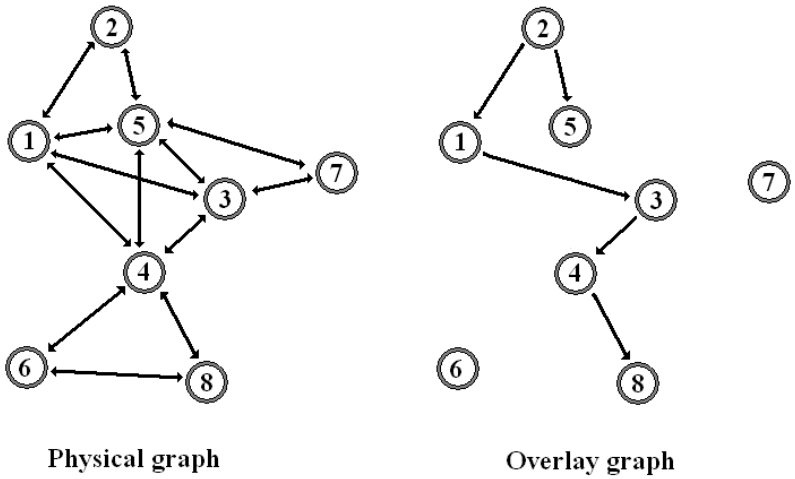


Figure 3.1: Physical and overlay graphs

t, and $C_{s,t}(i)$ is the number of cut sets between s and t in the overlay graph composed exactly by i edges. Then, the mean TPRR is:

$$R = \frac{\sum_{s \in V} \sum_{t \in V, t \neq s} z_{s,t} R_{s,t}}{n(n-1)} \tag{3.6}$$

where $n = |V|$, and $z_{s,t}$ is the probability that a data flow occurs between nodes s and t.

Accounting for the results in [101], we derive the symbolic expression of TPRR as a function of the link-failure probability p. More specifically, the Algorithm 7 allows to exactly compute the TPRR (3.5) using the overlay graph. The algorithm is invoked by initializing G to the overlay graph $G_{s,t}$, the set SS to empty, and n to s. Then, the node n is included in the set SS as well as the redundant nodes, in order to ensure that the set of all emitting edges from a particular SS is a minimal cut set[2] If the singled out set SS is already in the hash table HASH, nothing needs to be done. Differently, SS is a minimal cut set and it has to be added to the hash table. Then, the procedure computes the unreliability (the probability that all the links fail) for the cut set and recursively calls itself for each node adjacent to the cut set SS.

3.3.2 Polynomial bound on shortest-path reliability

In this sub-section, the performance gain achieved by a multi-path RDP with respect to any shortest-path one is estimated by resorting to an upper bound which holds for any shortest-path scheme.

The RDP of a shortest-path protocol, at best, singles out a unique route $P_{s,t}$ for the flow (s,t). Let us define with $h^o(s,t)$ the overlay distance between (s,t), i.e. the length of $P_{s,t}$ measured in number of hops on the overlay graph. Denoting with $h(s,t)$ the physical distance between (s,t), namely the hop distance measured on the physical graph, we have:

$$h(s,t) \leq h^o(s,t); \forall s,t \in V \tag{3.7}$$

since the link set $E_{s,t}$ of the overlay graph is a sub-set of the link set E of the physical graph and so the overlay distance $h^o(s,t)$ can not be less then $h(s,t)$. So, the TPRR for a shortest path routing protocol can be upper bounded as:

$$R_{s,t}(G_{s,t}) = (1-p)^{h^o(s,t)} \leq (1-p)^{h(s,t)} \tag{3.8}$$

[2]A node is redundant if it is adjacent to SS and has no way to reach t without exploiting any node in SS.

Algorithm 7 Recursive(G,HASH,SS,s,t,notRel,symbNotRel)

```
{Reliability = 1 - Recursive(É) output}
{G is the adjacency matrix related to the overlay graph}
{HASH is a collection of minimal cut set, initialized to empty}
{SS is the under analysis minimal cut set, initialized to empty}
{n is initialized to s}
neighborListPurge()
routingTableClear()
if n = t then
   return
end if
merge(G, SS, n) {Merging node n in SS}
absorb(G, SS, t) {Absorbing redundant nodes in SS}
if HASH.isPresent(SS) then
   return
end if
HASH.insert(SS)
find a cutset C of SS
symbTempNotRel = Ò(1-p) Ô + C.size.toString
tempNotRel = 1.0
symbTempNotRel = symbTempNotRel + " + p * (Ò + symbTempNotRel;
for each edge in C do
   tempNotRel = pFailed * tempNotRel
end for
for each node adjacent to SS do
   Recursive(G,HASH,SS,n,t,tempNotRel,symbNotRel)
   tempNotRel = pSuccess * tempNotRel
end for
symbTempNotRel = symbTempNotRel + Ò)Ó
notRel = notRel + tempNotRel;
```

To estimate the distance $h(s,t)$ which clearly depends on the network topology, we make some reasonable assumptions. More specifically, we assume, according to [102], that the node density δ is uniform (according to the first interference principle) as well as the transmissions range r, and the physical network area A is a circle. Moreover, we assume the traffic pattern random as [102], namely each destination node is chosen with equal probability

($z_{s,t} \equiv z$), and the node s is located at the centre of the network (to neglect the boundary effect). Under these assumptions, the number of nodes in the circle of radius x is:

$$n(x) = \pi x^2 \delta, \qquad 0 \leq x \leq \sqrt{\frac{A}{\pi}} \tag{3.9}$$

The probability that the node s communicates with a node belonging to a circular neighborhood of radius x can be written as:

$$P(X \leq x) = \frac{\pi x^2}{A}, \qquad 0 \leq x \leq \sqrt{\frac{A}{\pi}} \tag{3.10}$$

where X is the random variable representing the path length between (s,t). From (3.10), the probability density function is:

$$f_X(x) = \frac{2\pi x}{A}, \qquad 0 \leq x \leq \sqrt{\frac{A}{\pi}} \tag{3.11}$$

Consequentially, the average path length $\overline{L}$, measured in distance units, is:

$$\overline{L} = E[X] = \int_0^{\sqrt{\frac{A}{\pi}}} x f_X(x) dx = \frac{2\sqrt{A}}{3\sqrt{\pi}} \tag{3.12}$$

and the average physical distance, measured in number of hops, is:

$$\overline{h} = \left\lceil \frac{\overline{L}}{r} \right\rceil = \left\lceil \frac{2\sqrt{A}}{3\sqrt{\pi} r} \right\rceil = \left\lceil \frac{2\sqrt{\frac{n}{\delta}}}{3\sqrt{\pi} r} \right\rceil \tag{3.13}$$

where n is the total number of nodes in the network and $\lceil \rceil$ rounds to the higher integer.
Thus, the upper bound on the TPRR for any shortest path RDP is:

$$R_{s,t}(G_{s,t}) \leq (1-p)^{\left\lceil \frac{2\sqrt{\frac{n}{\delta}}}{3\sqrt{\pi} r} \right\rceil} \tag{3.14}$$

3.4 Reliability analysis

The aim of this section is to show the effectiveness of the proposed analytical framework to assess the tolerance against link failures for any RDP strategy by means of a performance comparison among both shortest-path and multi-path routing protocols. At this end, three shortest-path routing protocols, Optimized

Link State Routing (OLSR) [103], Dynamic Address RouTing (DART) [55] and AODV [42], and two multi-path ones, Augmented Tree-based Routing (ATR) [59] and Ad hoc On-demand Multipath Distance Vector (AOMDV) [104], are considered. More specifically, OLSR and DART are both proactive protocols, and DART, unlike OLSR, is hierarchical, i.e. it groups the nodes belonging to the network in zones, namely siblings, and stores a unique route towards each zone for scalability purposes. AODV is a reactive routing protocol, while AOMDV generalizes AODV to exploit multiple paths with disjoint links between the source and the destination. Analogously, ATR generalizes DART, looking for multiple routes towards the same zone.

3.4.1 Overlay graph generation

The overlay graphs needed to compute the mean TPRR have been generated by simulation using Network Simulator 2 (ns-2) [72]. Fig.3.2 shows the generating process of the overlay graphs.
For each network topology, we run a ns-2 based simulation in order to populate the routing table of each node. The path information embedded in the routing table is then used to generate the overlay graph for each flow (s,t). The choice of using ns-2 to generate the routing tables has the following two advantages:

- the overlay graphs are straight generated by the RDP utilized by the specific routing protocol;

- the analysis can be easily extended to different routing protocols with a light effort, simply providing to the protocol code a function which prints out the node routing table.

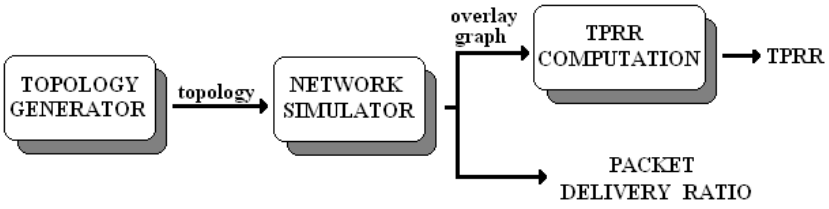


Figure 3.2: Overlay graph generating process

3.4.2 Overlay graph generation

The main characteristics of the setup for the reliability assessment are briefly summarized in the following. We adopt for both the physical and the link layer the parameter values usually utilized in ns-2 to simulate an IEEE 802.11a Lucent network interface with Two-Ray Ground as propagation model. The duration of simulation is set to 500 seconds to allow the routing tables to become consistent with respect to the network topology. The sizes of the scenario areas are chosen to keep the node density equal to 64 nodes/Km2, which avoids the presence of isolated nodes [79] by assuring a mean node connectivity degree of 12. The network topologies are randomly generated by independently and uniformly distributing the nodes in the scenario area.
We have performed measures for 100 trials for each network size. More specifically, we have reported the TPRR for the shortest-path RDPs (OLSR, DART and AODV), the shortest-path upper bound on TPRR, and the TPRR for the multi-path RDPs (AOMDV and ATR). Each figure shows the average and the variance of TPRR for each protocol as function of the link-failure probability. Fig.3.3 refers to a 4 nodes full-mesh network. In this case, the average TPRR reached by the shortest-path protocols agrees with the shortest-path upper bound. This means that, for very small networks, their RDP is often able to find the optimal route (one-hop route) between each pair of nodes. We note that DART RDP reaches lower values of TPRR with respect to other shortest-path RDPs, although the differences cannot be recognized in the figure. Regarding multi-path RDPs, both AOMDV and ATR outperform the shortest-path protocols also in such a small network. Fig.3.4 refers to a network with 8 nodes. In this case, the shortest-path protocols experience lower values of TPRR with respect to the shortest-path upper bound. Since the node connectivity degree is 12, every pair of nodes is physically linked and so the optimal route is one-hop long, in accordance with the upper bound depicted in Fig.3.4. However, the shortest-path RDPs reach lower values, i.e. they discover longer routes than the optimal ones. DART RDP performs worst due to its hierarchical nature, and the largest difference is about 0.08 in correspondence of the link-failure probability $p = 0.5$.
Regarding to AOMDV, for low link-failure probability, it outperforms any shortest-path protocol thanks to its multi-path characteristic, whereas, when the link-failure probability increases, such behavior does not apply. This behavior is reasonable, since AOMDV adopts the same route discovery of AODV, so that neither it can find the optimal routes.
Since ATR is a proactive routing protocol, it persistently broadcasts routing

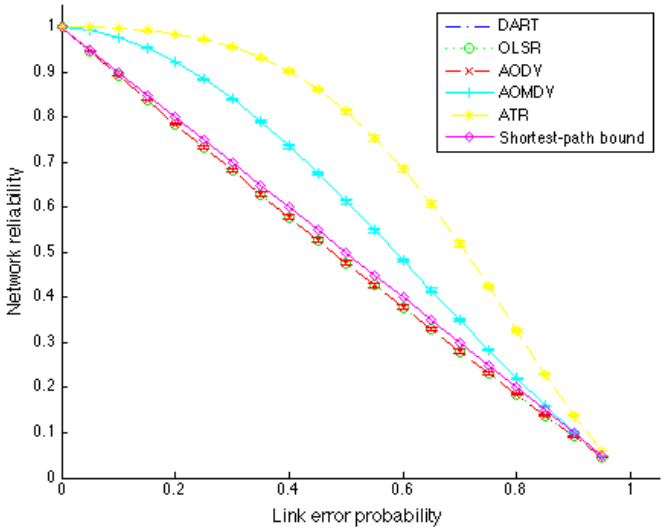


Figure 3.3: 4 nodes full mesh network

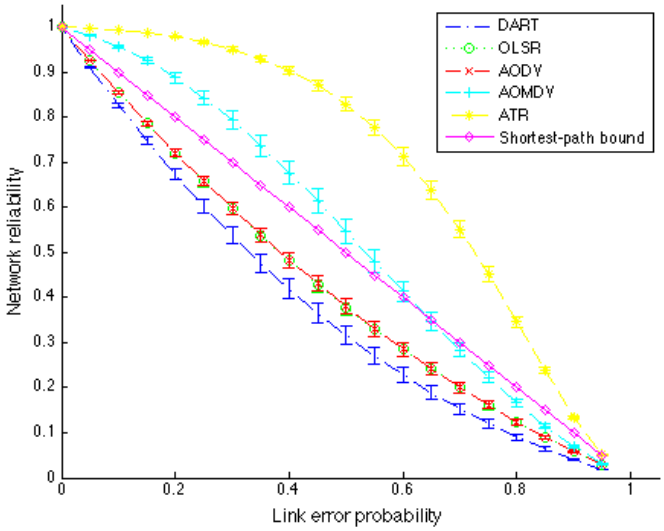


Figure 3.4: 8 nodes network

packets in order to discovery redundant routes. Therefore, it is able to find more paths than AOMDV. Clearly, the ATR routing overhead is higher than AOMDV one. The behavior of shortest-path protocols depicted in Fig.3.4 can be interpreted by resorting to Fig.3.5, which shows an example of the routes discovered by different RDPs. The first row shows the overlay graphs built by the shortest-path RDPs towards the node Ô2Õ from three source nodes (Ô1Õ, Ô3Õ and Ô4Õ). In this case, any RDP is not able to find out the optimal route for every flow and DART, due to its hierarchical nature, finds out less optimal routes than other ones. The second row presents the routes towards the node 4 singled out by the multi-path RDPs, which are able to discover redundant paths for the same flow.
Fig.3.6 and Fig.3.7 show the results for a 16 and a 32 nodes network respectively. All the previous considerations concerning Fig.3.5 are sill valid. ATR is able to discover more path than AOMDV, since AOMDV RDP outperforms any shortest-path routing protocol only for low link-failure probability.
On the whole, the TPRR analysis evidences that the multi-path approach, apart from the particular RDP scheme, is suitable for scenarios with *nearly reliable* links, whereas for *nearly unreliable* links the multi-path gain is negligible.
Finally, we show that the TPRR can be exploited to assess the trade-off that a routing protocol experiences between benefits due to multiple available routes and the overhead needed to discover them. In the following, we resort to TPRR to evaluate this trade-off with respect to the ATR RDP scheme.
The original ATR protocol looks for every available route towards the same

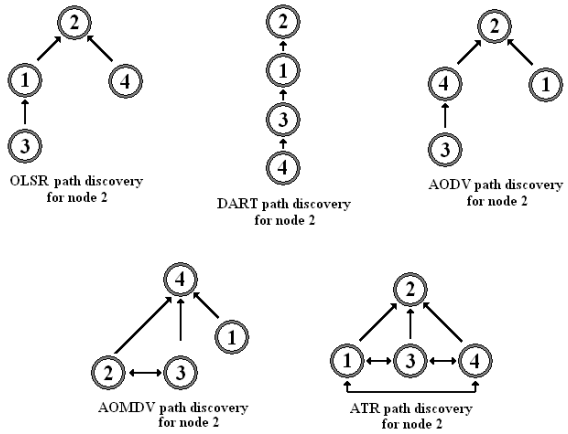


Figure 3.5: Route discovery process

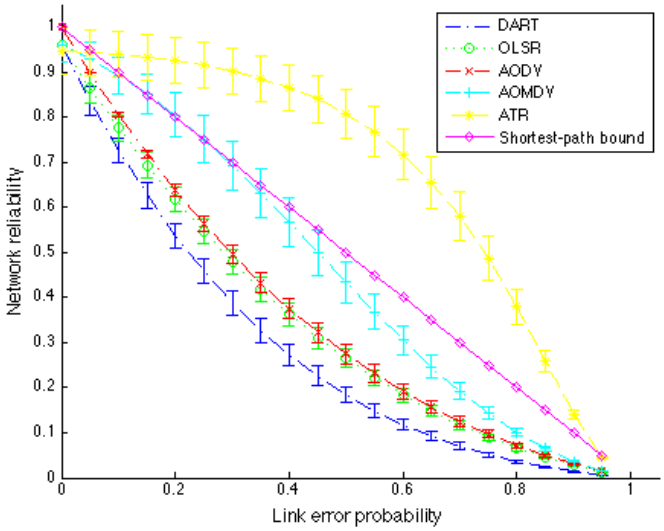


Figure 3.6: TPRR for a 16 nodes network

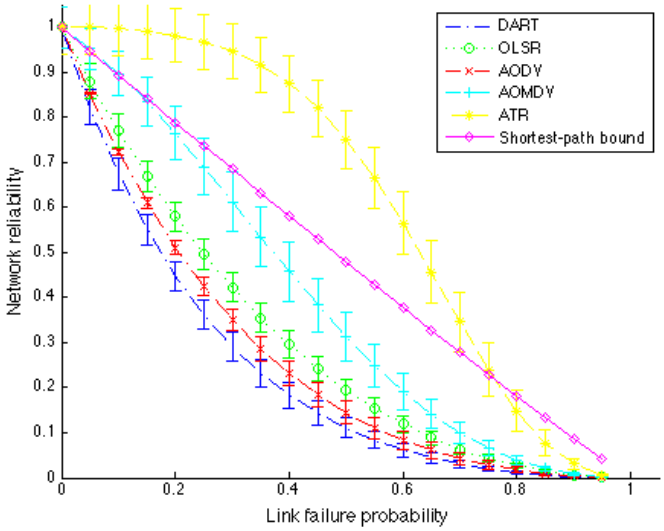


Figure 3.7: TPRR for a 32 nodes network

zone. To analyze the mentioned trade-off, we consider two ATR RDPs which introduce a limitation in the number of discovered routes, in order to keep down the memory overhead. Specifically, in the following we analyze both the *3-limited ATR RDP* and *5-limited ATR RDP*.
Fig.3.8 shows the average TPRR for a network with 16 nodes. It shows that the extra overhead paid by original ATR RDP does not provide a significant performance improvement with respect to the 5-limited ATR one, which is able to exceed the upper bound on TPRR for any shortest-path RDPs for every value of p.

3.4.3 Numerical simulations

In this sub-section we assess the effectiveness of the proposed framework by means of a widely used routing performance metric, the packet delivery ratio (PDR). Clearly, the PDR is an overall metric, which measures the performance of the whole routing process, whereas the TPRR measures the only RDP performances. The PDR measures the probability that a packet is received by the destination, while TPRR estimates the probability that at least one route exists toward the destination. It is evident that there exists dependence between the two metrics. If there is no route toward the destination the PDR has to be zero, and if all packets are correctly received than there exists at least a reliable route toward the destination. Clearly, the availability of good paths, i.e. high reliability, does not imply that the *packet forwarding algorithm* will be able to use them efficiently. Therefore, we have reported on the same figure both the TPRR and the PDR, just to verify the effectiveness of the proposed framework.
More specifically, to evaluate the PDR, we have modified both the physical and the link layer of ns-2. Regarding the former, we have introduced a uniform link-failure probability p for the data packets; clearly, this modification does not affect the routing and MAC packets, preserving so the RDP behavior. Regarding the latter, we have disabled the MAC retransmission for the data packets. The duration of simulation is set to 1500 seconds. The data traffic is modeled as a CBR flow over UDP protocol with a packet rate of 1 packet/s. The data traffic starts at 500s end stops at the end of the simulation. The number of node is 16 and the static network topologies are the same of Section 4.2. To generate the mobile network topologies, we have adopted, as mobility model, the Random Way-Point to simulate a moderate mobility: the speed values are uniformly taken in the [0.5m/s; 5m/s] range and the pause ones in [0s, 100s].

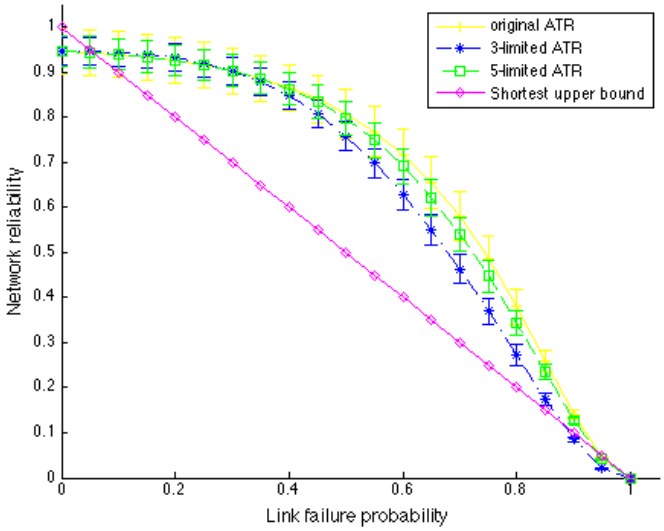


Figure 3.8: ATR RDP analysis

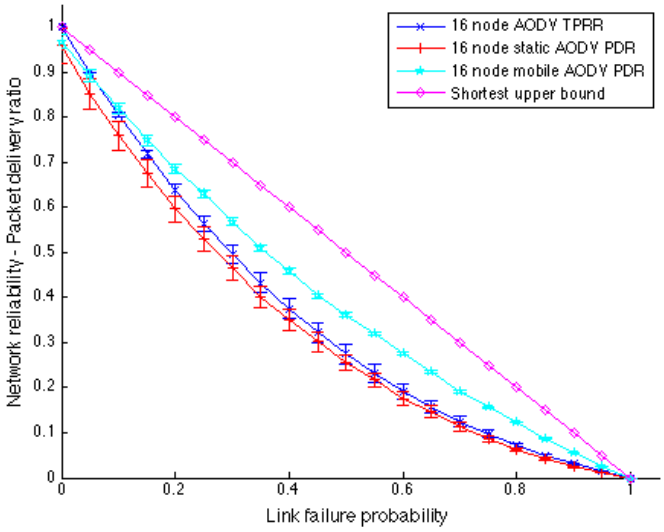


Figure 3.9: AODV PDR analysis

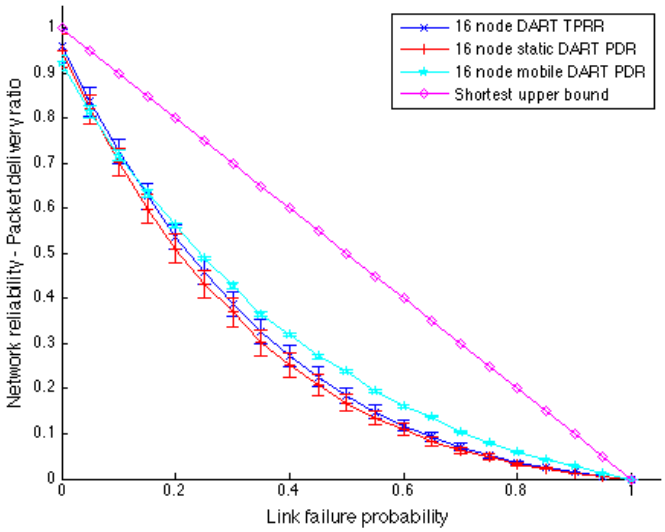


Figure 3.10: DART PDR analysis

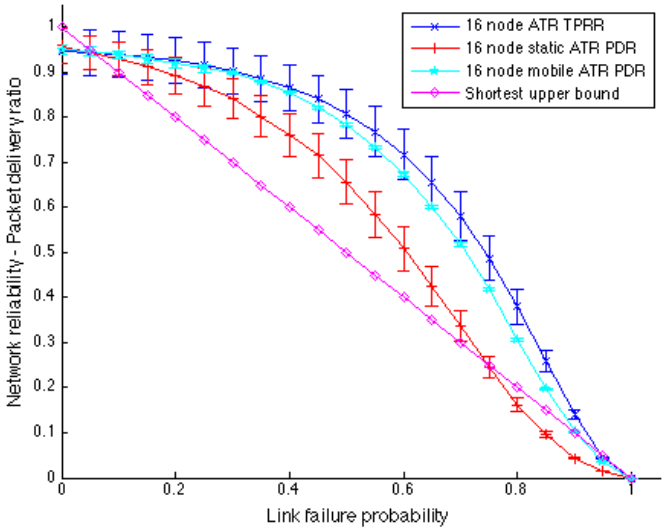


Figure 3.11: ATR PDR analysis

We have performed 100 trials for each protocol and for each value of p. The following figures report the average TPRR, the shortest-path upper bound on TPRR and the average PDR for both static and mobile topologies, as well as the variances. In Fig.3.9 we show the results for AODV. The PDR measured on static topologies agrees very well with the TPRR. Such a behavior can be justified by recognizing that, in this case, the RDP is the one relevant in the overall packet delivery process. In case of mobile topologies, the behavior is less marked. When a packet does not reach the destination, the sender starts a new route discovery. If the topology is static, the new and the previous routes will be the same, giving rise poor performances, whereas, the RDP can get the advantage by node mobility, since in such a case better routes can be discovered and used for long time intervals. The results of Fig.3.9, which refer to DART, confirm the considerations concerning Fig.3.10. Fig.3.11 refers to ATR; in this case, the PDR measured on static topologies does not perfectly agree with the TPRR, even if the two metrics present the same trend. We assume that the ATR *packet forwarding process*, which is liable for choosing one of the available paths, does not pick every time the best route, since it uses only local information for the selection process. The behavior of the PDR in presence of mobility confirms the considerations concerning Fig.3.9.

Chapter 4

Indirect Tree-based Routing

Mobile Ad hoc NETworks (MANETs) and peer-to-peer (P2P) systems are emerging technologies sharing a common underlying decentralized networking paradigm. However, the related research activities have been mainly developed by different research communities, nullifying therefore the idea of an unitary approach able to assure effectiveness integrated solutions. In this chapter, we propose a DHT-based routing protocol which integrates at the network layer both traditional direct routing, i.e. MANET routing, and indirect key-based routing, i.e. P2P routing. The feature of our proposal is the ability to build an overlay network in which the logical and physical proximity agree. The effectiveness of the proposed solution has been proved by numerical simulations.

4.1 Introduction

peer-to-peer (P2P) and mobile ad hoc networks (MANETs) share the same key concepts of self-organization and distributing computing, and both aim to provide connectivity in a completely decentralized environment [105, 106]. Moreover, both lack central entities to which delegate the management and the coordination of the network and relay on a time-variant topology. In fact, in P2P networks the time-variability is due to joining/leaving peers, while in MANET ones it is due to both node mobility and propagation condition instability.

Despite these similarities, the adoption of the P2P paradigm to disseminate and discover information in a MANET scenario arises new and challenging problems [105, 52]. The main issue concerns the layer where they operate:

P2Ps build and maintain overlay networks at the application-layer, assuming the presence of an underlying network routing which assures connectivity among nodes, while MANETs focus on providing a multi-hop wireless connectivity among nodes.
This issue is a major problem in trying to couple a P2P overlay network over a MANET: in [107, 108] it has been proved that simply deploying P2P over MANET may cause poor performances due to the lack of cooperation and communication between the two layers, causing so significant message overhead and redundancy. For these reasons, different cross-layer approaches have been presented and they can be classified according to the adopted solution for the resource discovery procedure.
More specifically, in *unstructured* P2P, peers are unaware of the resources that neighboring peers in the overlay network maintain [109, 110]. So, they typically resolve search requests by means of flooding techniques and rely on resource replication to improve the lookup performance and reliability. Differently, in *structured* P2P networks peers have knowledge about the resources offered by overlay neighbors, usually by resorting to the distribute hash table (DHT) paradigm and, therefore, the search requests are forwarded by means of unicast communications.
Clearly, the scenarios where MANETs operate make unsuitable both flooding and replication mechanisms, except for small networks and/or high joining/leaving peer rates. In the last years structured P2P networks have gained attention: EKTA [111] and DPSR [112] integrate a Pastry-like [58] structured P2P protocol with the DSR routing algorithm, while CROSSRoad [113] integrates a Pastry-like DHT over the OLSR routing algorithm, and VRR [61] proposes a routing algorithm which provides indirect routing by resorting to a Pastry-like structure too. All these techniques associate an identifier, namely a key, to each peer by means of an hash function and organize the keys in a ring structure. Since the identifiers are randomly assigned to peers, the P2P overlay topology is usually built independently from the physical one, and thus no relationship exists between overlay and physical proximity (Fig. 4.1). As shown in [114, 115], this implies that overlay hops can give rise to physical routes which are unnecessary long. MADPastry [116, 117] integrates the Pastry protocol with the AODV routing algorithm and tries to overcome this issue by resorting to clustering. However, the overlay and physical proximity are in someway related only for inter-cluster communications. In [54, 55], it is proposed to associate location-dependent identifiers to nodes with a distribute procedure and to organize node in a tree-based overlay structure.

In the following, according to [54, 55], we give a contribution toward the structured P2P approach presenting a DHT-based routing protocol, namely Indirect Tree-based Routing (ITR), which integrates both traditional direct routing and indirect key-based routing at the network layer. Indirect Tree-based Routing extends the Augmented Tree-based Routing (ATR) protocol presented in Chapter 2, by providing a fully functional P2P network. Unlike [54, 55], we resort to an augmented tree-based structure, in order to assure that the logical and the physical proximity agree, as shown in Fig. 4.3. For both direct and indirect routing, each node maintains a unique routing table which stores only physical 1-hop neighbors, i.e. only peers with which the node can communicate at the link layer. As result, each overlay hop consists of only one physical hop.
To test the effectiveness of our proposal, numerical simulations on IEEE 802.11 technology have been carried out across a wide range of environments and workloads. It is worthwhile to underline that ITR can be accommodated with slight modifications to operate over any link layer technology and, moreover, it does not require any change in both transport and application layers.

4.2 Design

From an operational point of view, Indirect Tree-based Routing like traditional P2P systems: namely, when a node stores a resource, it sends periodically a pointer (a pair <resource identifier, storing peer identifier>) to the *rendezvous-point*, i.e. the node responsible (according to the hash function) for that resource, whereas if a node has to retrieve a resource, it sends a resource query

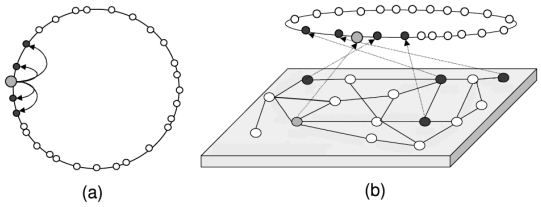


Figure 4.1: Traditional P2P overlay networks

to the rendezvous-point. Both these tasks resort to the algorithm presented in the following, while the rendezvous-point's reply and the following communications needed to retrieve the resource will follow the routing procedure illustrated in Sec. 2.3.
Similarly, for MANET communications, each node periodically sends its current identifier to the rendezvous-point. When a node has to communicate with that node, it will send a identifier query to the rendezvous-point. After the reception of the query reply, the node can start a MANET communication (further detail can be found in Sec. 2.3).

As described above, Indirect Tree-based Routing routes packets accounting for the location-dependent identifier of the destination. Since the identifiers are transient, and since they have to be recovered resorting to indirect key-based routing, both traditional MANET communications and resource queries are forwarded in a similar manner. In the case of traditional MANET communications, a source node knows the IP address of the destination, but not its identifier. In the same way, as regards a resource query, a peer knows the key associated with the needed resource, but not the identity of the peer storing the resource.

To overcome this issue, ITR resorts to two globally known hash functions which return location dependent identifiers, the former defined on the IP address space and latter defined on the resource key space.
Clearly, peer identifiers are assigned to nodes according to the network topology, and thus, there is no assurance that the identifier computed by one of the hash functions is valid, i.e. it has been assigned to a node. As mentioned in Section 4.1, previous proposals overcome the problem organizing the peer identifier space with a virtual ring and forwarding the resource queries toward the ring. The forwarding stops when the query reaches the peer with the iden-

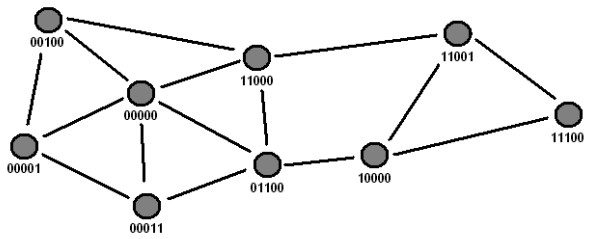


Figure 4.2: Physical network topology

Algorithm 8 forwarding(dst)

```
//l is the bit length of a network address
//src is the forwarder identifier
//dst is the peer identifier computed by the hash function
//computing the level-i sibling to which dst belongs to
//with respect to src
i = level = sibling(src,dst)
bitPosition = 0
nextHop = NULL
cost = maxCost
while nextHop = NULL do
  for each entry in routing table towards the i-th sibling do
    if sibling(dst, entry.nextHop) < level OR (sibling(dst, entry.nextHop)
    == level AND entry.routeCost < cost) then
      nextHop = entry.nextHop
      level = (dst, entry.nextHop)
      cost = entry.routeCost
    end if
  end for
  peerLocation.reset(bitPosition++) //setting the i-th bit to zero
end while
return nextHop
```

tifier closest to the computed identifier, according to a globally known metric. However, each overlay hop may correspond to multiple physical hops (4.1). Differently, our proposal is able to forward both resource and identifier queries without introducing overlay overhead. The procedure is illustrated by Alg. 8, and we make an example to illustrate the basic idea by considering the topology depicted in Fig. 4.2. We suppose that the node *00000* has to forward a resource query (or a identifier query) to the identifier *10100* computed by one of the hash functions. According to Fig. 4.2, the computed identifier is not valid, i.e. it has not been assigned to a node. However, since the query source has at least one entry in its routing table towards the level-4 sibling *1XXXX*, that is the peer with identifier *11000*, the query can be forwarded through the network resorting to physical neighbors as illustrated in Fig. 4.3, reaching so the peer with identifier *11000*. Also the second and the third steps resort to physical neighbors, and the query reaches so the peer with identifier *10000*. Thanks to the augmented tree-based structure, this peer is aware that the iden-

tifier *10100* is not valid since the second section of its table, i.e. the section toward the *101XX* sibling, is empty. At this point, the peer forwards the query following up the tree-structure, namely resetting the destination identifier one bit at time from the right. As result, the query is able to reach a valid identifier, *10000*, without introducing any overlay overhead (three physical hops for three overlay hops).

4.3 Experimental results

To evaluate the performance of Indirect Tree-based Routing, we implemented it as a routing agent on the widely adopted network simulator ns-2 [72] version 2.33 using the wireless extension developed by the CMU Monarch project [20]. We ran different sets of experiments to explore the impact of different workload and environmental parameters on the Indirect Tree-based Routing performances, resorting to an experimental setup very close to the one used in [116, 117] to facilitate a comparison with previous works. Moreover, we compare the its performances with those obtained by the MADPastry protocol [116].
We adopt the standard values for both the physical and the link layer to simulate an IEEE 802.11b network interface with CCK11 modulation and Two-Ray Ground as channel model, resulting in a transmission range of 250 meters and a transmission rate of 11 Mbps. The duration of each simulation experiment is set to 3660 seconds. Nodes move in accordance with the *random way-point* model [118] with no pause time and at a steady speed, and the sizes of the scenario areas are chosen to keep the node density equal to 100 nodes/Km^2.
At the start of the simulation, 50 nodes are randomly allocated on a two-dimensional square space and the nodes start to move immediately. In the interval [700s, 1400s] each node has to store a fixed number of resources,

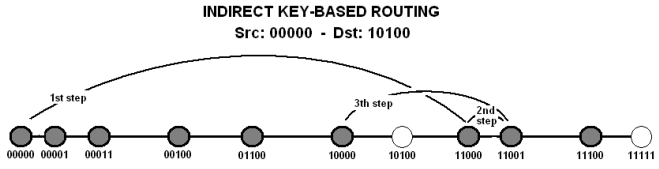


Figure 4.3: Indirect routing

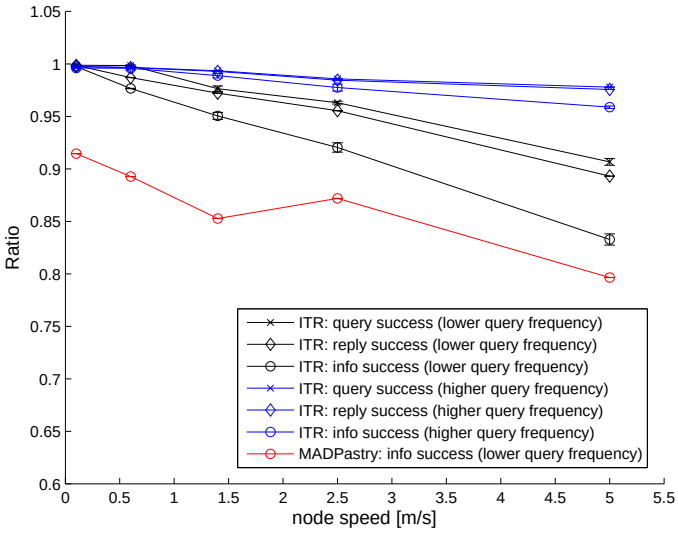


Figure 4.4: Success rates

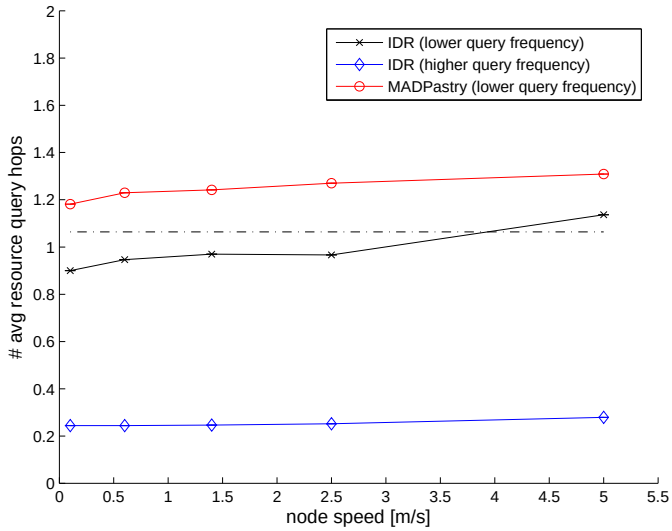


Figure 4.5: Path length

while in the interval [1600s, 3600s], each node sends periodically a query for a resource randomly selected according to a uniform distribution.
Like [116, 117], we evaluate the performances in terms of *query success rate*, i.e. the fraction of resource queries correctly delivered to the rendezvous-point and *network-layer overhead*, i.e. the number of all the network packets generated during the simulation.
Moreover, we introduce two new metrics: the *reply success rate* and the *resource success rate*. The former is defined as the ratio between the number of resource replies correctly delivered to the query sources and the number of generated resource queries. The latter is the ratio between the number of resources correctly delivered to the query sources and the number of generated resource queries, and we resort to it in order to compare the ITR performances with the MADPastry ones. Moreover, we evaluate also the average hop number of resource queries, i.e. the average number of times that a resource query has been forwarded. Such a metric allows us to assess the ability of a P2P protocol to effectively build a physical proximity-aware overlay network.
Regarding Indirect Tree-based Routing, we present the results for two different set of experiments as the node speed grows (Fig. 4.4-4.6). In both sets each node has to store one hundred resources, but the resource query frequency changes, respectively 0.1 and 0.5 query/s, to explore the impact of the caching techniques (the resources are indefinitely cached by each forwarder node, while the resource pointers are cached for 10 seconds). As regard to MADPastry, we set the number of resources to one hundred and the query frequency to 0.1. Each experiment ran five times, and for each metric we estimated both its average value and the standard deviation.
More in detail, Fig. 4.4 we account for the success ratios. Indirect Tree-based Routing outperforms MADPastry in the case of moderate mobility. Simulations, here non reported for sake of brevity, show that the performance gain becomes larger when the resource query frequency increases. If the resource query interval is smaller than the cache retain time, the ITR is able to delivery all the queries to the correct rendezvous-point as well as to retrieve all the required resources. Also in absence of cache hits, the ITR is able to correctly delivery almost all the resource queries and to correctly retrieve more than the 80% of the required resources.
Fig. 4.5 shows the results in terms of resource query hop count. As regards to Indirect-Tree-based Routing, the numerical simulations show that in absence of caching techniques the average overlay hop number agrees with the average physical hop number. In fact, by bounding the average shortest path length $\overline{h}$

measured in hop number as illustrated in Sec. 3.3.2:

$$\overline{h} = \frac{2\sqrt{\frac{n}{\delta}}}{3\sqrt{\pi}r} \tag{4.1}$$

where n is the number of nodes, δ is the node density and r is the transmissions range, we have that $\overline{h} = 1.06$ for a network with 50 nodes. Moreover, the same numerical simulations show the effectiveness of the adopted caching techniques. Regarding to MADPastry, the results shows clearly the presence of an overlay stretch effect.
Finally, Fig. 4.6 accounts for the last metric, the network-layer overhead. MADPastry, thanks to the reactive approach of its routing procedure, is able to outperform Indirect Tree-based Routing. However, we note that the highest values of ITR overhead account also for the differences in the number of resource queries between the two scenarios. We note that the proactive routing table maintenance affects the overhead for about the 20% of the generated routing packets. At the moment, we conjecture that the peak in correspondence of 1.4 m/s is caused by the timing of the distributed procedure for identifier allocation, but the analysis is still carrying on to gain more insight.

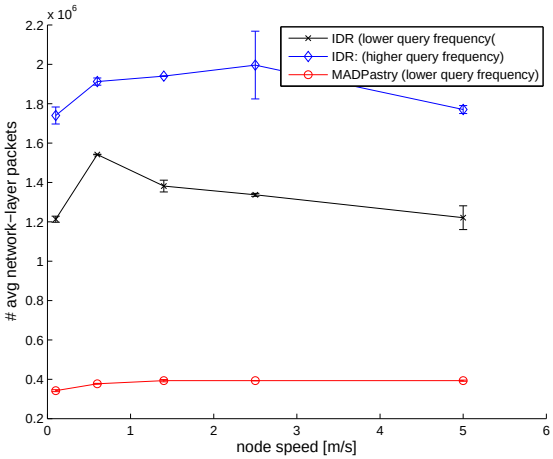


Figure 4.6: Network-layer overhead

Chapter 5

Hierarchical Opportunistic Routing

Opportunistic networks represent one of the most interesting evolution of mobile ad hoc network (MANET) paradigm. Generally speaking, opportunistic networks enable user communication in environments where disconnection and reconnection are likely and link performance is extremely non stationary. In this chapter, we propose a routing protocol, based on the *opportunistic routing* paradigm, able to assure connectivity in ad hoc networks characterized by high link dynamic, namely in disruption tolerant networks (DTNs). By means of numerical analysis, a comparison with both traditional and collaborative routing protocols has been state showing that our proposal is able to provide end-to-end connectivity in DTNs, taking advantage by the link dynamic.

5.1 Introduction

Since *opportunistic networking* paradigm is a very emerging concept, there is no clear definition commonly agreed in the research community. Nevertheless, the strategies adopted to provide end-to-end connectivity in presence of interference-prone wireless communications and transient network topologies exhibit a common key feature which allows one to distinguish opportunistic networking from traditional ad hoc networking [119].
Usually, ad hoc networking tries to *fortify* the environment [120] so that it behave like a wired network. More in detail, the wireless channel is *reinforced* by means of automatic repeat request (ARQ) or forward error control (FEC)

data-link techniques to counteract the time-variant impairment of the wireless propagation, while the transient network topology is *fortified* resorting to multi-path and/or flooding routing techniques.
These approaches are based on two hypotheses. The former is that the network topology is quite dense to assure the presence of a persistent path between each pair of nodes and the latter assures that the wireless propagation conditions are enough stationary to allow a persistent communication among neighbor nodes.
In the last years these assumptions have been gradually relaxed giving rise to the opportunistic networking paradigm, which, rather than counteracts, tries to take advantages by the time-variant nature of the environment to provide end-to-end connectivity in scenarios where traditional networking fails.
Opportunistic networking protocols can be devided in two main classes. The *collaborative routing* protocols exploit the time-variant nature of the network topology to provide connectivity for sparse topologies usually by resorting to a so-called *store-carry-forward* paradigm [121, 122]. delay tolerant networks (DTNs) are a typical application domain for collaborative routing, since they aim to provide connectivity in rural and developing areas where the costs associated with a traditional dense network are no affordable.
The *opportunistic routing* class exploits both the temporal diversity and the broadcast nature of the wireless propagation, usually by resorting to broadcast communications instead of traditional unicast ones, to provide connectivity in presence of hostile wireless propagation conditions. disruption tolerant networks (DTNs) are a typical application domain for opportunistic routing, since they try to provide connectivity to networks characterized by strong shadowing effects as well as intentional interference [123]. In the pioneer work [124] the authors suggest to broadcast the packets and to select the next forwarder at the receiver side to take advantage by all the opportunities provided by the wireless propagation. In other words, they exploit spatial diversity, which can assure more resilience to lossy links.
Since such a routing, referred to as *opportunistic routing*, allows several nodes to receive the same packet, the authors single out a sub-set of neighbor nodes, namely a candidate set, allowed to forward the packet to limit the network flooding.
Such a proposal is however unable to exploit all the opportunities offered by the wireless propagation since the candidate set is chosen at the sender side. In fact, if a node, which is very close to the destination, successfully receives the packet, it can not become the next forwarder unless it has been included in

the candidate set by the forwarding node. Moreover, to single out the nosed belonging to the candidate set it assumes that a link-quality estimation is available.
To overcome the above drawbacks, we propose a routing protocol in the context of disruption tolerant networks (DTNs), namely for networks characterized by intermittent or disruption-prone connectivity [125]. The proposed protocol extends a location-aware addressing schema, first proposed by [55] and presented in Sec. sec:2.2, to match it with opportunistic routing, building so an distribute procedure for candidate selection able to exploit all the opportunities offered by the wireless propagation.
To evaluate the effectiveness of such a proposal, we have carried out numerical simulations to state a performance comparison with two representative routing protocols in presence of hostile propagation conditions, i.e. in presence of shadow fading, across a wide range of environmental conditions.

5.2 System architecture

In this section we describe the proposed protocol, namely the Opportunistic DHT-based Routing (ODR) protocol, providing both an operational overview (Sec. 5.2.1) and a detailed functional description (Sec. 5.2.2 and Sec. 5.2.3).

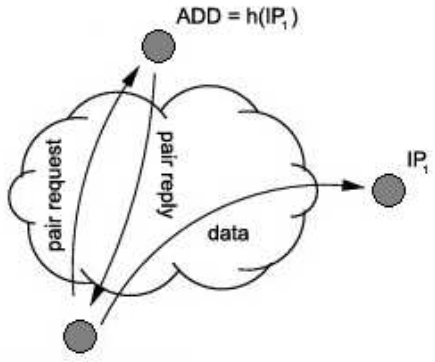


Figure 5.1: Location-dependent address discovery

5.2.1 Overview

As mentioned before, to accomplish the packet routing each forwarder locally broadcasts the packet to all its neighbors, together with an estimate of its distance from the destination. By means of such a distance, the receiving nodes are able to understand if they are potential forwarders, that is if they belong to the candidate set, by comparing their distances with the one stored in the packet header. Clearly, the candidate set is composed by all the neighbors closer than the forwarder to the destination as well as the forwarder.
Each candidate node delays the packet forwarding by an amount of time which depends on its distance estimate from the destination: the more a node is close to the destination, the more the delay is short. A subsequent reception of the same packet from a neighbor closer to the destination allows the node to discard that packet, while a subsequent reception from a farther neighbor gives rise to an acknowledge transmission. This iterative procedure allows that, at each step, the packet has been forwarded by the candidate node closest to the destination.
To limit the overhead due to distance estimation, we exploit a location-aware addressing schema which allows us to group nodes basing on their addresses. This approach lets nodes to estimate their distances from sets of nodes sharing the same address prefix, instead of individually tracking each node.
However, such a procedure requires the availability of a distribute procedure to allow nodes to retrieve the destination addresses before starting a communication. We accomplish this task by resorting to a Distributed Hash Table (DHT) system which exploits a globally known hash function $h(\cdot)$, defined on the IP address space and with values in the location-aware address space.
Every node is part of the DHT system, storing a subset of *pairs* <IP address, location-dependent address> in accordance with the hash function. More in detail, the pair $<ip_1, add_1>$ is stored by the node whose location-dependent address is equal to $h(ip_1)$, namely the *rendezvous-node*. Thus, to find out a location-dependent address a node simply sends a pair request to the rendezvous-node, as shown in Fig. 5.1. After the reception of the pair reply, the node is able to establish the communication. Clearly, the pair request and reply messages resort to the same data routing procedure illustrated above.

5.2.2 Distance estimation

Opportunistic DHT-based Routing (ODR) assigns the location-dependent addresses, namely strings of l bits, to nodes by means of a distribute procedure

which resorts to locally broadcasted hello packets. The address allocation procedure guarantees that nodes sharing a common address prefix are close in the physical topology, allowing so us to easily group nodes.
Each node store a limited-size distance table composed by l entries, one for each set of nodes sharing a common prefix, and the k-th section contains the estimated distance with the *nearest* node whose location-dependent address share a prefix of $l - k$ bits (further detail can be found in Sec. 2.2).
Clearly this approach allows us to reduce the overhead due to distance state maintaining by a logarithm factor, but it arises as well a new problem, since the hierarchy related to the sibling concept gives rise to an estimate inaccuracy. In fact, the k-th section stores the estimated distance towards the nearest node belonging to the set, i.e. the section stores a lower bound on the distance. We propose a solution to this issue in Sec. 5.2.3.
In the following, we resort to the distance metric presented in Sec. 2.3.2. This metric aims to estimate the expected number of packet transmissions (including the retransmissions) required to successfully deliver a packet to the ultimate destination. Clearly, the Opportunistic DHT-based Routing protocol can be easily extended to different metrics.

5.2.3 Packet forwarding

The packet forwarding process consists of three steps: the candidate selection, the candidate election and the candidate acknowledgment. To accomplish these steps, each node resorts to two queues. The former, namely the *packet queue*, stores the packets waiting to be forwarded, i.e. the packets for which the node is a candidate forwarder. The latter, namely the *ack queue*, stores for acknowledgment purposes the packets that have not anymore to be forwarded.

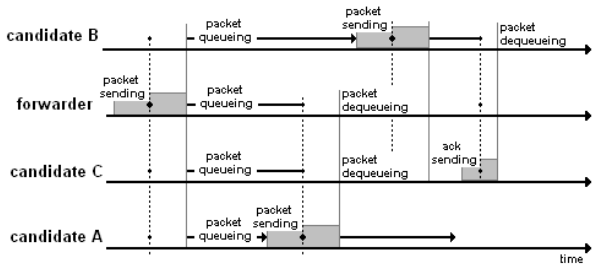


Figure 5.2: Typical ODR packet forwarding

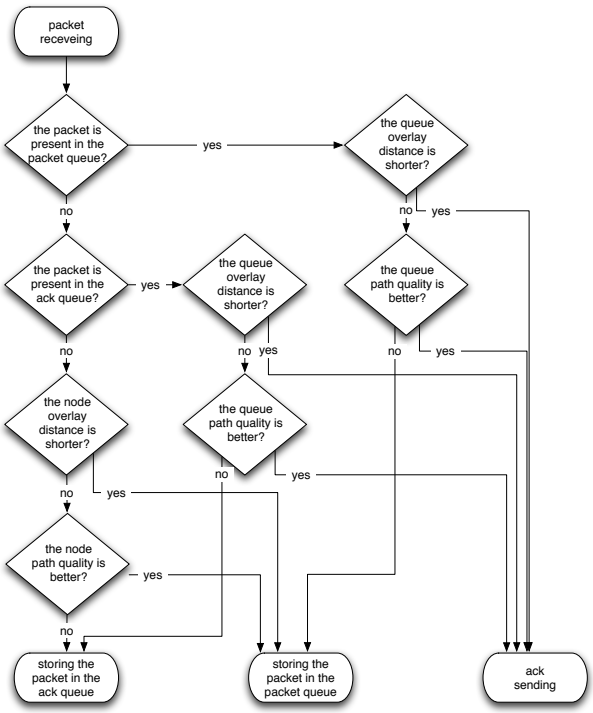


Figure 5.3: Packet forwarding process

The candidate selection ensures that, at each step, only nodes closer than the forwarder to the destination are allowed to re-forward the packet. More in detail, when a node forwards a packet, it stores in the packet header its location-dependent address along with its estimate distance from the destination, and then it locally broadcasts the packet.
A receiving node checks if its overlay distance to the destination, i.e. the length of the address prefix shared by the node address and the destination one, is shorter than the forwarding overlay distance and then checks if its path quality is better than the forwarder one. If both the checks fail, the node does not belong to the candidate set and it stores the packet in its ack queue. Differently, it stores the packet in its packet queue together with a delay time evaluated according to the following relation:

$$delay = \tau * \frac{q_p(r,d)}{q_p(f,d)} * \left[\frac{1}{o_d(r,d) - o_d(f,d) + 1} \right] \tag{5.1}$$

where τ is the maximum delay time (2 seconds in our implementation), f is the forwarding node, r is the receiving node, d is the destination one, q_p is the estimated quality and o_d is the overlay distance. By means of this heuristic approach for the delay estimation, we account for the estimate inaccuracy mentioned in Sec. 5.2.2, since the ratio between the estimated qualities ratio is weighted by a factor, i.e. the term in the square brackets in (5.1) depending on the overlay distances, which measures the size of the clusters of nodes, namely the siblings, to which the qualities refer to.
Thus, the delay times allow nodes to implement a distributed candidate election procedure, by exploiting a TDMA-based scheduling: since the closest node stores in the packet header its distance estimate from the destination and since it is the first that forwards the packet, the other candidates can listen such a packet transmission and therefore give up to the packet forwarding.
Such a strategy does not require explicit acknowledgment for each packet forwarding, although it is not tolerant to to the hidden terminal problem, as illustrated in Fig. 5.2, where the candidate *B* is unable to listen for the packet forwarding of *A*, and thus it forwards the packet as well. In such a case, it is necessary to resort to explicit acknowledgment, namely *C* stores the packet sent by *A* in the ack queue and thus, it is able to acknowledge to *B* that the packet was successfully received by a node (*A*) closer to destination.
Fig. 5.3 gives a detailed description of the whole forwarding process resorting to a flow chart representation.

5.3 Performance analysis

To evaluate the performance of the proposed protocol, we have implemented it as a routing agent on the widely adopted network simulator ns-2 [72] version 2.33 using the wireless extension developed by the CMU Monarch project [20].
We have compared the performances achieved by our protocol with those of two representative routing protocols, namely the Ad Hoc On-Demand Distance Vector (AODV) [42] and the Epidemic Routing [126]. The former is a traditional ad hoc routing protocol based on persistent unicast communications among neighbor nodes. The latter exploits the *store-carry-forward* paradigm and it has been proposed to provide connectivity for Delay Tolerant Networks.

5.3.1 Experimental setup

Usually, performance analyses for both traditional and opportunistic networking adopt a deterministic radio propagation model which is clearly unrealistic in the case of disruption tolerant networks. Therefore, we consider a propagation model, the *Shadowing* one, which accounts for the long-term fading effects by means of a zero-mean Gaussian variable $N(0, \sigma)$. According to it, the received mean power $P_{dB}(d)$ at distance d is:

$$P_{dB}(d) = P_{dB}(d_0) - \log \beta(d/d_0) + N(0, \sigma) \tag{5.2}$$

where $P_{dB}(d_0)$ is the received mean power at the first meter, β is the path-loss exponent and σ is the shadow deviation, both empirically determined for a certain environment. In our performance analysis, we set β to 3.8 to model a shadowed urban area, and we vary σ from 1.0 to 11.0dB in order to assess the behavior of the analyzed protocols under a wide range of variability levels of the propagation conditions. Moreover, we set the values of the parameters of the data link layer to simulate an IEEE 802.11b Orinoco network interface [77] with long preamble, CCK11 modulation and two-handshake mechanism, resulting in a transmission range of roughly 35 meters and in a nominal transmission rate of 11 Mbps.
The duration of each experiment is 3000 seconds and the nodes move in accordance with the *random way-point* model [118] with no pause time and at a steady speed over a rectangular $750 * 175\ m^2$ flat area.
After the initial 1000 seconds, a certain fraction of nodes starts to generate data traffic, since the initial period is used to assure that the routing protocols reach a steady state. Each node involved in the traffic generation sends packets of 1000 bytes to each other node in the network, deferring the subsequent transmissions of 1 second. The adopted data traffic allows us to assess the protocol performances under infrequent and concurrent transmissions, as it happens in the case of emergency message dissemination.

5.3.2 Numerical results

Since we are primarily concerned with Disruption Tolerant Networks, the performance comparison aims to evaluate the impact of the link dynamic for sparse networks in several environmental conditions. In fact, taking into account both the transmission range and the node density, the mean node connectivity degree is lower than 1 for all the considered scenarios. This value is reasonable to assure the presence of network partitions [79].

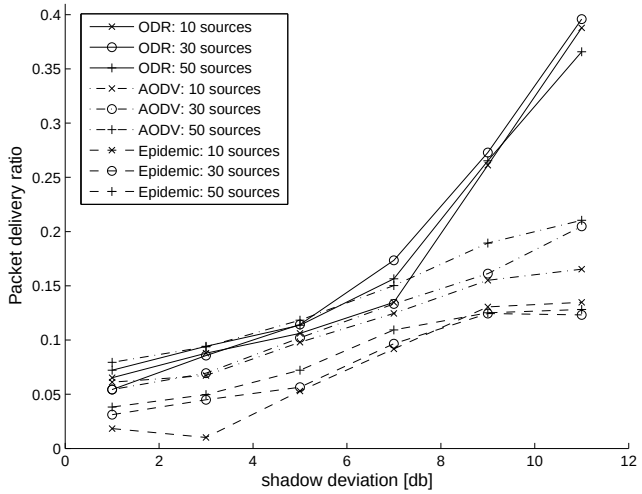


Figure 5.4: Packet delivery ratio for different data loads

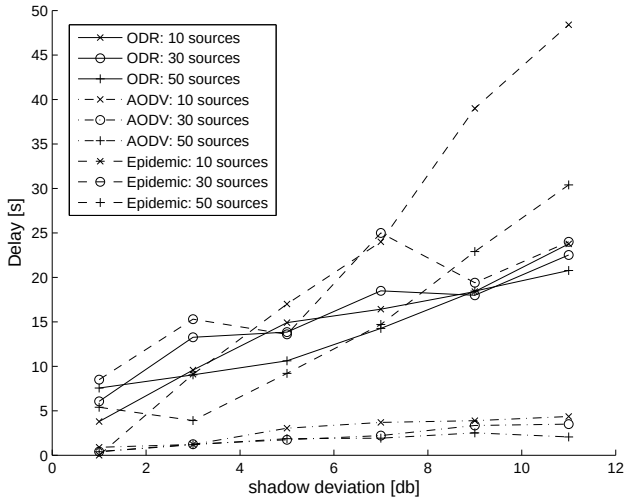


Figure 5.5: Delay for different data loads

The first set of experiments (Fig. 5.4 and Fig. 5.5) refers to a scenario with 50 nodes and a node speed equals to $0.01\ m/s$ and a growing number of nodes which generate data traffic.

As regards to the average packet delivery ratio (Fig. 5.4), the results show that the performances of all the analyzed protocols improve as the shadow deviation increases.
It is worthily to note that these surprising behavior is reasonable, also if unintuitive. In fact, the physical layer model of ns-2 accounts only for the effects of the long-term fading over the packet power (5.2), neglecting so the effects of information corruption due to fading as well. For such a reason, the fading introduces a time-diversity, which is exploited by the routing protocols to provide end-to-end connectivity.
More in detail, the proposed protocol outperforms the other one as the variability of the wireless propagation grows, providing so an effectively end-to-end connectivity (a delivery ratio equals to 0.4 can satisfy the requirements of several not real-time applications). Moreover, the same figure shows that the performances of all the compared protocols are substantially unaffected by the increase of data load, implying so that we have modeled a sustainable data traffic.
Fig. 5.5 shows the average packet delay vs. the shadow deviation. Clearly, both the Opportunistic DHT-based Routing and the Epidemic Routing protocols suffer of higher delay times with respect to AODV. The results of Epidemic Routing are expected, since it resorts to the store-carry-forward paradigm, i.e. the forwarder stores the packet until it moves near the destination. As regard to ODR, the delays measure both the time needed to retrieve the location-dependent address and the time for data packet forwarding, i.e. each delay measures the amount of time needed to route three packets.
In the second set of experiments the number of the nodes in the network grows and the node speed is equal to $0.01\ m/s$. Fig. 5.6 shows the average packet delivery ratio vs. the shadow deviation: clearly, all the protocol performances decreases as the network becomes more sparse and the AODV performs worst, since it is designed for dense ad hoc networks. The Opportunistic DHT-based Routing outperforms both Epidemic and AODV in all the considered environments, achieving the best performances for the higher values of shadow deviation as well.

In the third set of experiments, we analyze the node mobility effects. More in detail, we simulate a network with 50 nodes and 10 traffic sources and the results are presented in Fig. 5.6. Both ODR and Epidemic Routing perform worse than AODV as the node speed increases, since AODV is able to exploit a moderate mobility to achieve better performances. We do not present the results for Epidemic Routing in the case of speed value equals to $1\ m/s$ since in

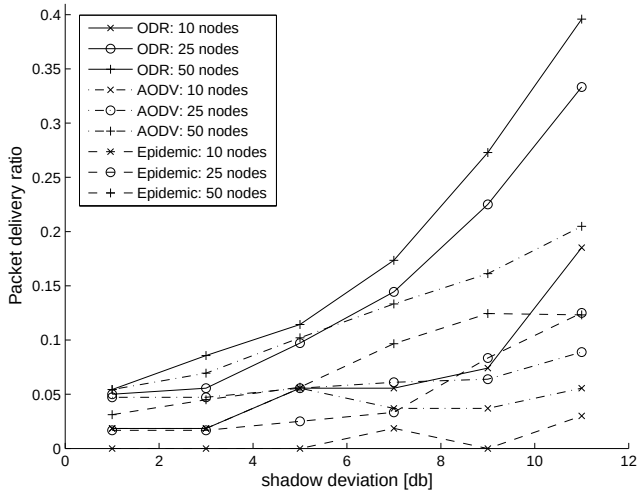


Figure 5.6: Packet delivery ratio for different density values

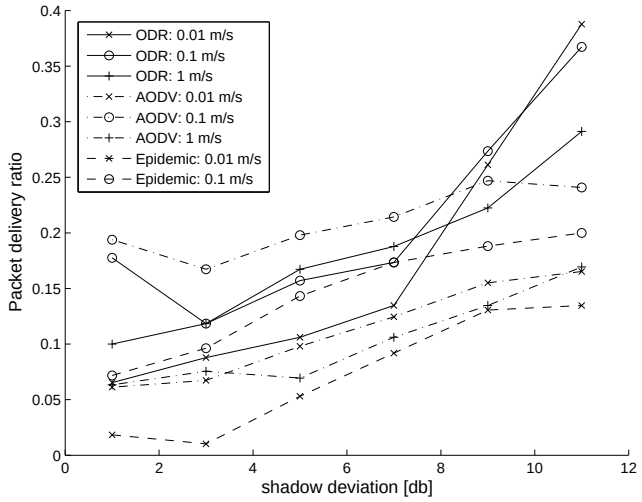


Figure 5.7: Packet delivery ratio for different speed values

such a case the delivery ratios are very small since such a protocol needs bidirectional unicast communications which become unavailable in case of sparse networks with high mobility.

Finally, as regards to routing overhead, the numerical results, not presented for the sake of brevity, shows that Opportunistic DHT-based Routing exhibits the worse performances with respect to both AODV and Epidemic Routing.

Conclusions

In this thesis, the adoption of the hierarchical routing paradigm to achieve a scalable network layer for ad hoc networks has been proposed. The main concept of hierarchical routing is to keep, at any node, complete routing information about nodes which are close to it and partial information about nodes located further away.

It has been shown that the overhead needed by current networking protocols for ad hoc networks increases so fast with the number of nodes that it eventually consumes all of the available bandwidth also in networks with moderate size. One of the main reasons for such a lack of scalability is that they have been proposed for wired networks and modified to cope with ad hoc scenarios. More specifically, they are based on the assumption that node identity equals routing address, that is, they exploit static addressing which of course is not yet valid in ad hoc scenarios.

Recently, some routing protocols have exploited the idea of decoupling identification from location, by resorting to distribute hash table services, which are used to distribute the nodeÕs location information throughout the network. In this paper, we give a contribution toward such an approach by focusing our attention on the problem of implementing a scalable network layer.

In this thesis a routing protocol for ad hoc networks, referred to as the Augmented Tree-based Routing (ATR) protocol, has been proposed. Such a protocol exploits both a location-aware addressing schema and a distribute hash table (DHT) system. The adopted addressing schema allows nodes to exploit hierarchical routing, limiting so the overhead introduced in the network, while the DHT system provides the mapping between transient identifiers and node identities. Simulation results and performance comparisons with existing protocols substantiate the effectiveness of the proposed protocol for large ad-hoc networks operating in presence of channel hostility and moderate mobility.

Since the Augmented Tree-based Routing protocol adopts a multi-path strategy and since most studies in the area of multi-path routing focus on

heuristic methods and the performances of these strategies are commonly evaluated by numerical simulations, an analytical framework to evaluate the performance gain achieved by multi-path routing has been proposed. The framework is based on graph theory and on terminal-pair routing reliability (TPRR) as performance measure. By resorting to numerical simulations based on a widely adopted routing performance metric, namely the packet delivery ratio, the proposed framework has been validated and the results show the effectiveness of TPRR as performance measure both in static and dynamic topologies.

Moreover, some features of the proposed protocol have been exploited design a peer-to-peer (P2P) system over a mobile ad hoc network (MANET) resorting to a cross-layer approach. It has been proved that simply deploying P2P systems over MANETs may cause poor performances. By coupling both the direct and the indirect key-based routing at the network layer and by resorting to the same hierarchical address space structure of ATR, we are able to build a P2P overlay network in which the logical proximity agrees with the physical one, limiting so the message overhead and avoiding the redundancy. The simulation results substantiate the effectiveness of such a system across different environmental conditions.

Finally, by extending the proposed location-aware addressing to match with the opportunistic forwarding protocol, a novel routing protocol for disruption tolerant network (DTN) and delay tolerant network (DTN) has been proposed. By exploiting both the temporal diversity and the broadcast nature of the wireless propagation, such a protocol can enable connectivity in ad hoc environments characterized by non stationary wireless propagation as well as sparse topologies. By means of numerical analysis, a comparison with both traditional and collaborative routing protocols has been stated to evaluate the effectiveness of the proposed solution.

Basing on the above results, the suggestions for future work regards their validation, namely the results must be substantiate by resorting to an experimental test-bed. The clear separation between link, network and transport layer is difficult to maintain in a wireless environment, as pointed out by the IETF MANET working group itself. Although the presented results have been obtained resorting to realistic channel models, they are anyway based on assumptions that have to be validated by means of experimental results. Moreover, future work could include the design of peer-to-peer systems based on the opportunistic networking paradigm, which could improve their reliability without introducing data replication.

Bibliography

[1] Leonard Kleinrock and Farouk Kamoun. Hierarchical routing for large networks; performance evaluation and optimization. *Computer Networks*, 1:155–174, 1977.

[2] Mark Weiser. The computer for the twenty-first century. *Scientific American*, pages 94–101, Sep 1991.

[3] I. Chlamtac, M. Conti, and J. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, July 2003.

[4] Laura Marie Feeney, Bengt Ahlgren, and Assar Westerlund. Spontaneous networking: an application-oriented approach to ad hoc networking. *IEEE Communications Magazine*, 39:176–181, 2001.

[5] James A. Freebersyser and Barry Leiner. *A DoD perspective on mobile Ad hoc networks*. Addison-Wesley Longman Publishing Co., Inc., 2001.

[6] W. Fifer and F. Bruno. The low-cost packet radio. *Proceedings of the IEEE*, 75(1):33–42, 1987.

[7] B.M. Leiner, R.J. Ruther, and A.R. Sastry. Goals and challenges of the darpa glomo program [global mobile information systems]. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 3(6):34–43, Dec 1996.

[8] Ieee 802.11 wlan web site.

[9] Motorola mesh networks.

[10] M. Scott Corson, Joseph P. Macker, and Gregory H. Cirincione. Internet-based mobile ad hoc networking. *IEEE Internet Computing*, 3(4):63–70, 1999.

[11] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, July 2002.

[12] Marco Conti. *Body, personal, and local ad hoc wireless networks*. CRC Press, Inc., 2003.

[13] Bluetooth special interest group web site.

[14] Etsi hiperlan web site.

[15] Ieee 802.15.4 wpan web site.

[16] Ieee 802.16 wman web site.

[17] Internet protocol. Request for Comments 791, Internet Engineering Task Force (IETF), September 1981.

[18] Kwan-Wu Chin, John Judge, Aidan Williams, and Roger Kermode. Implementation experience with manet routing protocols. *SIGCOMM Comput. Commun. Rev.*, 32(5):49–59, 2002.

[19] Elizabeth Belding-Royer. Routing approaches in mobile ad hoc networks. In *Ad Hoc Networking*, pages 275–300. IEEE Press Wiley, 2003.

[20] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, and J.A. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, 1998.

[21] Y. Tseng, S. Ni, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, March 2002.

[22] Curt Schurgers, Gautam Kulkarni, and Mani B. Srivastava. Distributed on-demand address assignment in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 13(10):1056–1065, 2002.

[23] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[24] Yi-Bing Lin and Imrich Chlamtac. *Wireless and Mobile Network Architectures*. John Wiley & Sons, Inc., 2000.

[25] M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, Nov/Dec 2001.

[26] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84, 1998.

[27] S. Giordano and M. Hamdi. Mobility management: the virtual home region. Technical Report SSC/1999/037, EPFL, October 1999.

[28] Ivan Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. Technical Report TR99-10, University of Ottawa, Canada, September 1999.

[29] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, 2000.

[30] Pai-Hsiang Hsiao. Geographical region summary service for geographical routing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):25–39, 2001.

[31] Joseph P. Macker and M. Scott Corson. Mobile ad hoc networks (manets): routing technology for dynamic wireless networking. In *Ad Hoc Networking*, pages 275–300. IEEE Press Wiley, 2003.

[32] S. R. Das, R. Castaneda, and J. Yan. Comparative performance evaluation of routing protocols for mobile, ad hoc. In *IC3N '98: Proceedings of the International Conference on Computer Communications and Networks*, page 153, 1998.

[33] Elizabeth M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6:46–55, 1999.

[34] Gary Scott Malkin and Martha E. Steenstrup. *Distance-vector routing*. Prentice Hall International (UK) Ltd., 1995.

[35] John Moy. *Link-state routing*. Prentice Hall International (UK) Ltd., 1995.

[36] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM '94: ACM Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[37] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, a. Qayyum, and L. Viennot. Optimized link state routing protocol. In *IEEE INMIC*, December 2001.

[38] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9*, page 298, 2002.

[39] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[40] Robert Castaneda, Samir R. Das, and Mahesh K. Marina. Query localization techniques for on-demand routing protocols in ad hoc networks. *Wirel. Netw.*, 8(2/3):137–151, 2002.

[41] S.R. Das, C.E. Perkins, and E.M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 3–12, 2000.

[42] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[43] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector (aodv) routing. Request for Comments 3561, Internet Engineering Task Force (IETF), July 2003.

[44] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[45] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, 2:562–566, Oct 1997.

[46] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Trans. Netw.*, 9(4):427–438, 2001.

[47] Prince Samar, Marc R. Pearlman, and Zygmunt J. Haas. Hybrid routing: the pursuit of an adaptable and scalable routing framework for ad hoc networks. In *The handbook of ad hoc wireless networks*, pages 245–262. CRC Press, Inc., 2003.

[48] Stefano Basagni. Distributed clustering for ad hoc networks. In *ISPAN '99: Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99)*, page 310, 1999.

[49] B. Chen and R. Morris. L+: Scalable landmark routing and address lookup for multi-hop wireless networks. Technical Report MIT LCS-TR-837, Massachusetts Institute of Technology, March 2002.

[50] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *Journal of Parallel and Distributed Computing*, 63(2):110–122, February 2003.

[51] Elizabeth M. Belding-Royer. Hierarchical routing in ad hoc mobile networks. *Wireless Communications and Mobile Computing*, 2(5):515–532, 2002.

[52] Aline Carneiro Viana, Marcelo Dias de Amorim, Serge Fdida, and José Ferreira de Rezende. Self-organization in spontaneous networks: the approach of dht-based routing protocols. *Ad Hoc Networks*, 3(5):589–606, September 2005.

[53] A.C. Viana, M.D. de Amorim, S. Fdida, and J.F. de Rezende. Indirect routing using distributed location information. In *PerCom 2003: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, pages 224–231, March 2003.

[54] J. Eriksson, M. Faloutsos, and S.V. Krishnamurthy. Peernet: Pushing peer-to-peer down the stack. In *IPTPS*, pages 268–277, 2003.

[55] J. Eriksson, M. Faloutsos, and S.V. Krishnamurthy. Dart: dynamic address routing for scalable ad hoc and mesh networks. *IEEE/ACM Transactions on Networking*, 15(1):119–132, 2007.

[56] Shu Du, Ahamed Khan, Santashil PalChaudhuri, Ansley Post, Amit Kumar Saha, Peter Druschel, David B. Johnson, and Rudolf Riedi. Safari: A self-organizing, hierarchical architecture for scalable ad hoc networking. *Ad Hoc Network*, 6(4):485–507, 2008.

[57] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, Aug 2001.

[58] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.

[59] M. Caleffi, G. Ferraiuolo, and L. Paura. Augmented tree-based routing protocol for scalable ad hoc networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–6, Oct. 2007.

[60] Marcello Caleffi and Luigi Paura. Augmented tree-based routing: Dht routing for scalable ad hoc networks. under review.

[61] M. Caesar, M. Castro, E.B. Nightingale, G. O'Shea, and A. Rowstron. Virtual ring routing: network routing inspired by dhts. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 351–362, 2006.

[62] Li Layuan, Li Chunlin, and Yaun Peiyan. Performance evaluation and simulations of routing protocols in ad hoc networks. *Computer Communications*, 30(8):1890–1898, June 2007.

[63] M. Caleffi, G. Ferraiuolo, and L. Paura. On reliability of dynamic addressing routing protocols in mobile ad hoc networks. In *WRECOM '07: Proceedings of the Wireless Rural and Emergency Communications Conference*, October 2007.

[64] N. H. Vaidya. Weak duplicate address detection in mobile ad hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international*

symposium on Mobile ad hoc networking & computing, pages 206–216, 2002.

[65] S. Nesargi and R. Prakash. Manetconf: Configuration of hosts in a mobile ad hoc network. In *INFOCOM '02: Proceedings of the 21th Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1059–1068, 2002.

[66] H. Zhou, L.M. Ni, and M.W. Mutka. Prophet address allocation for large scale manets. *Ad Hoc Networks*, 1(4):423–434, November 2003.

[67] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Darmouth College, July 2003.

[68] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networkks*, 11(4):419–434, 2005.

[69] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172. ACM, 2001.

[70] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *IPTPS: 1st International workshop on Peer-To-Peer Systems*, pages 53–65, 2002.

[71] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *STOC '97: Proceedings of the 29-th ACM symposium on Theory of computing*, pages 654–663, 1997.

[72] The VINT Project. The ns manual (formerly ns notes and documentation).

[73] Marcello Caleffi, Giancarlo Ferraiuolo, and Luigi Paura. A reliability-based framework for multi-path routing analysis in mobile ad-hoc networks. *International Journal of Communication Networks and Distributed Systems 2008*, 1(4-5-6):507–523, 2008.

[74] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev.*, 34(4):121–132, 2004.

[75] Mineo Takai, Jay Martin, and Rajive Bagrodia. Effects of wireless physical layer modeling in mobile ad hoc networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 87–94. ACM, 2001.

[76] Wu Xiuchao and A.L. Ananda. Link characteristics estimation for ieee 802.11 dcf based wlan. *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 302–309, November 2004.

[77] Proxim. Orinoco 11b client pc card specification, 2004.

[78] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM '03: Proceedings of the 22th Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1312–1321, 2003.

[79] C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 80–91, 2002.

[80] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, Mar 2000.

[81] Sabyasachi Roy, Dimitrios Koutsonikolas, Saumitra Das, and Y. Charlie Hu. High-throughput multicast routing metrics in wireless mesh networks. *Ad Hoc Network*, 6(6):878–899, 2008.

[82] Ron Banner and Ariel Orda. Multipath routing algorithms for congestion minimization. *IEEE/ACM Trans. Netw.*, 15(2):413–424, 2007.

[83] Asis Nasipuri, Robert Castaneda, and Samir R. Das. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mob. Netw. Appl.*, 6(4):339–349, 2001.

[84] S.-J. Lee and M. Gerla. Aodv-br: backup routing in ad hoc networks. In *WCN 2000: Proceedings of IEEE Wireless Communications and Networking Conference*, pages 1311–1316, 2000.

[85] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *ICC '01: Proceedings of the IEEE International Conference on Communications*, pages 3201–3205, 2001.

[86] Wuping Xu, Puliu Yan, and Delin Xia. Similar node-disjoint multipaths routing in wireless ad hoc networks. In *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, pages 731–734, September 2005.

[87] Song Guo, O. Yang, and Yantai Shu. Improving source routing reliability in mobile ad hoc networks. *Parallel and Distributed Systems, IEEE Transactions on*, 16(4):362–373, April 2005.

[88] Antonios Argyriou and Vijay Madisetti. Using a new protocol to enhance path reliability and realize load balancing in mobile ad hoc networks. *Ad Hoc Networks*, 4(1):60–74, January 2006.

[89] Wei Kuang Lai, Sheng-Yu Hsiao, and Yuh-Chung Lin. Adaptive backup routing for ad-hoc networks. *Compututer Communications*, 30(2):453–464, 2007.

[90] P.P. Pham and S. Perreau. Performance analysis of reactive shortest path and multipath routing mechanism with load balance. *INFOCOM 2003: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, 1:251–259, 2003.

[91] A. Agarwal and B. Jain. Routing reliability analysis of segmented backup paths in mobile ad hoc networks. *ICPWC 2005: IEEE International Conference on Personal Wireless Communications*, pages 52–56, Januray 2005.

[92] Lei Wei. Connectivity reliability of large scale random ad hoc networks. In *MILCOM 2003: IEEE Military Communications Conference*, pages 411–415, October 2003.

[93] A.M. Abbas and B.N. Jain. An analytical framework for path reliabilities in mobile ad hoc networks. In *ISCC 2003: Proceedings od Eighth IEEE International Symposium on Computers and Communication*, pages 63–68, 2003.

[94] H.M.F. AboElFotoh, S.S. Iyengar, and K. Chakrabarty. Computing reliability and message delay for cooperative wireless distributed sensor

networks subject to random failures. *Reliability, IEEE Transactions on*, 54(1):145–155, March 2005.

[95] Akhilesh Shrestha, Liudong Xing, and Hong Liu. Infrastructure communication reliability of wireless sensor networks. In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, pages 250–257, September 2006.

[96] A. Tsirigos and Z.J. Haas. Multipath routing in the presence of frequent topological changes. *Communications Magazine, IEEE*, 39(11):132–138, November 2001.

[97] A. Tsirigos and Z.J. Haas. Analysis of multipath routing, part i: the effect on the packet delivery ratio. *Wireless Communications, IEEE Transactions on*, 3(1):138–146, January 2004.

[98] A. Tsirigos and Z.J. Haas. Analysis of multipath routing, part 2: mitigation of the effects of frequently changing network topologies. *Wireless Communications, IEEE Transactions on*, 3(2):500–511, March 2004.

[99] M. O. Ball. Complexity of networks reliability computations. *Networks*, 10:153–165, 1980.

[100] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy. Modeling path duration distributions in manets and their impact on reactive routing protocols. *Selected Areas in Communications, IEEE Journal on*, 22(7):1357–1373, September 2004.

[101] Hung-Yau Lin, Sy-Yen Kuo, and Fu-Min Yeh. Minimal cutset enumeration and network reliability evaluation by recursive merge and bdd. In *ISCC 2003: Proceedings Eighth IEEE International Symposium on Computers and Communication*, pages 1341–1346, 2003.

[102] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, Mar 2000.

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

[104] Mahesh K. Marina and Samir R. Das. Ad hoc on-demand multipath distance vector routing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):92–93, 2002.

[105] Rüdiger Schollmeier, Ingo Gruber, and Michael Finkenzeller. Routing in mobile ad-hoc and peer-to-peer networks a comparison. In *Revised Papers from the NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing*, pages 172–186, London, UK, 2002. Springer-Verlag.

[106] A. Oram. *Peer-to-Peer - Harnessing the power of disruptive technologies*. OŔeillt, 2001.

[107] Gang Ding and Bharat Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. In *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 104–108. IEEE Computer Society, October 2004.

[108] Leonardo B. Oliveira, Isabela G. Siqueira, and Antonio A. F. Loureiro. On the performance of ad hoc routing protocols under a peer-to-peer application. *Journal of Parallel and Distributed Computing*, 65(11):1337–1347, 2005.

[109] Marco Conti, Enrico Gregori, and Giovanni Turi. A cross-layer optimization of gnutella for mobile ad hoc networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 343–354, 2005.

[110] Bin Tang, Zongheng Zhou, Anand Kashyap, and Tzi cker Chiueh. An integrated approach for p2p file sharing on multi-hop wireless networks. In *WiMob'2005: IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, volume 3, pages 268– 274, August 2005.

[111] H. Pucha, S. M. Das, and Y. C. Hu. Ekta: an efficient dht substrate for distributed applications in mobile ad hoc networks. In *WMCSA 2004: Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 163–173, 2004.

[112] H. Pucha, S. M. Das, and Y. Hu. Imposed route reuse in ad hoc network routing protocols using structured peer-to-peer overlay routing. *IEEE Transactions on Parallel and Distributed Systems*, 17(12):1452–1467, 2006.

[113] Franca Delmastro. From pastry to crossroad: Cross-layer ring overlay for ad hoc networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 60–64, 2005.

[114] Matei Ripeanu, Adriana Iamnitchi, and Ian Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1):50–57, 2002.

[115] R. Schollmeier, I. Gruber, and F. Niethammer. Protocol for peer-to-peer networking in mobile environments. In *ICCCN 2003: Proceedings of the 12th International Conference on Computer Communications and Networks*, pages 121–127, October 2003.

[116] Thomas Zahn and Jochen Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. In *Proc. of 5th Workshop on Applications and Services in Wireless Networks (ASWN2005)*, June 2005.

[117] Kei Takeshita, Masahiro Sasabe, and Hirotaka Nakano. Mobile p2p networks for highly dynamic environments. In *PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 453–457, 2008.

[118] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[119] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, November 2006.

[120] Hua Zhu and Kejie Lu. Resilient opportunistic forwarding: Issues and challenges. *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, Oct. 2007.

[121] Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[122] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li S. Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, volume 37, pages 96–107, October 2002.

[123] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald. When tcp breaks: Delay- and disruption- tolerant networking. *Internet Computing, IEEE*, 10(4):72–78, July-Aug. 2006.

[124] Sanjit Biswas and Robert Morris. Exor: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.*, 35(4):133–144, 2005.

[125] Marcus Brunner, Lars Eggert, Kevin Fall, Jörg Ott, and Lars Wolf. Dagstuhl seminar on disruption tolerant networking. *SIGCOMM Comput. Commun. Rev.*, 35(3):69–72, 2005.

[126] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks, April 2000.