UNIVERSITÉ DE LIÈGE
Faculté des Sciences Appliquées
Département d'Electricité,
Electronique et Informatique

Topologically-aware construction of unstructured overlays over ad hoc networks

Thèse présentée par

Sandrine Calomme

en vue de l'obtention du titre
de Docteur en Sciences de l'Ingénieur

Année académique 2008–2009

Abstract

The number of electronic devices, equipped with a wireless interface has exploded over the last decades. Unfortunately, their usage is often restricted to the connection to a wired infrastructure, even for local communications. That is quite surprising as the research area of infrastructureless, or ad hoc, networks has flourished for years. The literature on ad hoc networks is very rich, but their usage almost inexistent. Potential users have plenty of solutions at hand, but do not exploit them.

Even if ad hoc networks allow us to get rid of the infrastructure, they still require an implicit agreement on the solution to use. Nevertheless, it is very difficult to pick in the rich panel of protocols the best one, that would fit any ad hoc user in any ad hoc network. As an example, it has been demonstrated, for the routing, that each protocol has definite advantages and disadvantages, in every different scenario, and is well suited for certain situations [RT99].

Yet, a salient feature of ad hoc networks is precisely that the panel of situations is very large. The ad hoc network conditions are influenced by the number of ad hoc users, their relative positions, their capabilities, their mobility pattern, the applications they use, the traffic load and type, and so forth. Moreover, the users may themselves be heterogeneous, with different hardware and software capabilities, mobile behaviour and communication needs. Hence, there is a particular need in ad hoc networking for flexible techniques.

We contribute to this problem by studying the feasibility of overlay routing and giving some hints in that direction.

We explain how the overlay members can avoid the expensive process of building an overlay topology, before using their customised routing application. The rationale exploits the broadcast nature of ad hoc networks, and is qualified as a Reactive Overlay Approach. We also detail an elementary reactive overlay routing application and test it, by simulations, in a variety of conditions, including the network and overlay densities. This performance study shows the feasibility and the efficiency of overlay routing applications developed according to the Reactive Overlay Approach. It also evidences the impact of using an appropriate value for the *neighbourhood range*, defined as the maximum number of hops between two overlay neighbours.

Hence, we detail the *critical neighbourhood range (CNR) problem*, which, in short, consists in determining the minimum neighbourhood range value that generates a connected overlay. We solve it in the asymptotic case, i.e. when the number of nodes in the underlay or the size of the field tends to infinity. The mathematical results are interesting in the sense that they can be useful for a better understanding of the interaction between various typical characteristics of a connected overlay topology on an ad hoc network.

However, the theoretical, asymptotic, CNR is not adequate in practice. We thus also explore heuristics for estimating the CNR. We present a simple protocol which estimates an appropriate neighbourhood range for overlay routing applications. For the purpose of its evaluation, we define general performance criteria based on overlay flooding. Namely, these are the delivery percentage, bandwidth consumption and time duration of flooding on the overlay.

The main drawback of the Reactive Overlay Approach is the amount of bandwidth consumed during the flooding of overlay route requests. Hence, we also consider the Proactive Overlay Approach, which consists in building the overlay topology before the emission of any overlay broadcast message, and maintaining it. We compare the quality of various overlay topologies in the static case. We finally describe and evaluate the Overlay Topology Control (OTC) protocol, that maintains, in a mobile context, the overlay topology as close as possible to the overlay topology evaluated as the best.

The main objection that would arise against overlay routing on ad hoc networks is that the ad hoc nodes do generally own poor resources and that overlay routing consumes them even more than native routing.

The feasibility study we conducted with the reactive approach and the evaluation of OTC, designed in the context of the proactive overlay approach, confirm that the consumption of resources must be handled carefully. Nevertheless, they show that this problem is not insurmountable.

Acknowledgements

On the first place, I would like to acknowledge the advisor of this PhD, professor Guy Leduc, at University of Liège. He gave me the opportunity of doing this thesis, of course. But he also let me freedom of defining the project, the time for conducting it, his very helpful feedback and his confidence. Thanks for all.

My sincere acknowledgements towards Chris Blondia, Isabelle Guérin Lassous, Piet Demeester, David Simplot-Ryl and Pierre Wolper who accepted to be members of my thesis committee.

I would be very ungrateful if I had not written some lines (in French) to my parents. *A vous, mes parents, qui m'ont donné, parmi bien d'autres choses, le goût d'apprendre et de raisonner. Vous avez souvent prétendu que vous n'étiez pour rien dans ma réussite scolaire. Lorsque je vous observe maintenant interagir avec mes enfants, je comprends que c'est faux. Merci pour votre patience, votre ouverture d'esprit et votre rigueur.*

One important actor of this work, even if he is not cited on the list of publications, is my husband, Pierre. Many persons could tell that the period during which their husband or wife was doing and/or writing their PhD was not the most simple of their life together. I am afraid I was a typical example. Fortunately, I had the double chance of benefiting from the endless patience and support of my husband, and from his good advice taken from his own experience as PhD student and researcher.

There are two other persons who did not help (at all) on the technical point of view, but who had also an indisputable effect on the morale aspect. Niels and Louna, your natural energy and joy in life are a delight.

Finally, I want to thank all the other members of the Research Unit in Networking, in particular Sylvain and Cyril for their regular help, and Ibtissam. The coffee used to have a better taste before her departure from RUN.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Wireless devices become more and more present in our environment.

The technology that has most visibly grown in the last twenty years is the cellular phones. The current number of GSM users by the turn of the century was 250 million and is still expected to grow.

Many houses and companies install wireless routers, which let the users communicate through their laptop, PDA or phone without the need of staying connected by means of cables.

GSM and WiFi are already successful commercial stories that are now incorporated at the heart of our modern life.

They are typical examples of infrastructure-based wireless technologies. The data sent by the user must first reach the closest wireless base station or access point before being forwarded until its final destination. The base stations, access points and servers must be deployed before the network can be used, and constitute the so-called infrastructure.

The wired links offer a high bandwidth and low losses in comparison with wireless ones. Consequently, confining the wireless communication to the first hop guarantees some level of performance. It also allows the users to get equipped with a simple end device. Tasks as routing, resource reservation or network management are let to wired routers and their operators. Mobility is also dealt with by the network operator.

However, theoretically, the wired infrastructure is not a requirement. A network could be composed of wireless devices only. If two users are located too far from each other, their data can simply be forwarded by other wireless devices located between them, following a multi-hop path. This type of communication is qualified as infrastructureless. An ad hoc network is a set of wireless devices capable of organising themselves for communicating, without the need of pre-established infrastructure.

Research on ad hoc networks, initially called packet radio networks, began in the seventies, driven by military needs. Since the mid nineties, it has drawn more and more attention, catalysed by the impressive growth of wireless communicating devices and their users. Companies, universities, and home users are already equipped with 802.11 antennas. Many PDAs and laptops come with built-in WiFi, or can be equipped with a 802.11 add-on card. WiFi Voice over IP phones are also available on the market place. In metropolitan areas, the current concentration of wireless users in the free-band allows the creation of a unique, global ad hoc wireless network. However, this potentiality is not exploited. Wireless communications are still restricted to the first hop, that gives access to the wired infrastructure. It is often argued that the killer application for ad hoc networks is yet to arrive. However, one could argue that any user could be interested in just using its current applications anytime and anywhere, that is in an ad hoc manner.

1.2 Overview of the problem addressed by this thesis

Ad hoc networking allows to get rid of the infrastructure, but still requires some agreements, at least implicit, on the technical solutions the users will use. Nevertheless, it is very difficult to pick in the rich panel of protocols the best one, that would fit any ad hoc user in any ad hoc network.

As an example, it has been demonstrated, for the routing, that each protocol has definite advantages and disadvantages, in every different scenario, and is well suited for certain situations [RT99]. Yet, a salient feature of ad hoc networks is precisely that the panel of situations is very large. The ad hoc network conditions are influenced by the number of ad hoc users, their relative positions, their capabilities, their mobility pattern, the applications they use, the traffic load and type, and so forth. Moreover, the users may themselves be heterogeneous, with different hardware and software capabilities, mobile behaviour and communication needs. Hence, there is a particular need in ad hoc networking for flexible techniques.

We contribute to this problem by studying the feasibility of overlay routing and giving some hints in that direction.

Our motivations are the following:

- The overlay technique could be a response to the variety of ad hoc scenarios, by letting the users select and adapt an appropriate routing solution during their participation in an ad hoc network, in function of their perception of the network conditions. The core of the routing, until now, has relied on an implicit agreement, before the formation of the network, on one protocol that behaves well in many situations. This procedure worked fine for the Internet. It however seems, in the case of ad hoc networks, that the resources are so scarce, and the variety of conditions so large and unpredictable, that a single routing procedure cannot give satisfaction in any case.

- The overlay technique could be a response to the heterogeneity of users inside a given ad hoc network by letting any group of similar users, a so-called community, employ their custom solution.

- An overlay approach allows a transparent coexistence between the community members and other network nodes, without the use of a new packet header nor any modification of the ad hoc routing protocol used by all nodes. It increases flexibility in routing by permitting a partial deployment in the network of new protocols.

- The communication paths passing through the members of a given community could reasonably be preferred to paths going across external nodes. At each overlay hop, the intermediary overlay peers, reached before the final overlay message receiver, could benefit from the data contained, analyse it, and/or improve it. As an effect, the total consumption of resources by the overlay application could be lower with the overlay routing procedure than without it, even if the routing process alone is more greedy. The overlay application user could also perceive an increase in quality due to the actions performed at the intermediary overlay nodes.

Our investigations focus on which overlay topologies would offer a good ground for efficient overlay routing. In particular, we answer the following questions:

- Is it mandatory, in ad hoc networks, to maintain an overlay topology before running an overlay routing process ? If not, is it recommended or should it be advised against ?

- How can we model an overlay topology ?

- How can we evaluate the quality of an overlay topology with respect to overlay routing ?

- Which type of overlay topology seems the best ?

- How to build and maintain such a topology ?

1.3 Dissertation outline

In Chapter 2, we review the literature on overlays. We define the technique in details, with its advantages and drawbacks. We summarise its applications in the Internet, and the existing proposals for using overlays in ad hoc networks. We also summarise in this chapter the researches done in the context of ad hoc networks that are exploited in our work.

In Chapter 3, we test the feasibility of overlay routing. This preliminary study is conducted on AODV, a very popular ad hoc routing protocol belonging to the reactive family. In this context, we show that overlay routes can be found without maintaining an overlay topology. We call this type of overlay routing procedure the reactive overlay approach. We show that its efficiency depends on the good setting of a parameter, the overlay neighbourhood range.

In Chapter 4, we conduct an analytical study of this parameter, while in Chapter 5, we provide a heuristic for determining a good value for it.

In Chapter 6, we adopt a different approach, where the overlay topology is proactively maintained. We first study the performance of various overlay topologies in the static case and then describe the Overlay Topology Control (OTC) protocol. This protocol maintains, in a mobile context, the overlay topology as close as possible to the overlay topology evaluated as the best.

We finally summarise and discuss our results. We also propose some further investigations on building overlays in ad hoc networks.

1.4 Related publications

Most of the material presented in the body of this dissertation has been published at international conferences. Here is a list matching for each technical chapter the related publication(s).

Chapter 3: Feasibility of the Reactive Overlay Approach

- **Performance Study of an Overlay Approach to Active Routing in Ad Hoc Networks** [CL04]
  Third Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2004

Chapter 4: The Critical Neighbourhood Range Theoretical Study

- **The Critical Neighbourhood Range for Asymptotic Overlay Connectivity in Dense Ad Hoc Networks** [CL05]
  Fourth Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2005

- **The Critical Neighbourhood Range for Asymptotic Overlay Connectivity in Ad Hoc Networks** [CL06a]
  Ad Hoc & Sensor Wireless Networks journal

Chapter 5: The Critical Neighbourhood Range Heuristic Study

- **Efficient and Resilient Overlay Topologies over Ad Hoc Networks** [CL07]
  Second International Workshop on Self-Organizing Systems, IWSOS 2007

- **Elaboration d'un protocole de contrôle de topologie pour les overlays bâtis sur des réseaux ad hoc** [CL06b]
  (In French) Colloque Francophone sur l'Ingénierie des Protocoles, CFIP 2006

- **Neighbour-Based Overlay Topology Control in Ad Hoc Networks** [CL06c]
  Poster at ACM MobiHoc 2006

Chapter 6: The Proactive Overlay Approach

- The three publications mentioned for previous chapter

- **An overlay maintenance protocol for overlay routing on top of ad hoc networks** [CL08]
  IFIP Networking 2008

Chapter 2

Related and useful work

2.1 Chapter outline

The first part of this chapter is dedicated to ad hoc networks, with a focus on topology control and routing. We next present a survey on overlay networks. Previous works relative to the building of overlays on top of ad hoc networks are presented at the end of this second part. We finally position our own researches.

2.2 Ad hoc networks

An ad hoc network is defined as a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralised administration [BMJ$^+$98].

The *Packet Radio Network (PrNet)* [JT87] and *Survivable Adaptive Networks (SURAN)* [Bey90] projects are generally considered as the first researches on ad hoc networks. These were funded by the *Defense Advanced Research Projects Agency (DARPA)* in the 1970's. Their objective was to quickly deploy a communication network, in a region where the infrastructure is unavailable or insecure. These tactical networks had also to be survivable and to allow mobility.

The technological concept of switching packets in a multihop fashion, with wireless nodes acting as routers as well as end hosts, emerged in these times and are still the core of the ad hoc network technology.

In the 1990's, the research on ad hoc networks developed again, mostly in the scientific and industry communities, catalysed by new wireless technologies – Bluetooth, IEEE802.11, Hyperlan –, the broad availability of wireless cards working in the free radio frequency band, and by the appearance on the market of low-cost portable devices.

There was clearly an opportunity for bringing the technology to civilian applications. Emergency situations is a classical example. In case of fire, flood, or earthquake, the communication infrastructure could be severely damaged. An ad hoc network can be quickly deployed between the rescuers and evolve dynamically during their operations.

Ad hoc networks are also very helpful when a collection of users meet, require to communicate directly, and then separate. For example during a conference, an exhibition or trade event. The users can communicate rapidly, and unnecessary installation costs for such temporary networks are avoided.

The MANET (Mobile Ad-hoc Network) group was formed in 1997 with the aim of developing a routing framework for running IP-based protocols in ad hoc networks. During this decade, a lot of multihop routing protocols were designed and tested with simulation tools.

The ad hoc network research continued to flourish since then. The wireless environment brings additional complexity to the problems traditionally covered by the networking field. The error and packet loss rates are much higher. Mobility is no more an eventuality to cope with, but is a key design feature for routing algorithms. The energy resources and the medium carrying capacity are limited. The topology, unknown and variable, must be constantly controlled. Finally, the spontaneous nature of ad hoc networks makes heterogeneity an inherent feature. Each node may have different resources, and purpose.

These new challenges have thrilled researchers for the last decades and this will probably continue.

In the next sections, we give an overview of the fields in ad hoc research relevant to our own work: Topology control and Routing.

2.2.1 Topology Control

A group of wireless nodes may potentially form an ad hoc network if and only if there exists a multihop path between each pair of them, i.e. if it is strongly connected.

To achieve connectivity, each ad hoc node could use its maximum transmission range, in order to reach many neighbours. However, mobile devices have a limited amount of battery power. Moreover, this would create a lot of interferences, reducing the overall capacity of the network.

The goal of topology control (TC) is to dynamically adapt the nodes' transmitting range in order to maintain some global property of the communication graph (e.g. connectivity) while reducing the energy consumed by node transceivers [San05].

Note that topology control is only a sub-case of the more general problem of power control, discussed in [KK05]. When addressing TC, it is implicitly assumed that the energy employed by all nodes for sending packets varies much less than routes. By contrast, in the context of power control, every node may individually and independently adapt the energy it spends for sending each packet.

The point graph model [SH97] is generally used for analytical and probabilistic studies of controlled topologies. Under this model, the nodes antennas are assumed to cover a perfect circle area. This is a strong simplification of reality because obstacles are not considered and effects such as reflection, diffraction and scattering are ignored. Nevertheless, it is very useful for understanding basic properties of controlled topologies.

There exist two classes of topology control algorithms, namely homogeneous and non-homogeneous.

Homogeneous

With a homogeneous topology control algorithm, all nodes adopt the same transmission range value.

A common power level assignment ensures that all links are bi-directional. As discussed in [KK05], the bi-directionality of links is implicitly assumed in many routing protocols. For example because they are based on the Distributed Bellman-Ford algorithm [MW77], or make use of route reversals for discovering a path. The IEEE 802.11 medium access protocol also rely on bi-directionality as a Clear to Send (CTS) packet from node n should be heard by all nodes that are susceptible to alter the reception of node n's data packet if they would emit a packet simultaneously.

Assigning a common power level to all nodes has also the advantage of not sacrificing too much of a network's potential carrying capacity. In [GK00], it is proved in the asymptotic case that the best power control solution for a network composed of n nodes, would not provide a global capacity higher than $\sqrt{\log n}$ times the capacity obtained when every node emits every packet at a distance equal to the critical transmission range (CTR). The CTR problem consists of determining the minimum value that generates a connected network. When node positions are known in advance, the CTR equals the longest edge of the minimum spanning tree [SMH99]. However, in many realistic scenarios, this information is not available.

On the plane or in three dimensional-space, analytical results were only obtained for the asymptotic case, that is when the number of nodes tends to infinity. In finite networks, the CTR has been determined only for nodes randomly placed on a line [DM02]. Moreover, this result is difficult to interpret.

First studies of graph connectivity were developed in the context of the random graphs theory. A random graph is a graph generated by some random procedure [Bol85]. In 1960, Erdos and Rényi [ER60] showed that for many monotone-increasing properties of random graphs, like connectivity, graphs of a size slightly less than a certain threshold are very unlikely to have the property, whereas graphs with a few more graph edges are almost certain to have it. This is known as a phase transition phenomenon.

In classical random graph models, there is no *a priori* structure. All vertices are equivalent and there is no correlation between the existence of different edges. In ad hoc and sensor networks, nodes are more likely to be direct neighbours if they are located close to each other. Therefore random geometric graphs are more suited to model them. Random geometric graphs are constructed by placing points at random according to some arbitrary specified density function on a d-dimensional Euclidean space and connecting nearby points [Pen03]. Properties such as the longest edge of the minimal spanning tree, or the nearest neighbour link can then be established. Random Geometric Graphs were extensively studied by Penrose, who proved for example that connectivity in dense networks occurs when the last isolated node receives a neighbour. Experimentations later showed that a large amount of energy is spent for connecting a relatively small number of nodes. For a uniform random distribution of nodes on the plane, setting the range to the half of the critical transmitting range connects about 90% of the nodes [SB03]. Some of the geometric random graphs results can be applied in the study of connectivity in ad hoc and sensor networks [Pen99]. Various transition phenomena can also be observed in geometric random graphs [KWB01]. Monotone properties for this class of graphs have sharp threshold [GRK04]. Asymptotically, as the network density tends to infinity, a critical value transmission range can thus be established [GK99], [LWWY04], [WY05].

The CTR problem in dense networks has also been studied with the percolation theory in [DTH02]. With this theory, the nodes are distributed on R^2 following a Poisson distribution of density λ. Phase transitions are also observed. An important result is that there exists a critical density λ_c under which the probability that an arbitrary node belongs to a giant component is null and above which this probability is not null. It has been used in several studies handling connectivity or coverage.

In [SB03], the fixed radius model used in the geometric random graphs theory is extended by adding a new geometric parameter: the network deployment region size. The region may be fixed or may also enlarge when the number of nodes grows. In the former case, the network is said dense, because the geographical density of nodes also tends to infinity. In the latter, it is qualified as sparse. The authors use the occupancy theory [KSC78] and obtain an asymptotic formula for the CTR in sparse as well as in dense networks. In the occupancy theory study, n nodes are distributed in C cells. The theory allows to determine the probability of having no empty cell when $n, c \to \infty$. In [MP03], the authors exploit the same model and, using a bin-covery technique, derive tighter bounds for the asymptotic connectivity. We will use these results in Chapter 4.

Recent studies consider more advanced radio models and in particular take into account interferences [OD03, Kos05, HM04, BLRS07].

As described above, the homogeneous range assignment problem is quite tractable with probabilistic tools. It allowed researchers to study the foundations of TC. However, there are not many representative protocols for the homogeneous family because non-homogeneous protocols provide better performance in practice. We discuss the non-homogeneous range assignment problem below. Let us however mention the COMPOW protocol [NKSK02], that chooses a common power level in a set of discrete values. The solution consists of running multiple independent instances of a table-driven routing protocol, one at each admissible power level. By examining the different routing table obtained, the COMPOW agent figures out the lowest value which keeps the network connected.

Non-homogeneous

Non-homogeneous topology control problems relate to the case where a different transmitting range value can be assigned to every node. A non-homogeneous range assignment may produce asymmetric links. As stated above, this feature should preferably be avoided because they reduce the performance of most upper layer routing protocols. The non-homogeneous topology control problem is thus generally relaxed to set communication ranges so as to reduce the energy consumed in the network while preserving a connected backbone of symmetric links. The energy consumption may be evaluated by summing the energy consumed for transmitting one packet between each pair of neighbours. Even when the node placement on the plane is homogeneous, calculating the minimum value is a NP-hard problem [CPS99]. A variant of the problem consists of calculating the lowest energy path between each pair of nodes, and summing the energy of all these paths. Ideally, the controlled topology should be an *energy spanner* of the initial graph, also called *maxpower graph*, with all nodes allowed to use their maximal transmitting range. This means that the routes of the controlled topology should be a constant factor away from the energy-optimal routes on the maxpower graph. The controlled topology should moreover have a linear number of edges and a bounded node degree. Finally, it should be easily computable in a distributed and localised fashion.

An ad hoc network can be represented by an undirected graph $G = (V, E)$ in the Euclidean space, V being the set of ad hoc nodes and E the set of communication links. One could find in the Geometric Graphs literature several well-studied set of edges, with interesting properties related to the topology control problems. Computational Geometry then offers algorithms for building these structures. We define here three of them: the Minimum Spanning Tree (MST), the Relative Neighbourhood Graph (RNG) and the Gabriel Graph (GG).

Consider a pair of nodes $(u, v) \in V^2$ and $d(u, v)$ the distance that separates them. This pair of nodes is an edge of the GG if and only if the disk touching both u and v and having $d(u, v)$ as a diameter, does not contain any other node $w \in V$, including on its boundary.

The pair of nodes (u, v) is moreover an edge of the RNG if and only if the lune defined by the intersection of the circle centred at u with radius $d(u, v)$ and of the circle centred at v with the same radius is also empty.

Both the GG and RNG have a bounded distance stretch factor. This means that the total length, expressed in meters, of the shortest route between any pair of nodes on the GG and RNG obtained from a given max-power graph G is bounded by a multiple of its optimal length on G. This implies that they are also power spanners. In particular, the Gabriel Graph is energy-optimal.

A last non-homogeneous TC problem consists of calculating the broadcast tree rooted at any node with minimal energy cost. This is also an NP-hard problem.

Small power levels provide low-power routes and were thought to imply an increase of the overall network capacity by reducing the MAC layer contention. In [BvRWZ04], authors disprove that low interference is a consequence to sparseness of the resulting topology. The reason is that one must distinguish the logical degree of a node, i.e. its number of neighbours in the controlled topology, and its physical degree, i.e. the number of nodes within its transmission range. The expected interference observed in a network is related to the physical degree of nodes and not to their logical degree [San05]. Based on this observation, [BLRS03] proposed a non-homogeneous transmission range assignment mechanism that guarantees a bound on the maximal physical node degree. The protocol, called *k-Neigh*, moreover ensures that the controlled topology is made of bidirectional links only.

It has been demonstrated in [San05] that for any control protocol that preserves worst-case connectivity, there exists a placement of n nodes such that the maximum physical node degree in the controlled topology equals $n - 1$. However, setting the minimum number of physical neighbours to 9 is sufficient to obtain connected networks with high probability for ad hoc networks with the number of nodes ranging from 50 to 500 [BLRS03]. Each node that runs the *k-Neigh* protocol increases its transmission range until it covers this number of neighbours. A selection is then made on the set of covered physical neighbours, in order to present a power-aware logical topology to the upper-layer routing protocol. The *k-Neigh* protocol sets the maximal number of physical neighbours, thus it limits interferences. However, it preserves the connectivity only with a high probability. Oppositely, the protocol described in [WZ04], called *XTC*, does guarantee the connectivity of the controlled topology every time the max-power topology is connected, but the bounded degree property is satisfied with a high probability only. These two protocols share a similar operating mode. Potential neighbours are first discovered through the emission of hello messages. For *k-Neigh*, the number of neighbour candidates is bounded by the maximum physical nodes degree set, while for XTC all nodes heard are valid candidates. The current list of candidates is indicated in the hello messages, with an estimation of the distance or power necessary to reach them. Each node thus knows its admissible two-hops neighbourhood and is able to weight every potential edge. Using this information, it selects its final neighbourhood by applying a pruning criterion. We use a similar mechanism in the design of the Overlay Topology Control (OTC) protocol presented in Chapter 6.

2.2.2 Routing protocols

We discuss here the large class of unicast, single-channel routing protocols. Unicast is a basic primitive in any network. At the physical and link layer, the IEEE 802.11 technology has emerged for wireless LANs and MANs. Devices communicating through an 802.11 standard share a single logical channel.

Ad hoc routing protocols can first be categorised in three large groups: geographic position based, hierarchical and flat routing [HXG02].

We do not present geographic position based routing protocols. They rely on the assumption that nodes own a Global Position System (GPS) or at least run a mechanism for estimating their location. This information is useful for reducing the bandwidth consumed during route establishment, and allows the design of scalable and efficient solutions. A complete review of position based protocols can be found in [GSB03].

Literature on routing in ad hoc networks is very rich. Various taxonomies have been proposed. Their purpose is to evidence common design features and performance of the set of protocols in each defined class. We consider here two classification criteria:

- Is the protocol proactive or reactive ?
- Is the protocol hierarchical ?

As explained below, the reactive and hierarchical approaches are not compatible. Our classification criteria thus define three main routing families, namely proactive (flat), reactive and hierarchical[1].

The resources are generally scarce in ad hoc networks. Some nodes must conserve energy and the amount of bandwidth available is much lower than in wired networks. Besides these constraints, MANET's routing protocols must face two major challenges, which are scalability and mobility.

We briefly present below some key protocols in each category. We also discuss the performance of the protocols in each category with respect to the network size and the mobility level.

[1]Protocols generally qualified as hybrid in the literature fall under our hierarchical category

Proactive protocols

With a proactive routing protocol, each node builds and maintains a forwarding table. The forwarding table contains the necessary information for sending packets to any other destination node. This is a proactive approach, as this table is constantly updated even if the majority of the routes contained are not used. When a node must send a packet, the next hop onto its destination is directly available in its forwarding table. The proactive protocols are also called table-driven.

Proactive protocols can further be divided into two categories: uniform and non uniform [Fee99]. In a uniform protocol, every node adopts the same behaviour when receiving a routing control message.

The Destination-Sequenced Distance Vector routing protocol (DSDV) belongs to this category. It is an adaptation of the Bellman-Ford algorithm for the mobile context. Each node periodically emits its routing table and holds an individual sequence number, that it increments at each table broadcast. Routing entries also include sequence numbers. When a routing table entry is modified, its sequence number is replaced by the one included in the control message that raised the update. A modification may occur only if the control message sequence number is higher or equal. This means that the received topological information is fresher than the one used for building the entry. This mechanism avoids the creation of routing loops.

The Wireless Routing Protocol (WRP) is a refinement of DSDV. Each node indicates its next hop onto any destination in the distance vectors it sends. A neighbour will take into account the information it reads in a distance vector for a given destination if and only if it is not itself the predecessor announced for this destination. This avoids the count-to-infinity problem. A few optimisations are also brought through the use of four routing tables.

The Global State Routing (GSR) is based on the traditional Link State algorithm. A node does not flood its link state information but sends it to its neighbours only. The latter update their network view and in turn communicate it to their neighbours. The number of routing control messages is thus much lower but their size is relatively large, and grows with the network size.

Proactive uniform protocols are noticeable, as they were the first routing algorithms appropriate for mobile multihop networks. However, they all consume a lot of bandwidth. Consequently, they do not scale. Moreover, the amount of routing control messages increases with mobility. If the volume of control traffic was kept constant, some routing accuracy would be lost, and consequently, data packets would be dropped. Their main advantages in small and relatively static networks are the permanent availability of all routes and a constant volume of control traffic whatever the data traffic density and pattern.

The *Optimised Link State Routing* (OLSR) protocol [JMC$^+$01] is an optimisation of the link state routing algorithm based on the multipoint relays technique. Multipoint relays (MPRs) reduce duplicate retransmissions of a broadcast packet in the same region during its diffusion. They are a selected subset in the neighbourhood of a node that covers each of its two-hops neighbours (only bi-directional links are considered). An example is given by Fig. 2.2.2. Neighbours of a node N that do not belong to its multipoint relays set read and process the broadcast packets emitted by N but do not retransmit them.

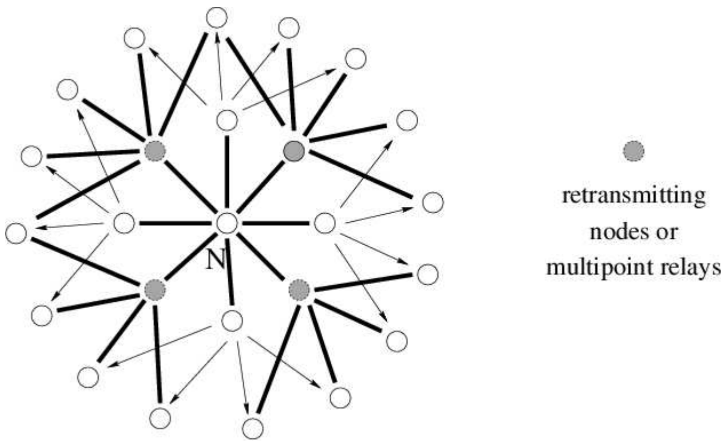


Figure 2.1: Multipoint relays (fig. copied from [JMC$^+$01])

Each node regularly emits hello messages containing its identifier and its set of neighbours. On this basis, each node knows its two-hops neighbourhood and compute independently a (small if possible) set of MPRs. If a neighbour is selected as a MPR, this status is indicated in further hello messages. Each node also regularly emit broadcast Topology Control messages that indicate the subset of its neighbours that has selected it as a MPR. The remaining neighbours are not included. The TC messages are flooded to the whole network, forwarded by successive MPR nodes.

The optimisation of the link-state routing algorithm is thus twofold. First, the flooding of the link state information necessitates the emission of less packets, because only MPR nodes forward them. Secondly, the TC messages only contain a subset of each nodes links. This (partial) topology information received at each node is sufficient for computing a path to every node that is optimal in number of hops.

The multipoint relay set of a node is recalculated either when a bi-directional link to a neighbour appears or breaks, and when a change in the two-hops bi-directional neighbourhood occurs. The routing table is also recomputed when a bi-directional neighbour is added or retrieved, and when a routing entry expires. Note however that no extra traffic is generated in response to link failures and additions. Sequence number are used for discriminating fresh from stale information, hence avoiding loops formation.

As all proactive protocols, the OLSR algorithm is well-suited for applications that do not tolerate much delay, and for a dense data traffic. The MPR flooding optimisation provides the best results in large and dense networks. OLSR has a good behaviour in large networks because, as some of the hierarchical protocols presented in next section, it is a neighbour-selection protocol. It thus belongs, unlike all routing protocols presented above, to the class of non-uniform protocols.

The *Fisheye State Routing* (FSR) protocol [PGC00] is also a proactive, link-state routing protocol with good scalability. The nodes periodically exchange their local link-state table with their neighbours only and use the notion of multi-level fisheye scope. The frequency used for sending a link-state entry depends on the distance it indicates for the destination. Entries for close destinations are more often emitted than entries for farther ones. This reduces the update overhead in large networks. Shortest paths are computed at each node on the local network graph. Long routes may be imprecise. However, they become progressively more accurate as the packets get closer to the destination.

Reactive protocols

Reactive protocols build and maintain a route only when a source has traffic to send through it. All reactive protocols are uniform. As stated above, this means that every node adopts the same behaviour when receiving a routing control message.

The *Ad-hoc On-Demand Distance Vector* (AODV) protocol [PR99] builds on the DSDV protocol described above. Each node maintains a node sequence number, that informs other nodes of the freshness of messages it injects in the network. The sequence number increases any time the node issues a new control packet (not if it forwards one). When a source needs a new route, it broadcasts a route request (RREQ) packet, uniquely described by the node address and a broadcast identifier. The broadcast identifier counter also increments at each new request emission. When a node receives a route request for the first time, it stores the following information: the source and destination address, the broadcast identifier, the source sequence number, the address of the neighbour from which it received the RREQ. This information allows the node to ignore further copy of the RREQ, and sets up, or updates, a reverse path to the source. An expiration time for the reverse path is also computed and recorded. During the RREQ propagation, reverse paths are thus built from every intermediate node to the source. These remain valid during a period of time sufficient for the RREQ to reach the destination and a route reply (RREP) to be unicast back to the source. When the destination receives the RREQ, it sends a route reply on the reverse path to the source. On its way back to the source, the forward route is set up. Intermediate nodes may also respond to a RREQ if they own a route to the destination associated with a sequence number greater or equal to the destination sequence number indicated in the RREQ. If a node receives multiple route replies for a given source-destination pair, it forwards th first reply to the source, and then only forwards replies for better routes, i.e. with less hops or higher destination sequence number. The source node begins to forward data as soon as it receives a RREP and can update its routing information later if it learns a better route. When data is forwarded by a node on a route, the associated expiration time is updated. Unused routes are regularly purged from the routing tables. Useful routes are said active. A node keeps track of any upstream neighbour on an active route. If an intermediate node detects a link breakage, for each active route having this link as next hop, it sends a special route reply, also called route error packet, to the associated active neighbours set. The route error packets follow the reverse paths to the interested sources. When a source receives a route error packet, or when its next hop link to the destination breaks, it emits a new route request. Link breakages may be detected by the explicit use of hello messages or by a mechanism of link layer acknowledgements, as only the breakage of links on active routes must be noticed. Because the route replies are forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

The *Dynamic Source Routing* (DSR) protocol presented in [JM96] is based on the concept of source routing and allows the use of unidirectional links. It is a topology-based on-demand routing protocol, while AODV is destination-based [Fee99]. Each node maintains a route cache, which is continuously updated as the node receives routing control messages. When a source has data to send, it consults its route cache. If there exists an unexpired route to the destination, data can be directly sent. Else, a route discovery process is initiated by the emission of a new route request. The RREQ contains the source and destination addresses, and a broadcast identifier. Intermediate nodes add their local address in the route record field. They silently discard duplicate route requests. No sequence number is required. Loops are easily detected because route request packets contain the identity of every node traversed. A reply is sent by the destination or by any intermediate node that holds a valid route to the destination. The responding node looks for a reverse route in its cache. If there is one, it is used for sending the RREP. Else, a route discovery for the source is initiated, with the RREP piggybacked in the new RREQ. When a link breaks, route error (RERR) packets are propagated to the sources using this link. When a route error (RERR) is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at this point. If needed, the sources re-initiates a source discovery process.

The *Temporally Ordered Routing Algorithm* [PC97] (TORA) maintains multiple routes for each demanded destination. When the destination receives a RREQ, it broadcasts an update message with a height field set to zero. Each node that receives the update message increments and records the height with the destination address. It then rebroadcast the update message with the new height value. Each pair of physical neighbours can then assign a direction to their common communication link by comparing the height they recorded for this destination. This process creates a directed acyclic graph (DAG) rooted at the destination. At stabilisation, each node knows at least one next-hop to destination. The updates must be broadcast reliably and ordered by a synchronised clock or logical time stamp in order to prevent long-lived loops. A node updates its height relative to a given destination only when its last downstream link to the destination breaks. The elevation of the node to a local maximum provokes the reversal of its upstream links and is propagated in the graph until a new DAG rooted at the destination stabilises. If the link break partitions the DAG, a new discovery process is initiated. The reaction to a link break is thus local and often quick. Route discoveries are much less often triggered than by AODV and DSR. Hence, TORA is proposed to operate in a highly dynamic environment. However, the applicability of the TORA's algorithm is limited by its reliance on synchronised clocks because every node must have a GPS or some other external time source. Moreover, although routes reconstructions are less often triggered than with other reactive protocols, these may take a longer time.

The *Associativity-Based Routing* (ABR) protocol [Toh97] has for objective to select stable routes. Each node periodically emits a hello message, or beacon. It also counts the number of beacons received from each of its neighbours since they are in each other communication range, the associativity ticks. This defines a new metric called the degree of association stability. The idea is that if two nodes are neighbours for a long time, their relative speed must be low. The route discovery procedure consists, as for AODV and DSR, of the flooding of RREQ messages. During their propagation onto the destination, the crossed nodes addresses and the associativity ticks assigned to the traversed links are included in the packet. The destination is then able to select the best route by examining the associativity tics along each of the paths. It sends a RREP back to the source along this path. The route maintenance procedure is quite complex. Depending on the position of the link in each route for which it was used, closer to the source or to the destination, the route repair may be local, consist of a partial or of a full new route discovery.

The route discovery procedure of the *Signal Stability-Based Adaptive Routing* (SSA) protocol [DRWT97] is also based on a broadcast route discovery and unicast route reply back to the source. As for ABR, its main objective is to prefer longer-lived routes. For this purpose, each node periodically emits hello messages. It also measures the signal strength of each beacon received and, on this basis, classifies its neighbour links as weak or strong channels. Route requests received on weak channels are ignored. Hence, route replies only follow strong channels on their way back to the source. If no reply is received within a specific timeout period, the source may re-initiates a route discovery, indicating in the RREQ that it will also accept routes through weak channels. A link failure is notified to any concerned source, which re-initiate a route discovery procedure.

Inside the reactive class, each protocol shows up some advantages and drawbacks. On one hand, AODV only provides routes made of symmetric links, while DSR can use asymmetric links. The route cache may provide a new needed route without discovery procedure. On the other hand, DSR packets are larger and its memory overhead may be slightly greater. TORA is suitable for large, highly dynamic, mobile environments with high nodes density. As DSR, it moreover provides multiple routes. However, temporary oscillations may occur and convergence relies on synchronised clocks. The limitation of ABR and SSA comes mainly from a periodic beaconing which may result in additional energy consumption. Their advantage is a bandwidth economy resulting from less route breakages. For a detailed comparison of these reactive protocols, the reader can refer to [BMJ+98, RT99].

In comparison with the proactive uniform protocols, reactive protocols are generally more efficient. They minimise control overhead and power consumption because routes are only established when required. The price to pay is the initial search latency, that may degrade the performance of interactive applications and does not fit well to the standard TCP procedure for connexion setup. In many protocols, the quality of a path cannot be controlled by the source and may evolve transparently for the source. These features make the proactive uniform class the best solution for small networks with a low mobility level. For larger and variable network topologies, reactive protocols work better. A drawback of reactive protocols is that the route discovery and recovery process is potentially both expensive and unpredictable. The signalling traffic grows with increasing mobility, network size and number of data flows. As mobility increases, the pre-discovered routes may break down, requiring repeated route discoveries. Source routing makes DSR less adequate in network with a large diameter because the header of unicast packets must describe the full path from source to destination. For all reactive protocols, the control traffic needed for discovering long routes is bigger than for short ones. At heavy traffic (directed to many destinations), more sources trigger a route search process. However, reactive protocols work well in small to medium size networks in which the mobiles move at moderate speed with respect to packet transmission latency. They also scale well for large networks when the traffic is light and mobility low.

Hierarchical protocols

On very large networks, flat routing protocols overload the network links, and deplete the nodes processing and memory capabilities.

Oppositely, with hierarchical protocols, the network is partitioned in several groups. Following the protocol considered, these groups are called clusters, trees or zones. The groups can be further explicitly organised in a hierarchy, while the others are implicitly hierarchical.

Each node in a group is able to forward a packet to another member of the same group. Only a subset of nodes in each group is responsible for forwarding packets outside the group. The routes for destinations located inside and outside the group can be computed with different algorithms.

Non-uniform, or hierarchical, protocols define different roles in route computation and/or forwarding process. The routing complexity is limited by reducing the number of nodes participating in a route calculation. The routing table and routing packets are also reduced because they include only a part of the network.

They can be divided into two categories: Clustering and neighbour-selection protocols.

Clustering is a conventional method used in hierarchical protocols. It consists of grouping together nodes that are geographically close to each other. Nodes that belong to overlapping clusters are gateways. In each cluster, a non-gateway node is elected as clusterhead. Only clusterheads and gateways propagate routing control messages. The routing information is moreover aggregated, i.e. only paths between clusterheads are announced and recorded. The various clustering protocols differ on the criteria used for organising and maintaining the clusters, and on the inter- and intra-cluster routing strategy.

The *Clusterhead-Gateway Switch Routing* (CGSR) protocol [CWLG97] uses the *Least Clusterhead Change* algorithm for grouping nodes around stable clusterheads. In a cluster, the clusterhead is a physical neighbour of every nodes. DSDV is used as the underlying routing scheme. Clusterheads and gateways maintain routes for every clusterhead. Other nodes only need a cluster member table which indicates the clusterhead associated with any destination.

The *Core Extraction Distributed Ad Hoc Routing* (CEDAR) protocol [SSB99] estimates a dominating set of the network. By definition, each node is within the communication range of at least one dominator node, called in this context a core node. Nodes use a reactive source routing strategy. A source forwards a route request to its dominator. The core nodes encapsulate the request in unicast packets and tunnel them to each of their neighbouring core nodes. This mechanism is called *core broadcast.* It consumes far less bandwidth than the legacy flooding procedure used by flat reactive protocols.

The *Cluster Based Routing Protocol* [Jia99] (CBRP) emphasises support for uni-directional links. Inter-cluster connectivity may be obtained via a pair of uni-directional links. Every member node of a cluster shares a bi-directional link to its clusterhead. A reactive source routing strategy is used. Only gateways forward route requests to external clusterhead nodes. When a request reaches its destination, it contains a loose source route specifying a sequence of clusters. The route reply traverses the corresponding clusterheads. These fill the reply with an optimised route portion, based on their local cluster view, that does not necessarily passes through them. The advantage over the previously described clustering techniques is that clusterheads are not data bottlenecks.

The *Hierarchical State Routing* (HSR) protocol [ICP$^+$99] is a multi-level clustering, link state routing protocol. The clustering scheme is applied recursively for organising the topology into a logical hierarchy. Only clusterheads at the lowest level of the hierarchy become members of the next higher level. The new members in turn organise themselves in clusters and so on. The clusterhead summaries link state information within its cluster and propagates it to the neighbour clusterheads (via the gateways). Each node is assigned a *Hierarchical ID* (HID) based on MAC addresses. The HID is sufficient to deliver a packet to its destination from anywhere in the network using the aggregated routing tables. The clustering hierarchy is a physical organisation linked to geographical proximity. Mobile nodes are further logically grouped in subnets, based on a logical functional affinity between nodes. Members of a given subnet share a common IP prefix, and one of them is elected as a home agent. The home agent keeps track of any mobile node of its subnet, by being informed of their HIDs. Each home agent advertises its own HID to the nodes belonging to the top level of the hierarchy. A packet can be directly sent if the source and destination nodes belong to the same cluster at the lowest level. Else, it climbs the hierarchy, and is forwarded to the home agent of the destination subnet. The home agent maps the destination IP address to a HID and the packet can be forwarded to the destination node. This method is similar to mobile IP, except that home agents also move.

In **neighbour-selection** protocols, each node selects independently a subset of its neighbours for each specified behaviour.[2] There is no negotiation process in which nodes must achieve consensus. The node's selection is only affected by local topological changes. The process of a routing control message or data packet by a given node depends on the role that its sending neighbour assigns to it. A given node may be selected for different roles by different subset of neighbours.

The *Zone Routing Protocol* (ZRP) [Haa97] applies a hybrid routing strat-

[2]The OLSR protocol presented above is also a neighbour-selection protocol, but it is not hierarchical.

egy, combining proactive and reactive behaviours. Each node proactively maintains routes to all other nodes within a given number of hops. If it must send a packet to a node that does not belong to its zone, it uses a reactive routing strategy. The emitted route request must only reach the boundary of the zone defined by the destination node. Effectively, the routing table of every node on this boundary includes an entry for the destination. For large routing zones, the protocol behaves like a proactive protocol and for small zones, it behaves like a reactive protocol. A tradeoff must be found in each network in order to combine the advantages of the two strategies. No location management scheme is required.

The *Landmark Ad Hoc Routing* (LANMAR) protocol [PGH00] is designed for an ad hoc network that exhibits group mobility. Namely, where one can identify logical subnets in which the members are likely to move as a group and to remain close to each other. Members of a given subnet share a common IP prefix, or *Group ID*, and one of them is elected as landmark. A proactive routing protocol propagates the routing information about all landmarks in the entire network. Each node thus has an entry in its routing table for each landmark. It moreover maintains detailed topology information about nodes within a local scope that covers all the members of the same subnet. When a packet must be sent to a destination located outside of the local scope, its Group ID allows to forward it to the corresponding landmark. When the packet arrives within the scope of the destination, it is forwarded on the shortest path to it, maybe deviating from its original route onto the landmark.

All hierarchical routing protocols use smaller routing tables than uniform proactive protocols. The routing overhead is greatly reduced because the routing control messages contain a shorter list of destinations. Clustering may also decrease the number of routing packets emitted. However, in the face of mobility, explicit cluster based hierarchical protocols will induce additional overhead in order to maintain the hierarchical structure. Moreover, some of them require a membership management scheme. If nodes are stationary, hierarchical routing may scale to very large networks. However, node mobility is a critical point. Frequent rearrangements of clusters may introduce excessive overhead that may nullify the clustering benefits.

Implicitly hierarchical protocols ZRP and LANMAR do not suffer from this drawback in a mobile context. However, LANMAR relies on the group mobility assumption. On the other hand, ZRP has a limited scalability. When the network grows, the local scope can be enlarged but this raises the routing and storage overhead. If the scope remains constant, ZRP's behaviour becomes similar to on-demand routing with unpredictable, potentially large communication overhead.

A coarse performance comparison of the ad hoc routing protocol families

We summarise on Fig. 2.2 the applicability conditions, with respect to the network size and the mobility level, of the ad hoc routing protocols families described above.

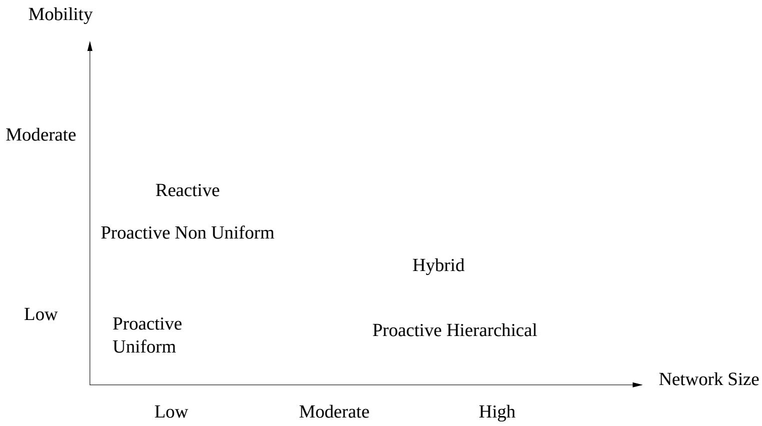


Figure 2.2: Applicability conditions of ad hoc flat unicast routing protocols

Proactive protocols are only suitable for small networks with a low degree of mobility. However, they present some advantages that make them the key solution in this particular case. For large networks, the hierarchical approach is the best. However, mobility impairs the maintenance of a hierarchical structure. Oppositely, in a highly mobile environment, the best performance are obtained with reactive protocols. Their performance however degrades with a high traffic density and in large networks. Unfortunately, the reactive and hierarchical routing philosophy are not compatible. Effectively, any hierarchy must be organised before a node requires a route.

In this dissertation, we use two well-known, and efficient, representatives of the reactive and proactive family, namely AODV and OLSR.

2.3 Overlays

An overlay network is a network built on top of another network. It generally spans a subset of the network systems, sharing an internal addressing space. It defines a virtual topology between these nodes, providing a simple network view, without the details of the underlying topology. The nodes can be connected by direct or virtual links. A virtual link can cover multiple physical hops on the underlying network.

Overlay solutions present several advantages over protocols installed on every router. They are:

- Incrementally deployable.
- Adaptable. Paths can be optimised with metrics that matter to the application.
- Robust. The overlay topology is controllable. With a sufficient number of nodes, it can be maintained so as to provide multiple disjoint paths between any pair of overlay nodes.
- Customisable. An overlay node may be a multi-purpose computer, with specialised equipment. It could for example offer a storage capability, that legacy routers do not provide.
- Standard. An overlay network can be built on the least common denominator network service of the substrate network.

They however also present some drawbacks.

- Management complexity. The overlay's managers are generally far from the machines. Their physical maintenance must be minimised, and tractable for untrained persons. The overlay's scalability is also an issue.
- An overlay must face real world's conditions. It must for example adapts to NATs and firewalls in the Internet, and to transient connectivity on ad hoc networks.
- Inefficiency. The efficiency of an largely deployed overlay service can approach the efficiency of router based services, but an overlay service cannot be as efficient as code running in every router. However, if the overlay is small, the absolute inefficiency is small as well.
- Information loss. The overlay hides the topology of its substrate. If the application needs the cost of the overlay links, these must be estimated, for example with probes.

2.3.1 Main applications on Internet

Over Internet, overlays have been used, or at least proposed, for:

1. Application-level multicast,
2. Peer-to-peer networking,
3. Enhancing the routing service available on the underlying network. For example:
    - Providing quality-of-service

  - Securing the communication
  - Building a resilient network

4. Testing new routing solutions.

Application-level multicast

Many services based on overlay networks provide some form of content distribution, mainly implemented by application-level multicast and peer-to-peer systems.

The potential advantages of performing content distribution at a higher layer than IP are:

- Incremental and ease of deployment. Application-level multicast protocols must not be installed on every routers of a network. This feature moreover solves many scalability problems encountered by IP multicast, at least for small groups.

- Ease of management. No management on the core routers is required. The management procedure can be applied on a multicast group only, or on the content distribution infrastructure, rather than on the whole network.

- Support for higher layer functionalities. Error, flow and congestion control and security are more easily deployed on end-systems than in the core network.

- Application-driven distribution. The distribution protocol, installed on the same nodes as their user application, or on dedicated servers, can be customised in regards with the heterogeneity of the network and the application characteristics.

However, the key concern with application-level multicast is the performance penalty. Unlike IP multicast, duplicate packets may be generated on the physical links. The users also observe larger end-to-end delays.

Table 2.1 gives the main features of some application-level multicast systems.

The *Narada* [hCRZ02] protocol self-organises a small or sparse group of multicast users into an overlay structure. It consists of a distributed 2-step process. A mesh is first built, which is a richer connected graph than the distribution structures that are next built. These are reverse shortest path trees (SPT) over the mesh. There are several incentives for building content distribution trees over a common mesh:

- The group management functions are abstracted out and handled at the mesh rather than replicated across multiple (per-source) trees.

Table 2.1: Classification of application-level multicast systems

Protocol	Content type	Group size	Caching	Adaptive coding	Infrastructure type	Overlay topology type
Narada	Real-time	Small	No	No	End users	Tree over a dynamic mesh
Nice	Real-time	Large	No	No	End users	Tree over a dynamic mesh
FastForward	Real-time	Large	No	Yes	Server-based	Dynamic tree over a static mesh
Scattercast	Real-time	Large	No	Yes	Server-based	Tree over a dynamic mesh
RMX	Real-time	Large	No	Yes	Server-based	Tree over a dynamic mesh
Overcast	Full-fidelity	Large	Yes	No	End users	Dynamic tree
Yoid	Any	Large	Yes	No	Hybrid	Independent dynamic tree and mesh

- A loop avoidance procedure is not necessary for a mesh. The heuristics for repairing partitions and optimisation are simpler than for a tree.
- There exist standard procedures for building a shortest-path tree over a mesh
- A mesh is more resilient to the failure of members.

However, there is no control over the resulting spanning tree for a given mesh. Building a good mesh is thus important, so that good quality trees may be produced. Narada overlay neighbours are first chosen randomly. The quality of the mesh is then incrementally improved. The procedure is based on the gain in latencies. Nodes running Narada determine latencies to other end systems by probing them in a controlled fashion. New overlay links may be added depending on the perceived gain in latency of doing so. The utility of existing links is also continuously monitored. Links providing a too low gain in latency are dropped.

The application-layer multicast protocol developed in the NICE project, itself called *NICE* [BBK02], defines a multi-layer hierarchy of fully meshed clusters. It has been designed for low-bandwidth real-time data streaming applications for large groups. A new multicast member contacts a rendezvous point at the highest layer of the hierarchy. The rendezvous points provides the identifier to every cluster leader at each level. The joining node determines its distance to each of them, through RTT measurements. It then enters, at each level, the closest cluster. The number of nodes per cluster is bounded. Hence, if a cluster becomes too large, it is split. If it becomes too small, because some nodes leave, it is merged with another cluster. Source-specific data distribution trees are implicitly defined by the hierarchical cluster structure. Every member forwards data to all members of its cluster at each level, except for the clusters that the previous sender of the message belongs to.

FastForward is a server-based distribution system for real-time content over a large group. It adapts the content to the available bandwidth. The overlay topology is a mesh that is statically configured so as to avoid single points of failure. The nodes participating to the diffusion of data are picked dynamically on the topology. Oppositely, tree-based overlays have single points of failure, but their topology is adapted in case of overlay link breakage.

Scattercast [Cha03] is a protocol for Internet broadcasting, defined as the simultaneous distribution of live content streams to a large audience. Unlike Narada, which connects end users (or their edge routers), the Scattercast architecture relies on a collection of strategically placed network agents, called Scattercast proxies. These collaborate with each other in order to form an overlay network composed of unicast interconnections between locally-scoped multicast regions. Such an infrastructure is called hybrid, as opposed to the infrastructures composed only of unicast tunnels, such as Narada. The Scattercast solution provides an overlay building protocol and a programming model for customising the infrastructure at the transport level. Its objectives are to build an efficient infrastructure for large-scale broadcasting, to offer the ability of adapting the infrastructure to suit the requirements of a large range of applications, and the ease of deployment. The overlay construction protocol, Gossamer, is based on Narada but is adapted for large groups. New members send join messages to a set of rendezvous points. For the mesh gradual improvement and tree maintenance, a restricted distance vector routing protocol is run at the application layer by the overlay members. The overlay node out- and in-degree are both bounded. That is, respectively, the number of overlay neighbours an overlay node will search for is limited, as well as the number of neighbours it will accept. The distance metric used is the unicast latency.

RMX [CMB00] is based on the Scattercast hybrid architecture. It provides semantically reliable distribution of content. For this purpose, it adapts the coding of the data to the heterogeneous capabilities and network connections of the clients.

Overcast [JGJ$^+$00] implements reliable single-source multicast for large groups (tens to hundreds of nodes). It builds distribution trees that adapt to changing network conditions without requiring router support. It permits the archival of content sent to multicast groups, by exploiting the permanent storage capability of the participating nodes. Overcast is designed for the delivery of full-fidelity content, rather than for real time streams. It suits for example the distribution of software, even over low bandwidth links.

The *Yoid* paper [Fra00] points out the low usage of native multicast, and the numerous application-level, sometimes proprietary, solutions used on the Internet for content distribution. It proposes a generic overlay solution with tools for content distribution, useful for many different applications. The possibility of storing content at intermediate nodes is a salient feature, shared with Overcast. It is composed of several protocols. One of them is responsible for the maintenance of a tree, on which the data will be forwarded. An independent mesh is also maintained for management tasks. Among other functionalities, it allows to recover fast when the tree becomes partitioned.

Peer-to-peer systems

This short overview of content distribution peer-to-peer (P2P) systems for Internet is mainly extracted from [ATS04]. Peer-to-peer systems are defined in this survey as distributed systems consisting of interconnected nodes able to self-organise into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralised server or authority.

The applications of P2P systems are various:

- Communication and collaboration: chat and instant messaging applications
- Distributed computation
- Internet Service Support, such as multicast systems, security applications
- Distributed Databases
- Content distribution

Most of the P2P systems fall into the last category. Basically, a content distribution system is employed for the publishing, lookup and retrieval of files by the members of the system, called peers. It is composed of the peers (nodes) and connections (edges) between them. It is built on top of the underlying physical network (typically IP) and is thus an overlay.

First content distribution systems were centralised, i.e. a central server stored meta-data describing the files stored in the peers, and provided to the peers looking for data the corresponding locations. The central server is a single point of failure and these early solutions, as for example Napster and Publius [WRC00], do not scale.

In their purest form, P2P systems are totally decentralised. All peers perform the same task. They act as server as well as client and there is no central coordination of their activities. The core functions of P2P content distribution systems are the overlay construction and routing of file queries.

In **unstructured** P2P systems, the placement of files is completely unrelated to the overlay topology. Content must thus be located. The simplest location mechanisms is to flood queries. This is not bandwidth-efficient and the lookup may take a long time. Hence, some unstructured systems, as for example *Gnutella* [Gnu] and *Kazaa* [Kaz], assign a more important role to some nodes, which act as local central indexes for files stored in near peers. Other mechanisms that accelerate the location of content are the use of parallel random walks, or routing indices which indicate to nodes the direction with a higher probability of hit.

In **structured** P2P systems, the overlay topology is tightly controlled and the placement of files is precisely defined. The peers organise themselves as a distributed routing table. They forward queries to the node where the corresponding file is stored (or where a pointer to it is stored). They provide a scalable solution for exact-match queries, that is, when the user is able to associate a unique identifier to the data it needs (not keywords). The drawback of structured systems is that the maintenance of the the structured topology is difficult when nodes enter and leave the system frequently.

In a *Content-Addressable Network* (CAN) [RFHK01], the keys are mapped onto a point in a virtual d-dimensional Cartesian coordinate space. The overlay topology is organised such that every overlay node is able to forward a query to the next overlay node on the straight line path to its destination through this space. The system is demonstrated as being scalable.

At each *Pastry* [RD01] routing step, the current node forwards a message to a neighbour whose identifier shares with the message key a longer prefix than its own identifier.

Chord [SMDK01] and *Tapestry* [ZKJ01] are two other well-known structured, scalable, P2P systems.

There also exist **loosely structured** P2P systems, where the location of files is not precisely specified but where content is placed such as to provide some routing hints. For example, *Kademlia* [MM02] and *Oceanstore* [KBC+02]. Note that all structured and loosely structured P2P systems are fully decentralised.

The different systems also differ by the advanced functionalities they offer, and by the way they implement them. They provide to the P2P systems different levels of security, scalability, performance, fairness and intelligence in resource management.

Improving the basic routing service

Overlays have emerged in the Internet as an alternative for introducing new functionalities difficult to deploy in the underlying IP infrastructure, or that require information that is hard to obtain at the IP level.

They can afford for example quality of service or fault-tolerant routing.

Quality-of-Service (QoS) Overlays empower third-party entities other than traditional ISPs to offer enhanced communication services to clients [SSBK03]. These entities can be a service provider or an organisation that uses overlays to provide enhanced services in its Virtual Private Network. In the first case, the service provider buys access links to ISPs, and install servers organised as overlays. By controlling the resources on the virtual links and controlling the traffic flowing on them, it is able to offer enhanced services to its own customers. This Service Overlay Network (SON) framework is the basis for the OverQos and QRON architectures.

In the *OverQos* architecture [SSBK03], the technical key introduced is an abstraction of the virtual links between overlay nodes. The abstraction allows to control the error rate, potentially at the expense of a reduction in bandwidth.

The *QRON* architecture [LM04] is proposed as a general unified framework for application-specific overlays. Authors argue that many functionalities are common to many overlay applications. These are the topology discovery, routing path selection, fault detection and tolerance, overlay link performance estimation and resource allocation. The authors give a proof-of-concept through simulations, showing that their routing algorithms can effectively find and provide QoS-satisfied overlay paths and can balance the overlay traffic burden among overlay nodes and overlay links.

Fault-tolerant routing The main objective of a *Resilient Overlay Network* (RON) is to improve the reliability of Internet communications for small communities (3 to 50 nodes). The underlay is expected to repair faults, while the robust RON overlay is able to route packets around faults quickly. The RON design allows the use of application-defined quality metrics and routing decisions. RON nodes detect and recover from path outages and degraded performance within several seconds (against several minutes for default Internet mechanisms). The default path between each pair of RON nodes is monitored. The estimation of its quality is stored and disseminated on the overlay. Link-state routing is then applied. The source of a flow decides if its packets should follow the default Internet path or if they should traverse a given RON node before reaching their destination.

Testing new networking solutions

Physical testbeds and overlays are the two ways in which researchers currently experiment with new architectures. In addition, overlays are also seen as a deployment path.

Mbone, 6bone, X-bone and Abone Overlays are also useful for implementing experimental networks, and may facilitate the migration from a networking solution to the next one.

The *MBone*, that started in 1992, was an experimental backbone for the experimentation of IP multicast traffic across the Internet. It connected IP Multicast capable networks (often universities and research institutions).

The *6bone* was a worldwide informal collaborative project. The 6bone started as a virtual network (using IPv6 over IPv4 tunnelling/encapsulation) operating over the IPv4-based Internet to support IPv6 transport, and has been slowly migrating to native links for IPv6 transport [6Bo].

The MBone and 6bone required manual configuration and management, both to establish connectivity and to ensure efficient resource utilisation. Oppositely, the *X-Bone* system offers a graphical interface for automated overlay deployment and management without requiring out-of-band (i.e. telephone or e-mail) communication between human managers. With the X-Bone tool, the overlay manager only selects participating nodes and a type of topology (e.g. ring in-order or full-mesh). The system automatically assigns addresses, determines whether and where virtual links, implemented as IP-in-IP tunnels [Sim95], are required, configures them and adds routes. There may be several overlays on a physical network, which can share both link and node resources. The system limits the link stress and manages inter-overlays resource contention.

The X-Bone exploited advances from the active networks [TW96] research area. Moreover, it was itself used for testing the active networks technology on the Internet, through the deployment of an overlay called the *A-Bone* [Ber00]. An active router or switch is programmable. It may implement several independent forwarding behaviours, interpret and even modify the payload of a packet. Hence, it is able to perform customised operations on packets flowing through it [TSS+97].

PlanetLab The global research overlay called *PlanetLab* [CCR+03] supports the development of new large-scale and geographically distributed network services. It has passed the 900 nodes mark during summer 2008. It aims at facilitating the testing of disruptive networking technologies and their deployment. Hence, it supports both researchers and clients. One key design principle is that the management of the overlay is partitioned in a set of independent sub-services [PACR03]. These are for example the discovery of overlay nodes, their monitoring, or the management of users account and credentials. They may evolve over time. In other words, even the management structure is engineered for innovation.

Virtualisation The design guidelines of PlanetLab came from the fact that a deep modification of the Internet architecture is very difficult. The current IP technology has been so successful and widely accepted that replacing it or introducing new capabilities would not be easy nor does motivate industrial players [Cou01]. Introducing a disruptive technology would not only require changes in hardware or software, but also a global agreement of many ISPs. As a solution, it is proposed in [APST05] to construct a virtual testbed on PlanetLab, composed of a multiplexed overlay substrate and a general client-proxy mechanism. The multiplexed overlay substrate hides the physical network and supports multiple simultaneous architectures. It allows researchers to select a set of virtual nodes and links that encounters the resource requirements of their architectural proposition, and to test it on a large-scale testbed and under live traffic as if it ran over a dedicated network. The client-proxy mechanism allows any client to opt-in to a particular experiment, and avoids to rely on the IP protocol.

The *VINI* (VIrtual Networking Infrastructure) environment implements these ideas. Multiple experiments can run on a shared physical infrastructure, the switch-over from one virtual network to another one is possible. Its implementation over PlanetLab nodes, PL-VINI, offers a credible path for deployment. Hence, VINI allows the network designers to evaluate their protocols or architectures in realistic conditions. It moreover adds control over their experiments. Network events, such as for example a link failure, can be injected in any virtual network created.

In [FGR07], the sharing of a physical infrastructure by concurrent virtual networks is considered as an architecture in itself, rather than means for testing new architectures. Authors suggest that the future network architecture is a platform that allows all networking functions to evolve. The *Cabo* framework partitions the functionalities assured by current ISPs and proposes to assign them to two distinct entities:

1. The infrastructure providers, that manages the physical infrastructure, and

2. The service providers, that deploy network protocols and offer end-to-end services.

The expected benefits of Cabo include easy deployment of end-to-end network services, the ability to run custom routing protocols, and better accountability.

2.3.2 Overlay topologies on top of the Internet

In every overlay application, building and maintaining the overlay topology is a basic functionality.

All studies summarised in this section, are dedicated to the design of overlay topologies. They all show that collecting topological information about the underlay can be very beneficial to the construction of efficient overlay topologies.

They also follow similar objectives. The overlay connectivity and its scalability are fundamental requirements. Above these, we can divide the metrics used for estimating the quality of an overlay into two categories. The first one is relative to its resilience. It can be measured for example by the failure recovery ratio or independence of overlay paths. The second one is linked to classical quality of service metrics. These can be expressed for example by the overlay path stretch, average hop count distance, latency stretch, average overlay latency, recovery path hop penalty, overlay bandwidth or overlay link stress.

The topology construction can be directly integrated in the applications, and optimised for them. A typical example is the various tree maintenances for multicast [KF02].

However, papers that discuss the quality of an overlay topology, rather propose a generic solution than focus on a specific overlay application. Many of them propose an overlay construction architecture or protocol while some study compare various overlay topologies without describing their practical implementation.

Most of them consider the problem of building edges between a given set of overlay nodes. The placement of these overlay nodes may be an additional degree of freedom, subject to optimisation.

Topology-aware protocols

The following three papers show the advantages of grouping nodes that are geographically close.

In [RHKS02], a binning scheme is applied to the construction of overlay networks.[3] The nodes partition themselves into bins such that nodes that fall within a given bin are relatively close to one another in terms of network latency. The binning scheme requires a set of well-known landmarks spread across the Internet. Each node measures its round-trip-time (RTT) with every landmark. The range of possible latency values is divided into discrete levels. Two nodes belong to the same bin if and only if they assign the same level to every landmark and obtain the same sequence of landmarks when sorting them by increasing RTT. Examples of applications are given both in the case of structured and unstructured overlay networks. The structured overlay construction protocol considered is CAN [RFHK01]. In its original version, CAN allocates nodes to zones at random. The protocol is re-evaluated after mapping of the bins and zones. For the construction of unstructured overlay networks, the following general problem is considered. Given a set of n nodes on the Internet, have each node pick any k neighbour nodes from this set, so that the average routing latency on the resultant overlay is low (assuming shortest path routing). The *ShortLong* heuristic is defined, where each node picks the $k/2$ neighbours closest to itself and then picks another $k/2$ nodes at random. As this approach is not scalable,

[3]It is also applied to the selection of servers, which is not related to the work presented in this dissertation

because a node should know the distance that separates it to every other node, it is approximated by the *BinShortLong* heuristic. In this case, every node picks $k/2$ neighbours in its bin and then picks the remaining $k/2$ neighbours at random. In all examples, the overlay latency stretch is significantly reduced in comparison to the one of random overlay topologies. This study thus shows that even rather coarse-grained topological information can significantly improve the application performance.

The overlay construction method presented in [ZZZ+04], *mOverlay*, has for objective to minimise the average overlay neighbour distance. The overlay built is a two-level hierarchical network. The top level consists of groups and the bottom level consists of hosts within groups. A group is composed of hosts that are close to each other. The overlay is built so as to establish most links between hosts within the same group and only one or two links between each pair of groups. When a new host arrives, it uses a recursive locating method to join the nearest group or forms its own group according to a grouping criterion based on distance measurements. This approach employs dynamic landmarks, and is expected to show higher scalability and robustness than approaches relying on static landmarks. It however requires a rendezvous point for bootstrapping. Authors envisage media streaming, media distribution and application-level multicast as potential applications.

The study presented in [Vie04], is conducted in the Service Overlay Network (SON) framework presented above. The SON is composed of a set of (service) provider nodes and a set of subscribers, called endsystems. All nodes are connected through ISPs that support bandwidth reservations. The provider nodes offer end-to-end QoS guarantees to the endsystems. The paper addresses the problem of finding the optimal set of edges between a given set of provider nodes and to assign each end system to one provider node. This topology design problem is expressed as an optimisation problem. Two provider nodes can establish a link if they are both connected to the same ISP. Likewise, an endsystem can access a given provider node if both are connected to the same ISP. Costs are classified in access and transport cost, respectively relative to the connection of an end-system to a provider node and to the connection of two provider nodes. These are assumed to be proportional to the amount of reserved bandwidth and normalised. The objective is to build a connected overlay with minimal total cost, composed by all access and transport costs, weighted by the amount of bandwidth reserved on the links. In the first step, only the links between pairs of provider nodes are considered and the overlay topology is built in order to minimise the total transport cost. In the second step, the endsystems are connected to provider nodes. In the final step, provider nodes that are not required for serving the set of endsystems are eliminated. Several heuristics are explored. In particular, authors show that the complexity of the problem can be reduced by grouping geographically close endsystems into clusters and by assigning all endsystems in a cluster to the same provider node.

Generic architectures

Saxons [Che04] and PLUTO [NP06] are two substrate-aware architecture for the construction of generic overlays. The former is based on end-to-end measurements while the second uses AS-level Internet topology and routing information from nearby BGP routers.

Saxons stands for Substrate-Aware Connectivity Support for Overlay Network Services, a layer dedicated to the management of an overlay structure. The software proposed is generic. Services can directly utilise the Saxons structure for overlay communication. It can also benefit unicast or multicast path selection services, such as [ABKM01, hCRZ02, JGJ+00], by providing them a small link selection base. As services with different communication patterns place different quality demand on the overlay structure construction, Saxons is customisable. The link density, for example, can be configured by setting a minimal and maximal overlay nodes degree. The Saxon nodes progressively learn all overlay nodes identifiers. They first contact a small set of bootstrap nodes, reachable through a DNS request. These provide to any joining node a random subset of the overlay members. The overlay nodes measure overlay links latency and bandwidth respectively with ICMP and UDP traffic. They estimate the distance to other, random, overlay nodes for finding nearby hosts. A routine is run periodically to adjust active links for potentially better structure quality. The quality routine measures the latency to a few randomly selected hosts and replace the longest existing active link if new hosts are closer. As a consequence, Saxons creates mesh structures with large hop-count distances. In order to avoid link oscillations, a link adjustment occurs only when the new link is shorter or wider than the existing overlay link for more than a specified threshold. The Saxons paper introduces the *ShortWide* approach for quality maintenance. This builds half of the active links to closest overlay hosts, and half active links with the widest bandwidth. The simulation results show that the ShortWide structure management policy outperforms other policies in terms of overlay path bandwidth while achieving competitive performance in terms of overlay path latency and hop-count distance. The *AllShort* policy, where the overlay links are built between close overlay nodes, produces overlay structures with high hop-count distances. The metrics used in the evaluation are the hop-count distance and the overlay bandwidth. The overlay hop-count distance is defined as the average hop-count distance along the overlay structure for each pair of overlay nodes The overlay bandwidth is estimated in terms of the bandwidth along the shortest overlay path, and along the widest bandwidth overlay path. Connectivity, stability and scalability are additional design objectives. For connectivity, Saxons provides partition detection and repair support such that upper-level services do not need to worry about the overlay partitioning. Connectivity messages are periodically broadcasted from a core node. If an overlay node does not hear them, it establishes a new overlay link with a random node until the core messages are heard again. In order to avoid partitions due to the link adjustment procedure, the upstream link to the core is always preserved. Concerning stability, Saxons hides node joins and leaves to the upper-layer by maintaining the overlay structure with infrequent link adjustments. Scalability is achieved by controlling the system management overhead when the overlay scales up.

Many overlay applications, like Saxons, use frequent probes, such as ping and traceroute, for learning something about the underlay topology. This information is then exploited for improving the overlay topology, with the final goal of getting better performance at the user level. Probes are also used for estimating the overlay link properties, like bandwidth and loss. This strategy is not likely to scale. In particular, it does not allow for many overlays over the same physical substrate. Hence, in [NP06], authors propose to implement overlays on top of a shared set of topology discovery services. The library of routing services is expected to provide useful foundation for a variety of overlay networks. It must be implemented from a set of primitives exported by a lower layer, called the topology probing layer. The argument for this layered architecture is to reduce the probing costs. The lower layer expose coarse-grain static information at large scale, while the upper layers perform more frequent probes over an increasingly narrow set of nodes. A non-exhaustive list of services is suggested. A possible implementation of the services is also given, as well as an implementation of the primitives used. Suggested services are finding the nearest neighbours of a node, finding disjoint paths between two nodes, and building a routing mesh. The routing mesh should be fully representative of the Internet, but with far fewer edges than a full mesh. It should be obtained by retaining only overlay links that are path independent. In general, selecting virtual links that share as few underlying physical links as possible both reduces redundant traffic on the physical links and eliminates fate sharing in the case of link failure. The above architecture has been implemented and evaluated over PlanetLab nodes under the name *PLUTO* (PlanetLab Underlay Topology services for Overlay networks). PLUTO builds a topology-aware mesh to be used by routing overlays. The mesh eliminates virtual links that contain duplicate physical segments in the underlying network, through a conservative link pruning heuristic. The pruning algorithm uses only passive measurements and seldom-changing topology information such as AS-level topology and geographical information. It does not itself add to monitoring overhead. The evaluation shows that the routing overhead is reduced by a factor of two, without negative impact on the route selection. Additional analysis shows that constructing a sparser routing mesh on the topology-aware routing mesh – rather than directly on the Internet – itself benefits from having the reduced number of duplicate physical segment in the underlying network, which improves the resilience of the resulting routing mesh.

Comparison studies of overlay topologies

A different approach to the study of generic overlays, adopted in [CFSK04] and [LM07], consists of comparing several overlay service topologies, without considering their implementation.

In [CFSK04], the overlay formation is modelled as a non-cooperative game. Each node chooses its overlay neighbours so as to maximise its benefits. The game starts from a connected random graph, and in each round, each player changes its link configuration to minimise its cost. The cost of one node is composed of the linking costs to all its neighbours and of the cost of the shortest path to every overlay node. In order to avoid the disconnection of the graph, the cost assigned to an unreachable overlay node is very high. On one hand, a node establishes links with neighbours in order to profit from low cost paths to other nodes and on the other hand it tries to minimise its linking costs. The game stops when a Nash equilibrium [Nas51] has been reached . By definition, this occurs when no player can benefit by changing its strategy, while the other players keep their strategies unchanged. The authors discuss the characteristics of the overlay obtained at the end of the game. Metrics studied are the stretch, resilience to failures and attacks and node degree distribution. The tolerance to a failure or attack is measured by the percentage of remaining connected node pairs. A failure is modelled by the removal of randomly selected nodes. An attack is modelled by the removal of nodes in decreasing order of degree, starting from the node with maximal degree. Three models for the linking costs are explored: A unitary cost with every node, random distributed costs, and a cost that increases with the neighbour's degree. Two scenarios are also explored. In the simplest, all pairs of nodes are directly connected and the distance between them is one. The second one uses a model of the Internet, with the latency as distance metric. Various maximum degree bounds are also tested. Main observations are the following. Increasing the linking cost with the neighbour's degree, has the effect of balancing node degrees in the graph. With this model, the final overlay topology is never a star. Star topologies suffer the most from attack, and resist the best to failures. More degree-balanced graphs are less resilient to failures but their cost in longer paths and lower resilience to failure pays off in case of attack. When all nodes have the same degree, an attack is equivalent to a failure. When the maximal degree bound is high, many nodes establish a link with a small amount of nodes near the centre. These core nodes serve as shortcuts to other nodes, then also reducing the stretch. There is an important tradeoff between performance and resilience of overlay networks. The more the node degrees are balanced, the more resilient the network is to attacks. Limiting the degree achieves high resilience at the cost of higher stretch. In general, the network with the lowest stretch have the worst attack tolerance. Imposing node degree bounds is a key to create networks with good attack tolerance property. To be resilient to attacks and failures, the game should impose low maximal degree, forcing nodes to establish redundant links. When the linking costs are low in comparison with the paths costs, the overlay topology tends to have more links, then decreasing the stretch. The different games evaluated can produce very different networks, from complete graphs to trees with different properties. The network obtained can present desirable properties, with respect to stretch and resilience, even though nodes are not interested in such global properties. There is a fundamental tradeoff between stretch and resilience, that can be controlled by restricting the maximum node degree.

In [LM07], the performance obtained over different overlay topologies are compared, in terms of failure recovery ratio and recovery path hop penalty. The failure recovery ratio is defined by the probability of having an available path on the overlay when the IP-path fails. The recovery path hop penalty is defined by the ratio of the length in hop between the recovery paths found on the overlay and their respective failed IP-path. The performance obtained over different overlay topologies vary a lot. Hence, an overlay topology must be carefully designed. The full-mesh overlay topology is compared to various overlay topologies satisfying a maximal overlay nodes degree. Authors argue that limiting the overlay nodes degree greatly reduces the monitoring overhead and show that, above a certain threshold, it does not degrade the performance. They moreover define two heuristics that favour the overlay link path diversity, i.e. avoid as much as possible the share of an underlay link by several overlay links. These provide the best performance, showing that the underlying IP-layer network information can benefit a lot in constructing efficient overlay topologies.

The node placement problem

In all above works, the problem of building an overlay is equivalent to the selection of virtual links between a given set of overlay nodes, already placed in the network. In [HWJ05], a node placement methodology is also given. It consists of selecting nodes in every ISP and selecting several ISPs, favouring the path diversity of the possible overlay paths while not degrading the performance obtained on them. The path diversity is good if and only if the number of routers shared by a pair of overlay paths, averaged over all possible pairs, is low. The study shows that, for reaching a good path diversity, several overlay nodes should be placed in every ISP. It also demonstrates that the choice of a good subset of nodes is even more critical for obtaining a low path latency. It presents a clustering-based heuristic, which identifies a subset of overlay nodes S_1 which provide a high path diversity, and a subset of overlay nodes S_2 that offer a set of overlay paths over which at least one provides very good performance. A good overlay nodes placement is produced by choosing randomly one node in each subset. Several ISPs should be also selected in order to provide a good overlay path diversity. For path latency, the choice of different ISPs is not relevant. In particular, the study shows that some bad ISPs should be filtered out and the others selected. The evaluation shows that the placement method increases significantly the performance of a RON. It also confirms that a single-hop detour performs as well as a multi-hop one, a feature already detected in the original RON paper [ABKM01].

2.3.3 Overlays on ad hoc networks

Application-level multicast

Application-level multicast afford similar advantages in the context of mobile ad hoc networks, as on Internet. These are the ease of deployment and management, the ability for an incremental deployment, and application-driven distribution. As the state maintenance must only be assured in member nodes, it is also more scalable than native solutions. It moreover helps face the additional problem of frequent topology modifications encountered in ad hoc networks. A distribution tree built of unicast tunnels remains operational even in case of underlay link or node failures.

Most of the application-level multicast solutions for ad hoc networks are mesh-based. As stated above, the maintenance of a mesh is easier than the maintenance of a tree. Moreover, the redundant links in a mesh offer more resilience than a tree, which is valuable in a wireless environment.

AMRoute [XTML02] is a mesh-based solution, independent from any underlying routing protocol. A new multicast member, source or receiver, joins the mesh by sending successive join requests on increasing rings. When another member receives its request, a bi-directional IP tunnel is established between them. The construction phase of the overlay is thus topology-aware, setting up short unicast tunnels. However, there is no maintenance procedure that would keep the overlay topology close to the dynamic underlay topology. Oppositely, the data distribution structure, which is a bi-directional shared tree, is rebuilt at regular interval. The maintenance procedure is a simple flooding method from a dynamically elected core node. AMRoute thus offers a periodic tree-reconstruction over a static mesh.

The *Progressively Adapted Sub-Tree in Dynamic Mesh* (PAST-DM) protocol [GM03] provides better efficiency than AMRoute through the maintenance of a dynamic, topology-aware, mesh and an advanced data distribution technique. The virtual mesh gradually adapts to underlay topology changes in a distributed manner. When a new member joins the overlay, it runs a neighbour discovery process using the expanding ring search technique. A maximum distance between any pair of overlay neighbours is defined. The maximum degree of the virtual mesh is also controlled. Each member keeps track of the other members in its vicinity, by querying its local underlay routing protocol table or periodic neighbour discovery operation. These mesh links are advertised to all multicast members by the link-state technique of the Fisheye State Routing protocol [PGC00]. Each multicast member thus maintains a global map of the virtual mesh. PAST-DM uses a source-based tree approach for content delivery because it is more efficient than a shared tree method. A Source-Based Steiner tree algorithm is defined. The algorithm is run by the data source and successive receivers on their local mesh map. At each node, the algorithm determines which subset of the overlay neighbours must still receive the data. It also splits the list of multicast members that are still to be reached in subgroups, such that each overlay neighbour is responsible for the delivery of the data to one subgroup.

NICE-MAN [Blo04] is an improvement of the NICE protocol for mobile ad hoc networks. As described in Sec. 2.3.1, the NICE protocol is an application-layer multicast system for the Internet. The author states that Internet overlay multicast protocols are not adapted to MANETs because they rely on the assumption of a quite static topology and low packet losses. In particular, it points out that latency is not a good distance metric in MANETs. It shows that latency measurements are very unreliable, and heavily depend on the current network load. Their values fluctuate a lot due to unpredictable delays through the medium access and retransmissions at the link layer. It proposes to use the hop distance, which is more stable. It also argues that the hop distance can very often be derived from the local unicast routing table. The NICE-MAN structure is also more dynamic than in its original version. A multicast node may change of cluster because of underlay topology changes, becoming closer to another cluster leader. A new cluster leader may also be elected if the current one is no more positioned near the cluster centre. The concept of Local Broadcast Clusters (LBC) is also introduced. Any overlay node is a LBC leader. New overlay nodes within transmission range of another overlay node (or LBC leader), do not join the overlay themselves. They send and receive multicast messages through their LBC leader. When moving, they implicitly change of LBC leader, and will enter the overlay if they are no more covered by any LBC leader. This approach greatly improves scalability by reducing the control overhead and building more stable overlays. Noticeably, the number of overlay nodes does not depend on the total number of members, but on the size of the area where group members are located.

The *Multicast Overlay Spanning Tree* protocol for ad hoc networks (MOST) [RNL07] is a tree-based approach. It must be used in conjunction with a link state routing protocol. The underlay routing table collects topological information, necessary for the maintenance of the shared tree. The protocol has been implemented and tested as an extension into a OLSR module. When a node joins the overlay, it broadcasts Join messages. It also switches to full-OLSR mode, which means that it starts advertising its complete physical neighbour set.[4] Each overlay node learns the identity of all the other overlay nodes by listening to Join messages. With the information collected by OLSR, it can compute its distance to every other overlay node. When the overlay nodes advertise their whole physical neighbourhood, each of them can also compute the distance between every pair of overlay nodes. Hence, each overlay node knows the complete overlay topology. An efficient minimum spanning tree algorithm is defined. By using it, every overlay node can compute locally and independently the same minimum spanning tree, covering all the overlay nodes, with an average running time comparable to a Dijkstra algorithm. MOST is well suited for managing numerous groups of small size, with arbitrary sources.

Peer-to-Peer systems

Ad hoc and P2P networks share several design goals and routing principles. They both are self-organised and decentralised. They are adapted to a dynamic environment, and give support for routing on a flat topology with frequent changes. Several papers exploit this synergy by combining the techniques developed in each field. However, it has also been pointed out, by simulation and experimentation, that the straightforward implementation of a Distributed Hash Table (DHT) over a legacy MANET routing protocol, with respect to the OSI layering concept, is inefficient [DB04,Del05,CGT05]. The P2P solutions employed in the Internet support frequent joins and departures of P2P participants, i.e. nodes churn. However, they suppose that the underlay topology is stable, and are not adapted for ad hoc networks. In order to face frequent path breakages, many proposals directly integrate the P2P application in the network layer, in order to exploit their interactions. The network routing process collects topological information that is useful for the P2P optimisation, while the P2P application limits the number of underlay paths that must be maintained in order to reach any destination through an indirect overlay path. This approach has been successfully used for deploying new application-layer services [KLW03, PDH04, DB04, ZS05, KKF06] as well as new unicast routing protocols [HDH03, ZS06]. However, most of these solutions require that all the ad hoc nodes take part in the P2P application. A different approach consists of using a cross-layer optimisation [Del05, CGT05]. In this case, the layered organisation is respected, but the protocols interaction capabilities are extended beyond standard layer interfaces [CMTG04].

As for ad hoc routing, we do not discuss solutions that use location information.

The synergy between P2P overlays and MANETs is discussed in the position paper [HDH03]. The *Dynamic P2P Source Routing* (DPSR) rout-

[4] In normal mode, an OLSR node advertises only the subset of its neighbours that have selected it as a multipoint relay.

ing protocol, that combines DSR and Pastry, is provided as a supporting example. The packets are routed over a Pastry ring with a key obtained by hashing their IP destination field. The unicast direct routing between Pastry neighbours is provided by DSR. The advantage of this technique is a reduction in the routing table size and routing control traffic. For any underlay of size n, each node has a maximum of $O(\log(n))$ Pastry neighbours. Each node thus only builds and maintains $O(\log(n))$ routes, while with DSR alone it could have to build n routes. Hence, the authors argue that DPSR should be more scalable than previous MANET routing protocols. Other structured P2P protocols could potentially be used as well.

The Optimised Routing Independent Overlay Network (ORION) [KLW03] is a P2P file-sharing system for ad hoc networks. As opposed to the P2P systems in the Internet, it does not employ static overlay connections but sets them on-demand, i.e., during the query processing. It thus fits the overlay reactive approach that we define in Chapter 3. It uses the AODV route discovery mechanism at the application layer, with hit on keywords instead of IP addresses. Techniques are defined for processing efficiently a file request containing several keywords. The application-layer query processing is handled simultaneously with the network layer process of route discovery, which substantially reduces the control overhead. It also has the advantage of transferring files on paths that are close to the shortest path, even in case of frequent topology changes. The system is qualified as an overlay because it is completely implemented at the application layer and does not depend on the underlying routing protocol. However, all ad hoc nodes must be able to process the P2P application messages, even if they do not store any file nor make use of the P2P application.

Ekta [PDH04] also integrates Pastry with DSR at the network layer, producing several optimisations:

- The DSR protocol, operating in promiscuous mode, collects and refreshes routes to the Pastry overlay neighbours, without any control overhead.
- In the original version of Pastry, when several entries are valid, the closest next overlay hop is selected. In Ekta, the freshest among the shortest next overlay hop routes is chosen.
- The next overlay hop may be a Pastry neighbour as well as any destination in the DSR routing table.
- Assume that a data packet arrives at a node and that the best next overlay hop is not associated with a valid physical route. If it is a Pastry neighbour, a DSR route discovery is done for that node. If, on the other hand, it comes from the DSR routing table, a DSR route request is broadcast for the prefix required, not for a full address.

This DHT substrate is also applied, as a case study, to the discovery of resources in MANETs. The resource identifier is hashed in the overlay namespace. The node with this key as overlay node identifier provides a list of nodes that offer this resource. The same resource discovery mechanism has also been proposed over DPSR in [DB04].

MadPastry [ZS05] combines AODV and Pastry. Key prefixes are assigned to a small set of dynamic landmark nodes. The nodes in the vicinity of a landmark dynamically assign themselves an overlay identifier which begins with the corresponding prefix. Hence, nodes that are close to each other in the logical overlay identifier space are also likely to be close to one another physically, i.e. overlay clusters are formed. As a result, this variant of Pastry provides better performance in the ad hoc environment than its original version.

The DHT indirect routing solution offered by MadPastry is used for performing direct physical routing in [ZS06]. The proposal lies on the concept of address servers. The overlay identifier of the address server of any node N is obtained by hashing the IP address of N. Each node informs its address server anytime its overlay identifier changes. When a node A needs to send data to some node B, it first sends a request to the address server of node B, with the MadPastry overlay routing process, for obtaining B's current overlay identifier. It then sends the data to B, using the MadPastry routing again. Simulations results are very good.

In [KKF06], another structured P2P solution for ad hoc networks is proposed, where the logical overlay is organised into a unique ring. Each overlay node maintains a physical route to only two nodes, one predecessor and one successor. On the physical path between two overlay neighbours, an overlay request may traverse an overlay node which is closer to its final destination than the next overlay hop to which it was initially travelling. In this case, the overlay request is deviated, which accelerates the lookup's convergence. The evaluation is only conducted in a static network, composed only of nodes that take part in the P2P application. We however expect that this elegant solution will also work in a network partially composed of overlay peers, if the overlay messages are grabbed by the overlay nodes (see Sec. 3.2.2).

CrossRoad [Del05] is a cross-layer architecture for service discovery, running Pastry over a proactive routing protocol. Services information is added as optional field of routing packets and automatically sent on the network. The additional information is stored in a data sharing module, defined in [CMTG04], accessible by all layers, in particular by the Pastry middleware. The Pastry nodes thus get the identifier of other Pastry nodes without any peer discovery process. Moreover, they become aware of the topology changes with the same delay of the routing protocols, with a very low overhead.

XL-Gnutella [CGT05] applies the same cross-layer architecture, for adapting the Gnutella unstructured P2P protocol to the ad hoc environment. In order to remain fully compatible with the Gnutella legacy version, an overlay edge selection algorithm maintains the number of neighbouring peers between 4 and 8.

In both cases, the cross-layer approach eliminates the need of bootstrap servers, reduces and stabilises the overhead of the protocol. Moreover, it improves the connectivity and the quality of the generated overlay, by prioritising the establishment of closer connections over farther ones. It also allows to implement the P2P middleware only on a subset of the ad hoc nodes.

Other works

Beyond the application-level multicast and peer-to-peer research fields, the deployment of layer 4 overlays in MANETs has not yet received a lot of attention. We only noticed a few works that we present below, intended respectively for service discovery, data security, delay-tolerant routing and distributed computing.

In [KKRO03b] and [KKRO03a], an overlay approach is proposed for service discovery. The additional layer builds and maintains an overlay structure that adapts to the variable underlay topology and offers efficient mechanisms for service trading. Centralised and flooding-based approaches are rejected because of the decentralised and resource limited nature of ad hoc networks. Traditional P2P systems are also avoided because of their complexity. Hence, a semantically restricted flooding method is proposed. The basic structural elements of the overlay topology is a service ring. A service ring ideally groups together devices that offer similar services and at the same time are geographically close to one another. In [KKRO03b], a single node is responsible in each ring for summarising the set of services offered. This node also belongs to a new ring at the next-level in the hierarchy. In [KKRO03a], a two-dimensional overlay structure which is similar to, but less strict than the one used in the Content Addressable Network (CAN [RFHK01]) is used. Service advertisements are propagated along one dimension of the overlay, so that each node knows all the services offered in its ring. Service requests are propagated along the other dimension, traversing successive rings. In both cases, large parts of the network do not need to be visited at all by the service queries. As the service discovery is presented as a necessary primitive for the usability of ad hoc networks, the protocol is designed for and evaluated in an ad hoc network fully composed of nodes that support it.

The system described in [LD07] provides secrecy for application data. It runs over a software system for application-layer overlay networks, called HyperCast [LJG03]. Each overlay node shares a secret key with authenticated overlay neighbours. The neighbourhood key method avoids network wide re-keying operations, without requiring that payload data be re-encrypted at each hop. An overlay protocol component of the system establishes and maintains the overlay topology, which is a spanning tree topology. The presented overlay maintenance protocol is a variant of Perlman's spanning tree for bridges [Per85]. The link quality and hop count are both tested as metrics for comparing the paths length to the root node. The tree can be easily used for multicast as well as for unicast forwarding. Its performance for unicast transmission is compared to the one of AODV. The delivery ratio is a bit inferior and the number of packets emitted per message delivered is higher because of path stretch. However, the delivery delay is also lower because AODV buffers packets in case of broken or unavailable route, while the data is always directly flooded on the overlay spanning tree. The evaluation consists of simulation and experimentation, with and without the security functionality. The protocol's min-hop version provides better results in simulations, while the link-quality is the best in experimentations. Overall, the efficacy of application-layer ad-hoc networking with (and without) data security has been successfully tested. However, in this work again, the topology maintenance protocol is designed for an ad hoc network totally composed of overlay nodes.

Oppositely, for the delay-tolerant approach called Ad-hoc Storage Overlay System (ASOS) [YLJC+06], it is more desirable to designate only a subset of the nodes to act as ASOS peers. These should be the storage-abundant nodes. They organise themselves into an overlay, and offer a global service for the whole network. When the destination of a data flow becomes unreachable, because of a path failure or network partition, the undeliverable data is reliably stored in the overlay. The data is delivered later to the destination, when connectivity improves. In the push mode, each ASOS node detects the appearance of a new node in its local routing table, and sends the information it may have stored locally for this destination. In the pull mode, a destination may retrieve the information from the overlay, when receiving an advertisement containing its identity. Each ASOS peer thus periodically aggregates its knowledge of peers and files stored in the overlay, and broadcasts this information on the whole network. The paper does not discuss the overlay topology maintenance. The ASOS peers are simply assumed to form a multicast group inside which periodic hello messages are sent.

The overlay topology itself is not a central question in works about distributed applications over ad hoc networks. The ad hoc devices have increasingly powerful hardware. Hence, the combined computational power of systems built with MANETs can be large. However, research in this field mostly focus on the middleware architecture for distributing users applications among the participating nodes. The main functionalities studied are the modelling of the applications needs and of the nodes resources, the mapping between them, the nodes and applications monitoring, and re-mapping during the execution.

2.4 Positioning our work

We study the feasibility of overlay routing on top of ad hoc networks and give some hints in that direction. We discuss in particular which overlay topologies would offer a good ground for efficient overlay routing.

In previous sections, we have presented a survey of ad hoc networking, with a focus on topology control and routing, and of overlays. The deployment of layer 4 overlays in MANETs has not yet received a lot of attention. However, this field has already been largely explored for the Internet. In the next section, we expose how these results influenced our problem conception and research options. We then point out works in the field of ad hoc topology control and routing that were of particular interest for our own investigations. We finally analyse the relevance of our results to the four main application fields of overlay networking.

2.4.1 Problem definition and solution keys

Community overlay

In [PDH04], authors claim that since an ad hoc network is typically formed of nodes that collaborate with each other, it is rarely necessary to construct an overlay that consists of a subset of the nodes. We take a different angle of view.

In many places, the current concentration of wireless users in the free-band allows the creation of large, spontaneous, ad hoc networks. We propose the overlay approach as a response to the heterogeneity of user resources and needs. Overlay routing is a technical means for each group of similar users, or community, to employ its custom routing strategy.

Hence, we do not consider the placement of nodes as a degree of freedom. The community is made of wireless devices that are more susceptible to communicate with each other than with a device external to the group. It can be defined for example by some hardware capabilities or application.

The construction of an overlay topology consists in selecting virtual links between this given set of ad hoc nodes, and not the placement nor choice of overlay nodes in the ad hoc network.

Topology-awareness

Ad hoc networks can appear spontaneously, with very little coordination of its users. We consider that this technical feature is a strength of the technology and must be respected in the design of new protocols. We also take into account that nodes and links are volatile in the ad hoc environment.

Hence, we adopt a flat strategy for building overlays. We assign the same role and behaviour to any overlay member. We do not make use of landmark nor bootstrap nodes or rendezvous point, and do not build any hierarchy.

We avoid scalability problems by defining protocols that are fully distributed and local. The locality principle imposes that every overlay node exchanges only a few control messages with other overlay nodes, located in their vicinity. Hence, we select overlay links that are as short as possible. This creates an overlay path stretch but, as shown in [ZS06], it can be advantageous to travel numerous routes that are relatively short and likely to be up-to-date instead of travelling on a single long and direct path. The metric used for the overlay links length is the hop metric.

Genericity

A substrate-aware architecture for the construction of generic overlays on ad-hoc networks, similar to the two proposals for Internet Saxons [Che04] and PLUTO [NP06], would be a powerful tool for the development of community overlay routing. In such framework, our work would contribute to the management of the ad hoc overlay structure. In particular, finding the nearest neighbours and building a routing mesh are services proposed in the SAXON architecture, for which we provide technical solutions in the context of ad hoc networks.

In every overlay application, building and maintaining the overlay topology is a basic functionality. The topology construction could be directly integrated in the applications, and optimised for them. However, as in many papers that discuss the quality of an overlay topology on the Internet, we rather propose a generic solution than focus on a specific application.

Many works show that collecting topological information about the underlay can be very beneficial to the construction of efficient overlay topologies. We obviously agree with that observation but we avoid as much as possible cross-layer optimisations for obtaining this type of data.

We also reduce the number of assumptions about the underlay topology or routing protocol to a minimum.

For our performance evaluation, we define in Sec. 5.3 generic performance criteria based on overlay flooding.

Evaluation of the quality of the overlay topologies

We first define the overlay connectivity and scalability as fundamental requirements. Chapters 4 and 5 are respectively dedicated to a theoretical and to a heuristic analysis of the overlay connectivity, based on similar studies conducted in the field of topology control.

We then insist on performance. We use our generic performance criteria in Sec. 6.2 for comparing several overlay topologies, without considering their implementation. One overlay topology tested in [LM07], the Adjacent Connection with maximal degree, is directly related to our work. In an Adjacent Connection overlay topology, there exists an overlay link between a pair of overlay nodes if and only if there is no other overlay node on the IP-path between them. This rule is a particular case of the pruning method that we present on Sec. 6.2.3. In order to set a maximal degree, if an overlay node's degree exceeds d_m in the AC topology, only its d_m closest neighbour overlay nodes are kept. We note that the Adjacent Connection overlay topology performs well. The overlay topologies that we study are also based on the discovery of a constant number of the closest overlay nodes.

For the purpose of resilience, we propose to limit the minimal overlay nodes degree. However, much more work could be conducted on the overlay resilience. In particular, we let the study of the tolerance to failures and attacks as further work.

The path independence of the overlay edges is also an interesting feature that we do not take into account.

2.4.2 Technical ground from the ad hoc network field

Topology control

We expect that a good overlay topology must be sufficiently dense so as to provide an overlay path between each pair of overlay nodes and sufficiently sparse to limit the number of packets emitted during the diffusion of messages on the whole overlay topology. As this trade-off is similar to the topology control (TC) problem in ad hoc networks, the literature on this field has been an obvious source of ideas for our investigations.

Roughly, we adapt the homogeneous topology control principle to overlays in the Reactive Overlay Approach and non-homogeneous topology control methods in the Proactive Overlay Approach.

With a homogeneous topology control algorithm, all nodes adopt the same transmission range value. Similarly, for the Reactive Overlay Approach, all overlay nodes adopt the same neighbourhood range value, defined as the maximum number of hops between each pair of overlay neighbours. In Chapter 3, we conduct a feasibility study of this approach. In Chapter 4, we present the Critical Neighbourhood Range (CNR) problem, which is an extension of the CTR problem. We use a model based on random geometric graphs and provide asymptotic results, i.e. when the number of nodes tends to infinity. We do not take into account interferences. The network field length may evolve with the number of nodes or have a constant size.

We next conduct in Chapter 5 a heuristic study, for finite networks, on how many overlay neighbours each overlay node should try to reach in order to form a connected overlay with a high probability. This type of heuristic is used in the discovery phase of the non-homogeneous TC protocol *k-Neigh*.

We propose its use in the context of both the Proactive and Reactive Approaches. In the latter case, we suggest in Chapter 6 to optimise the overlay topology by applying a pruning rule, similar to the one defined for the non-homogeneous TC protocol *XTC*, that builds the RNG upon the max-power graph modeling the ad hoc network.

Routing

The objective we fixed for our overlay topologies is that they must offer a good ground for efficient overlay routing. We try to handle this problem in a generic way. Hence, we do not target any specific underlay or overlay routing protocol.

Concerning the underlay, our simulations are conducted over AODV and OLSR, which are well-known and efficient representatives of the two main families of flat routing protocols. We do not test our solutions over hierarchical protocols.

Concerning the overlay, we try to abstract quality metrics for an overlay topology in regards with many potential overlay routing protocols. From our survey of ad hoc routing protocols, we infer that the efficient diffusion of the overlay routing control messages is a key point. We thus propose in Section 5.3.2 generic performance metrics based on the flooding of overlay messages.

2.4.3 Applications

We expect the overlay solutions for ad hoc networks to present the same advantages and drawbacks than on the Internet. We however suggest that the overlay technique specially fits to ad hoc networks because of their high needs in flexibility. The overlay topologies studied in the following and the protocols we propose for building them can be used for deploying Application-Layer Multicast. It can also be exploited for P2P applications, but with a possible restriction to unstructured solutions. Enhancing the underlay routing solution and allowing the test of new routing protocols are the main motivations of our work.

Application-layer multicast

As described at the beginning of Sec. 2.3.3, the use of application-level multicast presents several advantages in ad hoc networks compared to native multicast.

Most of the proposed solutions are mesh-based and do not require any rendezvous point. These features are shared with our work.

As we do not specifically target on multicast, we do not define how a data distribution tree should be built. However, we do describe means for building a dynamic mesh on which good quality trees could be produced.

Our technique uses a join procedure based on an expanding ring search, common to many existing application-layer multicast protocols for ad hoc networks.

Like AM-Route, our technique is topology-aware, setting up short unicast tunnels. In order to keep good performance in a mobile context, AM-Route adjusts periodically the data distribution structure over its static mesh, while we propose to dynamically eliminate tunnels when they become longer, replacing them by new shorter ones.

PAST-DM also adopts a dynamic mesh strategy.

Like in NICE-MAN, we use the hop distance as metric for the length of unicast tunnels. However, NICE-MAN is based on clustering while our overlay topologies are flat.

Our approach is very different from MOST, which is tree-based. Moreover, MOST must be used in conjunction with a link-state routing protocol while the techniques we propose are intended to be independent from the underlying routing protocol[5]. MOST intensively uses the topological information provided by the underlay routing protocol in order to maintain an efficient, topology-aware, data distribution structure. For the purpose of genericity, the OTC protocol, presented in Sec. 6.3, does not use topological information about the underlay. However, when this type of information is available, OTC can exploit it easily. As an effect, its bandwidth consumption lowers a lot[6].

Peer-to-peer networking

The overlay construction and routing of data are two core functionalities of P2P systems. We only study the former in this dissertation. The latter strongly depends on the user application, while we have set genericity as an important objective.

Our technical work complies with the general definition of a P2P system, given in [ATS04]. We discuss the self-organisation of interconnected nodes into a network topology with the purpose of sharing resources, capable of adapting to failures, while maintaining an acceptable connectivity and performance, without requiring the support of a global centralised server or authority. However, we do not address the reliability to nodes churn, as typically evaluated for the P2P protocols. Our work is not specific to P2P systems and the link breakage is a more general concern for ad hoc routing.

Our overlay topologies are optimised for efficient overlay messages flooding. Hence, their applicability to unstructured P2P systems is direct. As an example, we compare them to the XL-Gnutella [CGT05] topologies in Sec. 6.2.4. We also suggest on Sec. 7.2 as further work to combine the

[5] We discuss this point in the concluding chapter.

[6] We infer this by analysis of the XL-Gnutella paper [CGT05], as discussed in next paragraph

cross-layer architecture used for XL-Gnutella and the results we obtain with our Proactive Approach, on Sec. 6.2.

Several papers demonstrate the interest of structured P2P networking over ad hoc networks. In most of the works which implement a DHT substrate in the networking layer, all nodes are assumed to support the protocol. We do not impose this restriction. A community of users can cover the whole ad hoc network, but in the general case, it is only composed of a subset of the ad hoc nodes. The proactive approach seems the best for providing a support to structured P2P protocols. We however let this as an open issue.

Enhancement of the routing service

Improving the basic routing service is an important motivation for our work.

On the Internet, several works have shown that overlays can afford quality of service and fault-tolerant routing. In this dissertation, we do not target a specific overlay application, providing only one, but important, building block for this type of usage: The topology maintenance procedure. As the Internet and ad hoc networks have very different topological characteristics, the overlay topology maintenance procedures proposed for the Internet do not apply to an ad hoc environment.

The only SON for ad hoc networks we noticed in the literature is the ASOS system, presented in [YLJC+06]. Noticeably, this paper does not discuss the overlay topology maintenance procedure. More research is thus required, on a per-application basis, in order to build a SON over an ad hoc network.

We are not aware of papers about fault-tolerant overlay routing in ad hoc networks. RONs route the packets around points of path breakage. As this type of faults is very frequent in ad hoc networks, the usage of RONs in this environment seems to us an interesting research direction. Several papers discuss the best overlay topology for RONs over the Internet. However, as mentioned above, the underlay of Internet and of ad hoc networks have very different characteristics. Hence, the results presented in Chapter 6 (the Proactive Approach) could be helpful for the sub-problem of RONs overlay topology maintenance in ad hoc networks.

On ad hoc networks, there have been only a few proposals for enhancing the QoS on ad hoc networks with overlays. The service rings introduced in [KKRO03b, KKRO03a] are completely different from the overlay topologies we study in this dissertation and suppose that the network is fully made of overlay members. The HyperCast paper [LJG03] seems encouraging for our own researches in several points. Firstly, it demonstrates, both by simulation and experimentation, the efficacy of application-layer routing on ad hoc networks. Secondly, we could inject in its overlay protocol component the solutions proposed in this dissertation and test them. In this case, the spanning tree topology of HyperCast would be replaced by a topology-aware mesh. Moreover, it could be possible to deploy the HyperCast protocol over a subset of the ad nodes only. Thirdly, it shows that measuring the overlay link length with the hop metric, as we do, gives better results than measuring it with the delay metric.

The overlay topology maintenance is generally not discussed in papers related to distributed applications over ad hoc networks. Hence, our advances could be complementary with these works.

Test of new routing solutions

It has been proved, on the Internet, that overlays allow to experiment new routing protocols and can be used as a deployment path. This motivated the design of the X-Bone, an automated overlay deployment and management system. It is not dedicated to a given overlay application. Rather, it lets the designers of an overlay solution concentrate on the specificities of their application, without entering the details of the overlay maintenance. Among various facilities, the X-Bone system controls the overlay topology. Our generic work on overlay maintenance for ad hoc networks provides this building block for an ad hoc X-Bone.

In Appendix A, we explain how the Reactive Overlay Approach, presented in Chapter 3, can be exploited in the active networking field.

In the same way, if a kind of PlanetLab experiment had to be deployed on ad hoc networks, our work could be integrated in a sub-service controlling the overlay topology.

The history of Internet let us imagine that the success of the ad hoc technology, through the deployment of large, spontaneous, infrastructureless networks, requires a global agreement on one routing protocol. Plenty of routing protocols have been successfully evaluated with simulators but have not been tested on ad hoc devices. On one hand, picking one routing protocol, convenient for any ad hoc network, is very challenging. On the other hand, even if such a protocol emerges, being widely accepted, it will become difficult to test or add new routing capabilities on ad hoc networks. In both cases, the construction of virtual testbeds on the basis of the overlay technique, presents the same advantages for ad hoc networks than for the Internet.

Because of their high needs of flexibility, we consider for ad hoc networks, as suggested for the Internet in the Cabo framework [FGR07], that the sharing of a physical infrastructure by several virtual networks could be considered as an architecture in itself. This circumvents the problem of picking one routing protocol, that would be convenient for any ad hoc network. In this context again, our work could contribute to the maintenance of the virtual networks topology.

Chapter 3

Feasibility of the Reactive Overlay Approach

3.1 Chapter overview

We consider, in an ad hoc network, a subset of nodes that have a commonality of interests and are likely to communicate more with each other than with external nodes. We assume that communication paths passing through the members of this logical subnet are preferred to paths going across external nodes. Therefore, the members deploy their own routing application, on top of the underlay routing protocol common to all ad hoc nodes.

We first explain how the subnet members can avoid the expensive process of building an overlay topology, before using their customised routing application. The rationale exploits the broadcast nature of ad hoc networks, and is qualified as a Reactive Overlay Approach.

We then detail a compatible overlay routing application, Overlay-AODV, and evaluates it. Overlay-AODV adapts the route discovery of AODV for the application layer. The study of its performance shows the feasibility and the efficiency of overlay routing applications developed according to the Reactive Overlay Approach.

3.2 The Reactive Overlay Architecture

Consider a subset of nodes in an ad hoc network. The Reactive Overlay Approach allows the subnet members to run an overlay routing application without maintenance of an overlay structure. In other words, the discovery of overlay neighbours is not required for building overlay routes. It is achievable if and only if the overlay routing application is reactive. By definition, in this case, the overlay routes are built on-demand, when some overlay application needs them. An interesting side-effect is that each overlay node discovers a set of overlay neighbours on-the-fly during the dissemination of overlay route requests.

3.2.1 The overlay neighbourhood range

With the reactive overlay approach, the overlay structure is implicitly defined by setting the maximum distance between a pair of overlay neighbours. We call this parameter the *overlay neighbourhood range*, or shortly the neighbourhood range, and denote it NR.

We do not present in this chapter how an appropriate NR value can be estimated. Our goal is to determine if the knowledge of this value at the overlay nodes is sufficient for running efficient overlay routing applications.

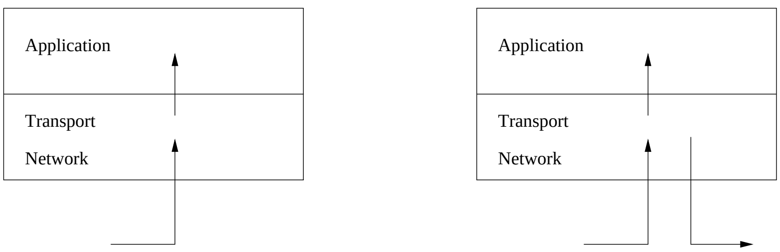
3.2.2 Broadcast overlay messages processing

The simplest procedure for flooding a packet is to send it to all neighbours except to the one from which the packet has possibly been received. This however does not require to know the neighbours identity. In wired networks, a router copies the packet on every interface, except the input one. In ad hoc networks, a broadcast packet with *Time To Live* (TTL) set to one is emitted. In this way, if there is no loss, the packet is received by all neighbour routers.

Likewise, an overlay broadcast message on ad hoc networks is simply encapsulated in IP broadcast packets with the Time To Live field set to the neighbourhood range (NR). As stated above, the neighbourhood range is the maximum number of hops on the optimal path linking an overlay node to any of its overlay neighbours. Setting the TTL field to NR ensures that only valid overlay neighbours will receive the message.

When a node receives a broadcast packet, it first checks if there is some application listening to the type of encapsulated message. If there is such an application, it delivers the message to it. Otherwise, it decrements the TTL field and reads the packet identifier, or *broadcast ID*. If the broadcast packet has already been received or if the new TTL value equals 0, it drops the packet. Otherwise, it re-emits the packet.

An alternative would have been to firstly check if the packet must be re-emitted or not and then deliver a copy to any listening application. The two possible procedures are described on Figure 3.1. We selected the first one, that we call the grabbing architecture, because it saves bandwidth. This property is illustrated by Figure 3.2. Overlay nodes are grey-shaded. The neighbourhood range of all overlay nodes equals 2. Overlay node C thus has two overlay neighbours B and E. It emits a broadcast overlay message at time t_0. The encapsulation of an overlay message and its emission by the network layer is depicted by a dashed arrow. On the upper sub-figure, every broadcast packet with a non-null TTL value is forwarded, without distinction. The network forwarding decision is illustrated with a continuous arrow. At time t_1 the initial packet is forwarded by nodes B and D. At time t_2, the overlay application at nodes B and E forwards the message. It is encapsulated and emitted by the network layer with $TTL = 2$. At time t_3, the packet is forwarded by nodes A and D. These nodes consider that this packet is received for the first time because the respective IP sources are nodes B and E, not C. Finally, at time t_4, node C repeats the packet again. This terminates the flooding of the overlay message on this simple topology, with a total amount of packets emitted equal to 8. The grabbing architecture is illustrated by the lower sub-figure. In this case, the network layer on node B and E does not rebroadcast the received packet because there is one overlay application listening to the received overlay message. The flooding procedure ends with a total amount of packets emitted equal to 6. The gain in bandwidth is even more noticeable on traditional larger two-dimensional topologies.



(a) Grabbing architecture: Overlay broadcast messages are only delivered to the application layer ($NR = 2$)

(b) All broadcast packets are forwarded, without distinction ($NR = 2$)

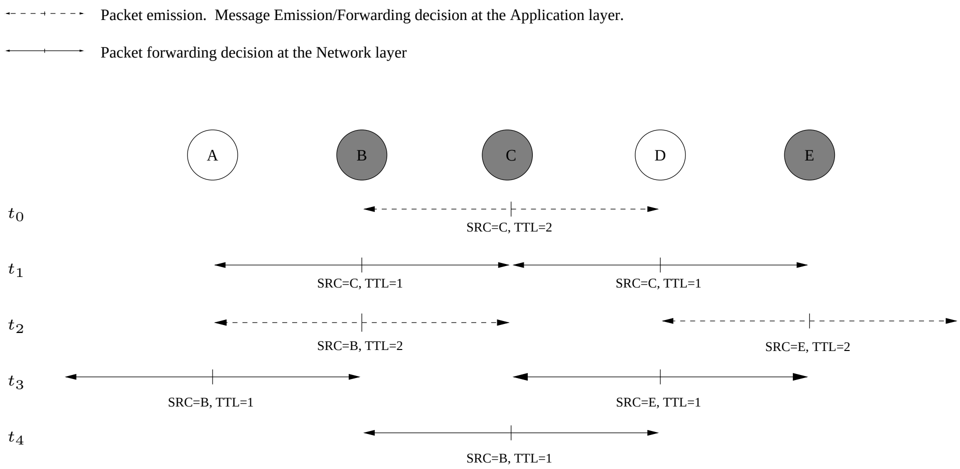
Figure 3.1: Overlay broadcast messages processing at an overlay node with and without grabbing

3.2.3 Unicast overlay messages processing

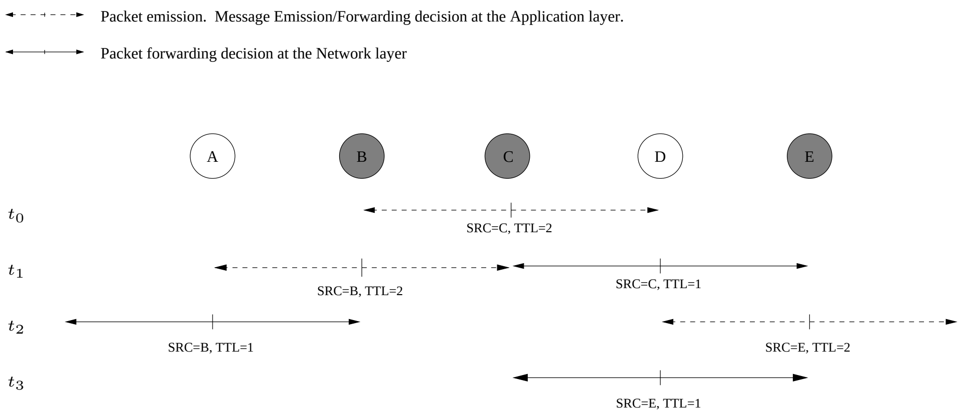
The overlay paths are discovered by the overlay routing application. At the network layer, we only assume that a routing algorithm forwards the messages between two successive overlay nodes on the overlay routes built.

The general flow of packets on an overlay path is shown in Figure 3.3.

When an overlay message is generated by a user overlay application, the corresponding overlay routing protocol provides the IP address of the next overlay hop on the overlay path to the final destination of the message. The



(a) All broadcast packets are forwarded, without distinction



(b) Overlay broadcast messages are only delivered to the application layer

Figure 3.2: Bandwidth is saved when the broadcast messages are grabbed

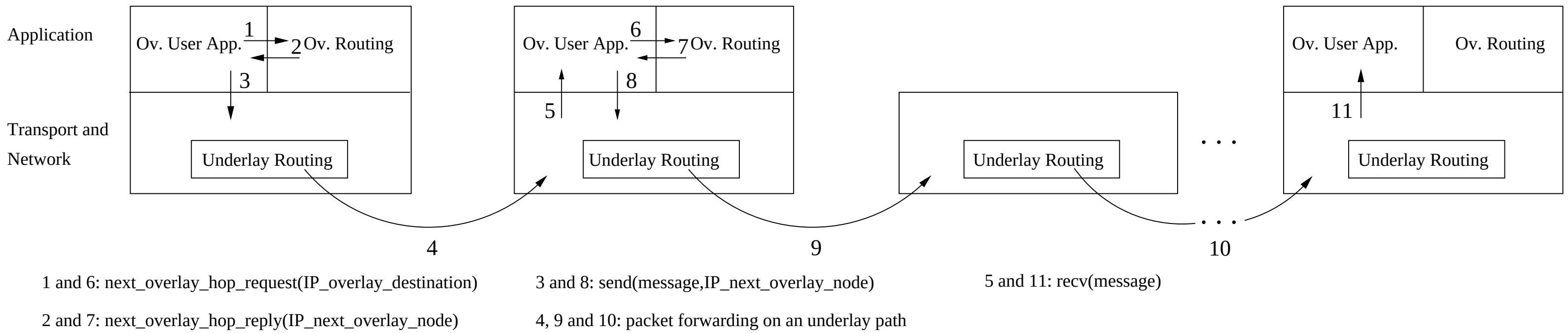


Figure 3.3: Overlay routing: Main steps of an overlay message transmission from source to destination

overlay user application gives the message to the transport layer, indicating this intermediate overlay node as destination. The transport layer in turn delegates the packet forwarding to the network layer. The latter uses the underlay routing protocol for determining the physical path to the next overlay node.

When an overlay node receives the message, it delivers it to the listening application. The local overlay application consults the overlay final destination. If it is a distant node, it asks the corresponding overlay routing application to determine the next overlay hop as described above.

3.3 The Overlay-AODV overlay routing application

As stated above, the overlay routing application presented in this Section is an adaptation of the AODV protocol to the application layer. In the following, we will call it Overlay-AODV. We describe its design and then evaluate it in detail on top of AODV.

3.3.1 Overlay route construction

An overlay route is created only when it is needed by a user overlay application. As for AODV, the route discovery follows a route request/route reply query cycle. An overlay source node in need of a route broadcasts an overlay Route Request (ORREQ) packet across the network. Any node with a current route to the destination, including the destination itself, can respond to the ORREQ by sending an overlay Route Reply (ORREP) back to the source node. Once the source node receives the ORREP, it can begin sending data packets along this route to the destination. In order to prevent unnecessary network-wide dissemination of ORREQs, the originating node uses an expanding ring search technique. It initially uses a small Overlay Time To Live (OTTL), in case the destination node is located close to itself. If necessary, it later sends other ORREQs with a progressively increasing OTTL, until it receives an ORREP. The maximum OTTL value is bounded because the destination node can really be unreachable.

In the AODV protocol, the route reply (RREP) messages are sent in unicast, following the reverse route built on the intermediate nodes. On the overlay, the ORREP messages also follow the reverse path, but they are encapsulated in broadcast packets. Indeed, if an ORREP is to be unicast from one overlay node to its predecessor P, the underlying reactive protocol may not know a passive path to P and may have to search a route before forwarding it, which would generate unnecessary delays[1].

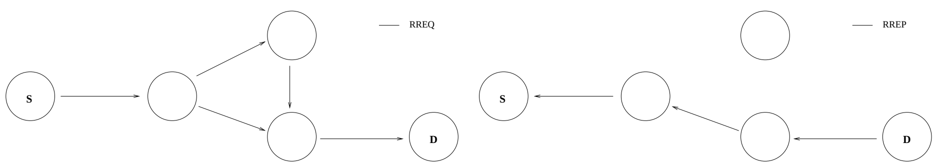
[1]Notice that this is an optimisation for underlay reactive routing protocols. If, for example, we had chosen a proactive protocol like OLSR [JMC$^+$01], it should have been

When an overlay node receives an ORREP, it analyses the next overlay hop field. If this field contains its identity, it is the good predecessor so it processes the message. If not, it simply drops it.

Ad hoc random access MAC protocols often treat unicast and broadcast packets differently. Unicast packets may be preceded with MAC layer control frames, such as RTS/CTS followed by ACK, to reduce the risk of interference. Broadcast packets, on the other hand, are sent blindly without any control frames to assure the availability of the destinations [TG00]. This makes broadcast packets more likely to be lost than unicast ones. To avoid ORREP losses, the reception of an ORREP is confirmed by the emission of an ORRACK. Two retransmissions of an ORREP are allowed. The ORRACKs are unicast; this is more reliable, does not increase the route construction time and finally opens the successive underlay path pieces between consecutive overlay nodes on the global overlay path, decreasing the end-to-end transmission delay of the first overlay data message.

Overlay-AODV can work as well in sparse as in dense mobile overlay environments. The ORREQ and ORREP messages are encapsulated in IP broadcast packets with the Time To Live field set to the neighbourhood range.

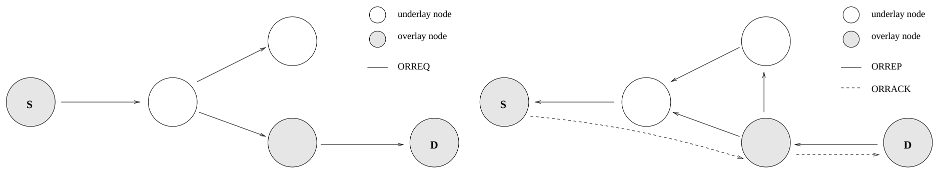
If AODV uses an optimal path between two overlay neighbours, the overlay messages flowing between them will not go through more than the $NR - 1$ underlay nodes. Because AODV sometimes provides paths a little longer than the optimal one, we will assume, for small neighbourhood ranges, that the maximum number of underlay nodes traversed by a message between two overlay neighbours equals NR. Figures 3.4 and 3.5 illustrate respectively the AODV and Overlay-AODV route discovery processes on a tiny topology. The neighbourhood range is set to two.



(a) Route requests propagation (b) Route replies propagation

Figure 3.4: AODV route discovery process

more appropriate to unicast the ORREPs because the underlay path would have been immediately available.



(a) Overlay route requests propagation

(b) Overlay route replies and route reply acknowledgements propagation

Figure 3.5: Overlay-AODV route discovery process ($NR = 2$)

3.3.2 Overlay route maintenance

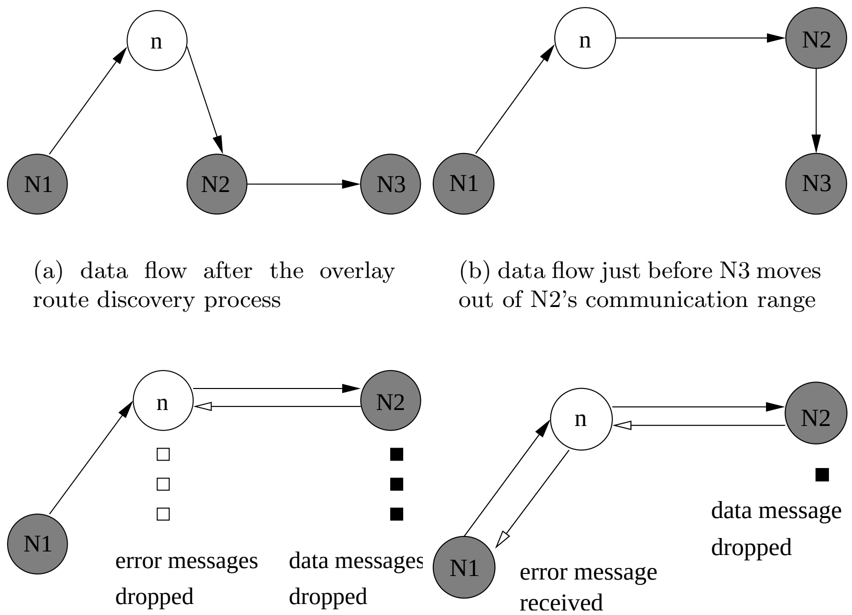
Because nodes are moving, link breakages are likely to occur, which in turn breaks underlay paths between successive overlay nodes.

As AODV, the overlay routing application can operate in two modes: Hello or overlay link break detection.

In the hello mode, each overlay node lying on an overlay route emits at regular intervals a broadcast overlay hello (OHELLO) message carrying its identity (IP address and overlay application type). The OTTL field of the overlay message is set to one and the TTL field of the IP packet which carries it is set to the neighbourhood range. A predecessor not receiving these messages knows that the overlay node can no more be used as next overlay hop and informs the source that the overlay route is broken.

In the overlay link break detection mode, when the source of a underlay routing path receives a route error message (RERR), the overlay routing application is informed. This is the only modification we impose to AODV, and it is necessary only in this mode.

In AODV, when a link breaks on an active route, the node upstream of the breakage broadcasts a RERR message containing a list of all the destinations which are now unreachable due to the loss of the link. On the other hand, when an overlay link breaks on an active overlay route, the upstream overlay routing node sends a separate *unicast* message to each of the overlay neighbours affected by the breakage. This is less bandwidth-efficient and, if the flow is unidirectional, the underlay path to the predecessor must be discovered. The source may thus be notified of the break later in this way. However, this is necessary in the overlay link break detection mode, because there exist scenarios where, if we sent broadcast overlay route error (ORERR) messages, all successive ORERR messages emitted by an overlay node would be lost. Let us for example assume that the neighbourhood range in the sequence of events illustrated by figure 3.6 is set to one. Consider the overlay path N1-N2-N3 in part (a) of this figure. N2 is in the radio range of N1 but AODV has found a two hop route between them, going through the underlay node n. In part (b), N2 has moved out of range of N1 but the data messages continue to be delivered because the underlay route N1-n-N2 is still valid. In parts (c) and (d), the underlay path between N3 and N2 is broken, so N2 sends a route error message. If this message is broadcast, as in part (c), its Time To Live field is set to the neighbourhood range value, one. When n receives the broadcast packet, it drops it because the Time To Live, after decrement, is null. If the message is unicast, as in part (d), it does reach N1.



(a) data flow after the overlay route discovery process

(b) data flow just before N3 moves out of N2's communication range

(c) broadcast error messages do not reach the data source

(d) unicast error messages do reach the data source

Figure 3.6: Overlay error messages must be unicast

Table 3.1 summarises the control messages used by AODV and the overlay routing application, and their emission mode.

3.4 Overlay-AODV evaluation with the reactive overlay approach

We study the impact of the network density and diameter, of the overlay density and of the mobility level on the protocol. The network density refers to the number of ad hoc nodes per area unit while the overlay density refers to the proportion of overlay nodes in the network.

We use *n*s-2 simulator [VIN] with the extensions from the Monarch

Table 3.1: The control messages used by AODV and Overlay-AODV

	AODV packets		Overlay routing messages		
Message type	Emission mode	IP TTL	Emission mode	Overlay TTL	IP TTL
Route Request	Broadcast	$[1, NETD]$	Broadcast	$[1, ONETD]$	NR
Route Reply	Unicast	$NETD$	Broadcast	$ONETD$	NR
Route Reply Ack.	-	-	Unicast	1	$NETD$
Route Error	Broadcast	1	Unicast	1	$NETD$
Hello	Broadcast	1	Broadcast	1	NR

$NETD$ = estimated network diameter
$ONETD$ = estimated overlay network diameter

Project [Ric]. We use the AODV implementation provided with ns-2.26.

We first analyse the overlay routing application performance with static nodes only and then study the impact of mobility.

3.4.1 Static networks

Static tests description

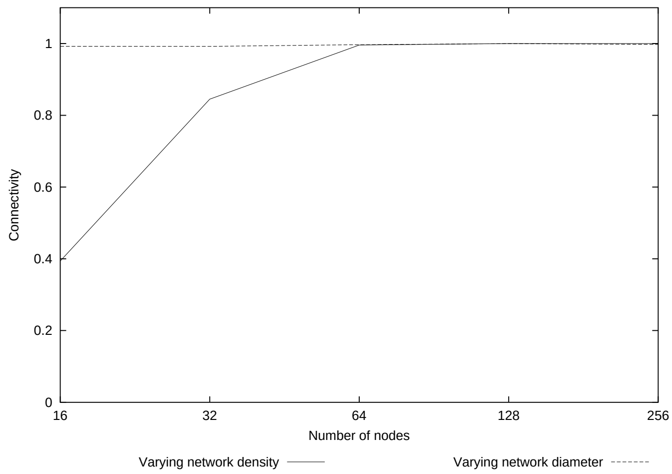
We operate with static and uncongested nodes. The nodes transmitting range is set to 250 meters. Two nodes are said to be connected if there exists a multi-hop path between them.

With a first set of experiments, we study the influence of the network density on performance. The nodes are distributed uniformly and independently in a 1000 meter long square field. The density obtained with 64 nodes is taken as a reference and noted D_{ref}. We test densities of 0.25, 0.5, 1, 2 and 4 times D_{ref} by disseminating respectively 16, 32, 64, 128 and 256 nodes in the field.

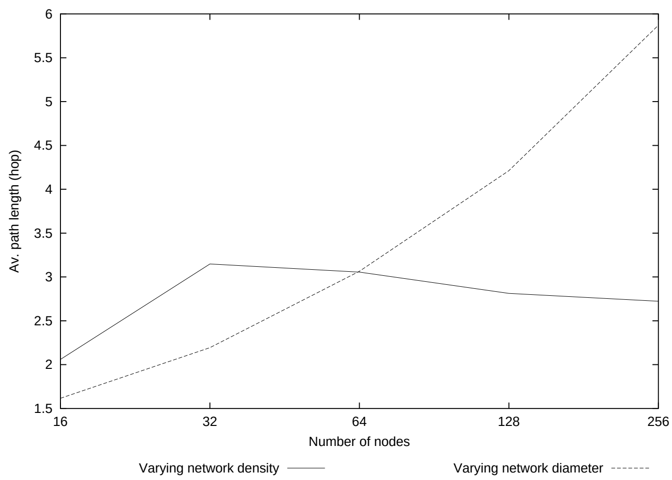
In order to study the network diameter effect, we construct a second set of topologies. In this set, the network density is equal to the reference density D_{ref} for all experiments. The network diameter obtained with 64 nodes is taken as a reference and noted d_{ref}. We also use 16, 32, 64, 128 and 256 nodes, but in fields length respectively of 500, 707, 1000, 1414 and 2000 meters in order to maintain a constant network density. We thus handle topologies with a diameter of 1/2, $1/\sqrt{2}$,1,$\sqrt{2}$ and 2 times d_{ref}.

We generate 50 different topologies for each network density and diameter. We then randomly choose 10 pairs of connected nodes for each of them.

The average percentage of connected nodes pairs for both sets of experiments is depicted in figure 3.7(a). The number of nodes employed in the first set of experiments determines the network density. This has a strong impact on the global average connectivity. Above the reference density, all



(a) Connectivity



(b) Average path length

Figure 3.7: Static tests: Connectivity and path length characteristics versus number of nodes

Table 3.2: Static tests two simulation sets

	Set 1	Set 2
Network density (D_{ref})	0.25, 0.5, 1, 2, 4	1
Network diameter (d_{ref})	1	$1/2$, $1/\sqrt{2}$, 1, $\sqrt{2}$, 2
Overlay density (%)	12.5, 25, 37.5, 50, 62.5, 75, 87.5, 100	

pairs of nodes are connected. At half of the reference density, about 80 % of the pairs are connected and dividing again the density by two leads to a very poor connectivity of about 35 %. The first set thus allows us to study the protocol in sparse and dense networks. In the second set of experiments, the network density is constant and the figure indicates a global connectivity almost equal to 100 % at all network diameters.

The path length characteristic of the two sets is drawn in figure 3.7(b). The network diameter strongly affects the global average optimal path length, while the network density has almost no effect on it. For the first set, the path length drops in the left part of the figure because we chose connected pairs of nodes. Connected nodes are seldomly far away from each other in very sparse networks.

The parameters of the two sets of experiments employed in the static study are summarised in table 3.2. The graphs presented below are divided following these sets; we analyse the network density and the network diameter influence on performance at various overlay densities. The values assigned to the overlay density are 12.5, 25, 37.5, 50, 62.5, 75, 87.5 and 100 percent. As there are fifty scenarios and ten connections used at each network density and diameter, all the points presented are an average on five hundred runs.

At the beginning of each simulation, the source tries to send one UDP datagram containing a 512 bytes data payload. Simulations are stopped after 50 seconds. If the packet has been received, the simulation is said to be *successful*.

Static tests goals

In our reactive approach of overlay routing, there is no neighbour probe before a route is needed. Overlay neighbours are implicitly detected during route discoveries. As long as the probability of finding a route composed uniquely of overlay nodes is high, it is sufficient to use all physical neighbours that are overlay members as next hops. The Time To Live field of the ORREQ packet, the neighbourhood range (NR), can be set to one. The physical neighbours belonging to the overlay eventually resend the ORREQs, while other ones simply drop the overlay control messages. However, if the network or overlay density is not high enough, a unitary neighbourhood range may not be sufficient for finding overlay paths. In the static study, we run each simulation with an increasing neighbourhood range until it becomes successful and then log the performance obtained: Path length, control traffic and delivery delay. We thus show the overlay routing application performance in the ideal case under the Reactive Overlay Approach, i.e. when the neighbourhood range used is the minimum one allowing to find a route between the source and destination nodes. The network characteristics, density and diameter, and the overlay density effects on performance are pointed out. We take the AODV protocol as a reference for evaluating the overlay routing application.

Control traffic

The number of control packets needed to send the first data packet is often referred to as the *normalised routing load*.

We include in the control traffic of the overlay routing application all AODV packets emitted to build underlay routes between successive overlay nodes.

Figure 3.8 shows the overhead normalised routing load, which we define as the ratio between the Overlay-AODV and (native) AODV normalised routing loads. The normalised routing load of AODV, taken as reference for evaluating the overlay routing application, corresponds to the control traffic employed by AODV to establish the communication in a pure underlay network and, given a topology and a source-destination pair, is the same at all overlay densities.

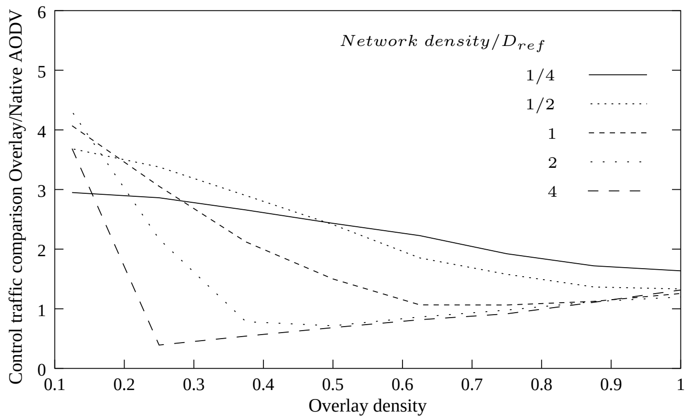
At high and moderate network densities, a decrease of overlay density first helps reduce the control traffic by performing a kind of natural gossiping [LHH02]. Then, a reduction of overlay density increases the control traffic because longest neighbourhood ranges are required for finding an overlay route. The control traffic sensitivity to the overlay density is stronger at high network densities. At the lowest network density, a decrease in overlay density does not help because the number of rebroadcast ORREQ is low, even if all the network nodes belong to the overlay.

The distance between the source and destination is the most determinant factor increasing the absolute normalised routing load, for the overlay routing application as for AODV. However, it does not influence a lot the overhead normalised routing load.

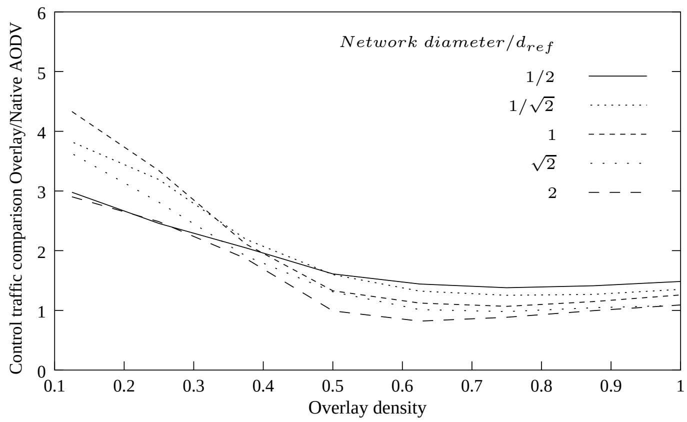
Path length

As the overlay density decreases, the overlay route followed by the packet may deviate from the shortest underlay path.

The overlay routing application as well as AODV may build routes a

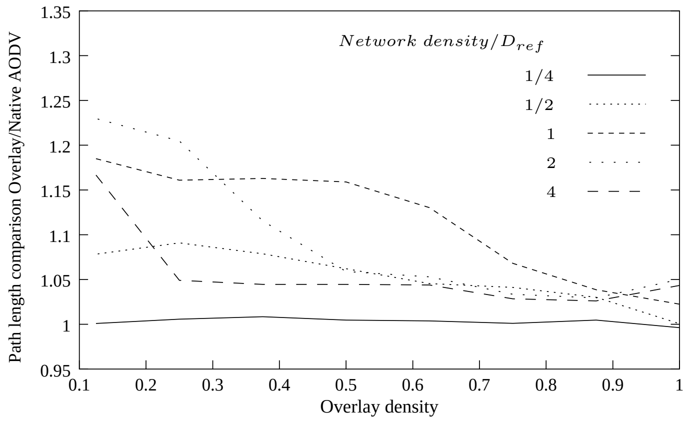


(a) Varying network density

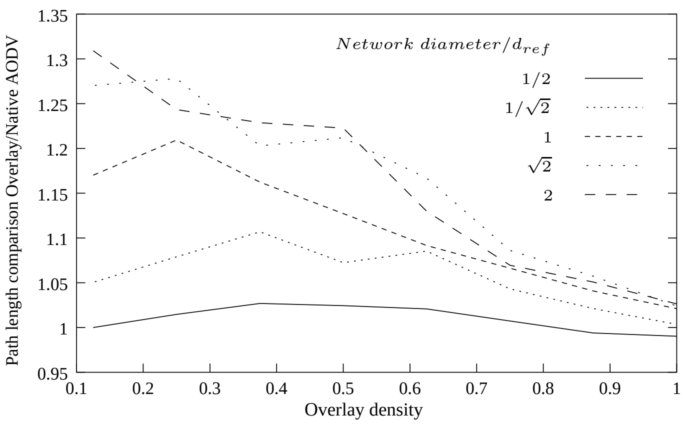


(b) Varying network diameter

Figure 3.8: Static study: Overlay/Native AODV control traffic comparison



(a) Varying network density



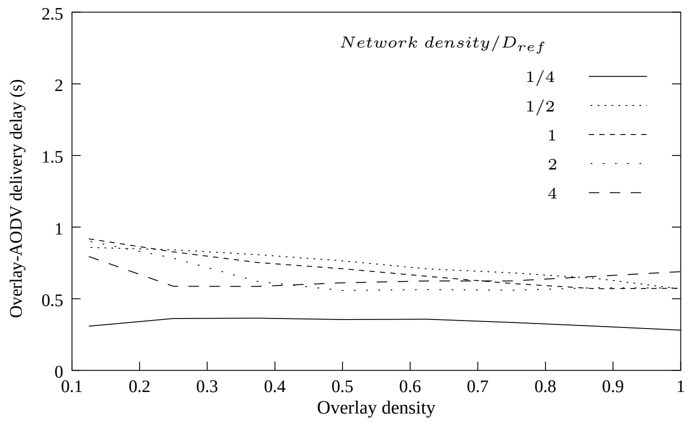
(b) Varying network diameter

Figure 3.9: Static study: Overlay/Native AODV average path length comparison

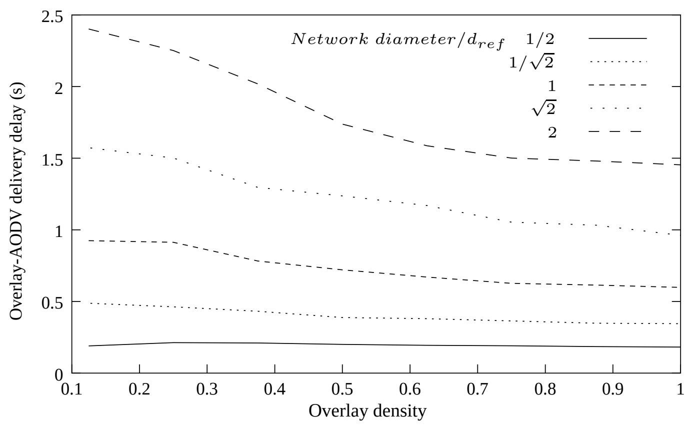
little longer than the optimal one, respectively between the source and destination overlay nodes, and between successive overlay nodes on the overlay route. This sub-optimality strengthens when the distance between source and destination gets long.

These two effects are shown by figure 3.9, where the ratio of the average path lengths obtained with the reactive overlay application and AODV is drawn. The overhead average path length induced by the overlay routing application over AODV seldomly exceeds 30 %.

Delivery delay



(a) Varying network density



(b) Varying network diameter

Figure 3.10: Static study: Overlay-AODV delivery delay

Table 3.3: Dynamic tests common parameters

Simulation Parameter	Value
Simulator	ns-2.26
Data Packet Size	512 bytes payload
Node Max. IFQ length	50
Packet rate	4 per second
Nodes transmitting range	250 m
Field length	1000 m
Number of nodes	64
Number of connections	1
Simulation duration	600 s
Number of trials	20

The last performance parameter studied is the delivery delay. This can be divided into two parts: the time necessary for a node to find a route and for the packet to progress on this route, which we respectively call the *route discovery delay* and the *path delay*. Because there is only one overlay message sent, the route discovery delay constitutes the main part of the total transmission delay in the static study.

The network diameter has a big influence on the AODV delivery delay. As shown in figure 3.10, this is also the case for the overlay routing application. There are two reasons for it. Firstly, the network diameter increases the average number of ring searches. Secondly, it raises the average control traffic which in turn lengthens the contention delay during the route requests propagation. When the path length is constant, the variation of the delivery delay is weaker but we can observe that it has the same shape as the overhead control traffic. In summary, the total delivery delay in the static study is very close to the average route discovery delay, which depends on the network diameter and on the amount of control traffic.

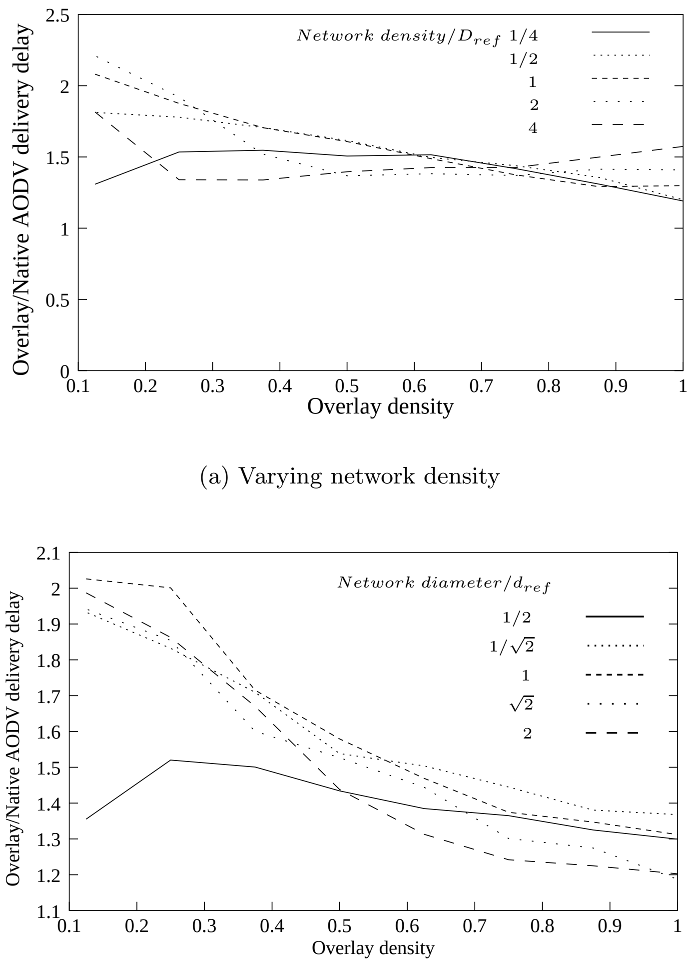
A comparison between the total delays obtained with the overlay routing application and AODV is shown on Figure 3.11. In the static study, the overhead delay increases from 25 to 125 % when the overlay density diminishes, independently of the underlay topology characteristics.

3.4.2 Dynamic networks

Dynamic tests description

The common parameters for all dynamic simulations are listed in table 3.3.

We now work at the reference geographical density, vary the overlay density and the mobility level. As for the static tests, we explore overlay



(a) Varying network density

(b) Varying network diameter

Figure 3.11: Static study: Overlay/Native AODV delivery delay comparison

densities of 12.5, 25, 37.5, 50, 62.5, 75, 87.5 and 100 percent. We generate 10 different topologies for each mobility level. We also use one source-destination pair per simulation in order to conduct our analysis in an uncongested environment. For mobility, we apply the random waypoint mobility model [JM96]. The simulations duration is 600 seconds. Each node is at rest at the beginning of the simulation for *pausetime* seconds, then chooses a random destination on the field and moves to it with a randomly and uniformly chosen speed in the range of 0 to 20 m/sec. When it has reached its destination, it pauses again for *pausetime* seconds, then moves again, and so on. We vary the pause time to simulate different mobility levels : 0, 30 , 60, 120, 300 and 600 seconds. A zero pause time indicates that nodes are continuously moving while a 600-second pause time means that nodes are at rest for the entire simulation duration. We randomly chose 2 pairs (src_i,dst_i) in each topology as test connections. As there are ten scenarios and two connections used at each mobility level, all the points presented in the graphs below are an average over twenty runs.

In this environment, physical as well as overlay links can break. AODV and the overlay routing application both operate in the link break detection mode.

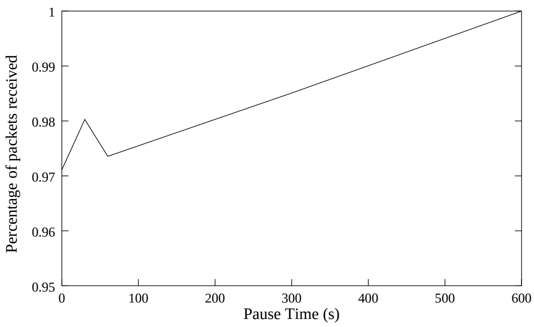
Dynamic tests goals

In the dynamic tests, for each simulation, we set the neighbourhood range value a priori. This means that overlay routes may not be found, even if the chosen source and the destination are connected at the underlay level. A source which does not find a route after three consecutive full ring searches separated by ten seconds intervals, estimates that the network conditions are bad and stops trying to send data. The simulation ends and the performance are logged. In this situation, the control traffic is high and the percentage of packets received low. If a route is always found, the simulation is stopped after 600 seconds and the performance logged. This means that the control traffic and delay overhead costs observed in the dynamic study are higher than the ideal ones analysed with the static study.

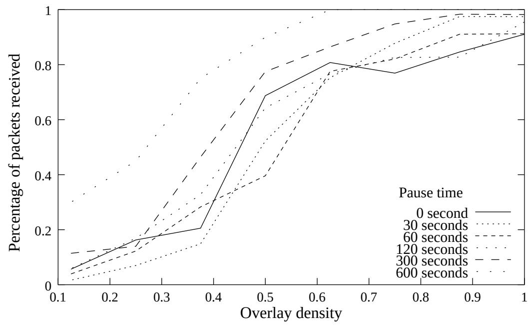
In the dynamic study, we first investigate the qualitative effect of mobility on the overlay routing application. We then analyse the impact of a bad neighbourhood range choice on performance by comparing the results obtained for a pause time of 600 seconds to the ones obtained in the static study. We finally point out the overlay density effect on performance once again.

Percentage of packets received

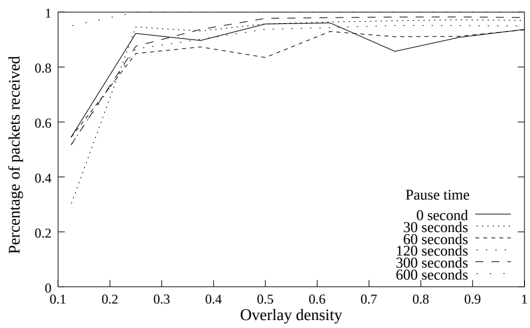
Figure 3.12 compares the percentage of packets received using AODV and the overlay routing application with a neighbourhood range set to 1, 2 and



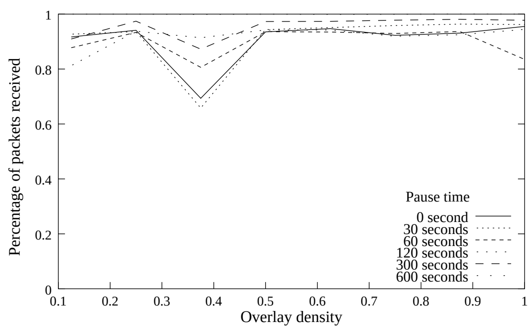
(a) AODV



(b) Overlay-AODV, NR=1



(c) Overlay-AODV, NR=2



(d) Overlay-AODV, NR=3

Figure 3.12: Dynamic study: AODV and Overlay-AODV delivery percentage

3 hops.

In each graph, the curve at a 600-second pause time, related to static nodes, can be interpreted as an evaluation of the connectivity at the overlay level. At the reference geographical density, the probability of finding a underlay route between the source and destination nodes is very high. This is not the case for overlay routes. An overlay route can be found only if there is a underlay path between the source and destination nodes on which overlay nodes are separated by a number of hops lower or equal to the neighbourhood range value. If the overlay density or the neighbourhood range is not high enough, an overlay route may not exist.

With a unitary neighbourhood range, if the overlay density is below 60 %, the connectivity at the overlay level drops and the percentage of overlay messages received is very low, compared to AODV. This shows that a unitary neighbourhood range would not be a pertinent choice for the throughput in a large panel of network and overlay densities, and consequently that an overlay approach is relevant. We will not show the results obtained with a unitary neighbourhood range for other performance analyses.

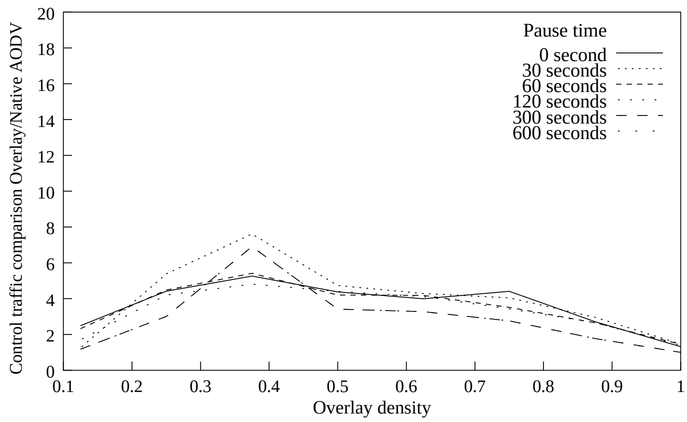
With a neighbourhood range value of two, except for the lowest overlay density, the overlay network is fully connected and the percentage of messages received is above 80 %, at all mobility rates. In a pure overlay network, the percentage of packets delivered by the overlay routing application varies from 100 to 94, instead of 97 with AODV alone. This difference is due to longer route breakage detection time and periods during which an overlay route cannot be found. When the overlay density decreases, the opportunity of finding an overlay route also does and the percentage of packets received is more affected by frequent route breakages than in networks entirely composed of overlay nodes.

At the lowest overlay density, a neighbourhood range value of three improves a lot the percentage of messages received. However, a longer neighbourhood range does not only increase the connectivity at the overlay level, it also results in bigger control traffic and extended route discovery delays during which some messages may be dropped from the route waiting queue. For overlay densities between 25 and 50 %, the throughput obtained with a neighbourhood range of only two is better.

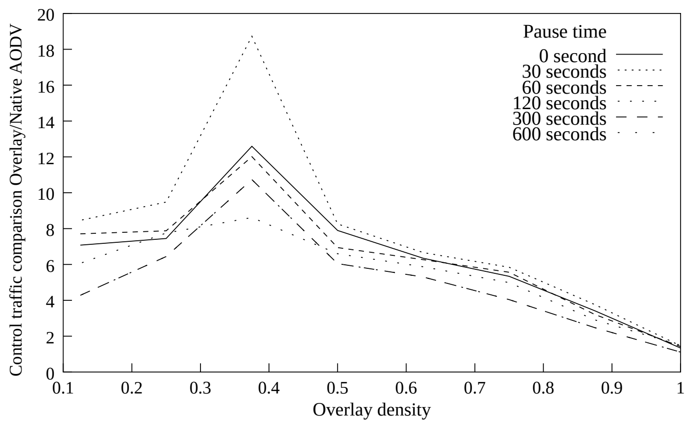
We can also observe that, as for AODV, the overlay routing application throughput is first damaged by the quantity of moving nodes, then the mobility helps find routes between source and destination. This is particularly observable at low overlay densities, because the connectivity at the overlay level is low.

In summary, the quantity of mobility affects more the overlay routing application than AODV. However, if the neighbourhood range value is adapted to the network conditions, the throughput is over 80 % in all cases.

Control traffic



(a) NR=2



(b) NR=3

Figure 3.13: Dynamic study: Overlay/Native AODV control traffic comparison

Figure 3.13 shows the ratio of the control traffic induced by the overlay and native AODV protocols.

The overhead control traffic is not high when the overlay density is very high or very low. However, for intermediate overlay densities, it can be very big. In network where all nodes are members of the overlay, whatever the mobility rate, the control traffic induced by the overlay routing application is similar to the AODV one because the diffusion control of the broadcast packets and messages is the same. In a very sparse overlay network, the propagation of the broadcast messages, embedded in broadcast packets, is controlled by the underlay nodes. The IP Time To Live field of the packets carrying the ORREQs reaches zero before an overlay node receives and re-broadcasts them. The control traffic is thus low but the received data percentage obtained is weak. For intermediate overlay densities, the ORREQs messages are received by overlay nodes. When an overlay routing control message is rebroadcast, it is embedded in a new IP packet and the neighbouring underlay nodes do not drop it, even if they already have received an IP packet containing exactly the same overlay routing control message. Useless overlay route requests are thus propagated.

As expected, the control overhead obtained for motionless nodes is higher than in the static study. On one hand, if an overlay route search procedure ends unsuccessfully because the neighbourhood range is not large enough to enable communication, the source node starts a new expanding ring search after ten seconds. This is particularly prohibitive for static nodes, because the sources make three successive unsuccessful route searches before the simulation gets stopped. On the other hand, if the neighbourhood range used is larger than the optimum one, the control traffic can also increase because of the propagation of unnecessary overlay route requests. This explains why the control overhead obtained is doubled when using an overlay range value of three instead of two and why it reaches the highest values for overlay densities between 25 and 50 %.

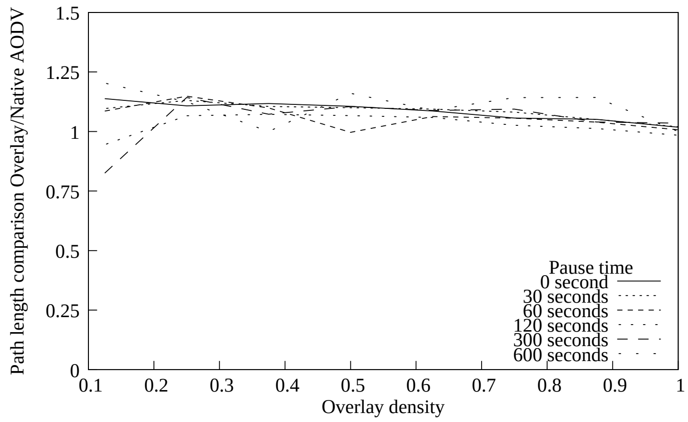
If the overlay density and the mobility are high enough to allow communication most of the time, the control traffic overhead is almost constant for all pause times, indicating that the mobility does not affect the overlay routing application much more than AODV in terms of control traffic amount.

Path length

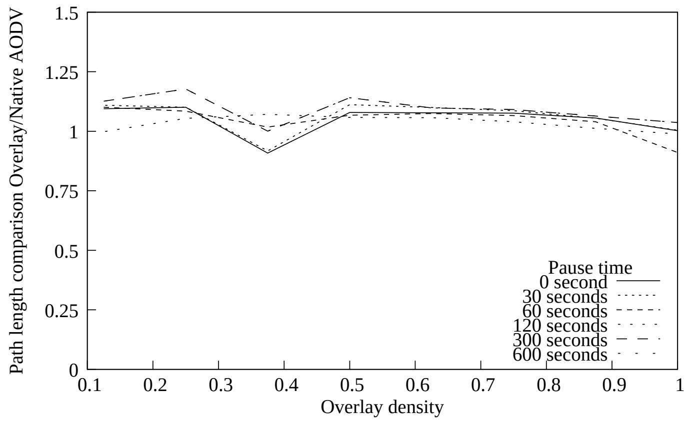
We logged the number of hops an overlay message followed before reaching the destination node.

The results, presented on Figure 3.14, must be analysed carefully. When the overlay density and pause time values correspond to a low delivery percentage, the overlay path stretch is under-estimated, because messages are more likely to be received when the source and destination nodes are close from each other.

However, when the delivery percentage is good, we can draw the same conclusion as for the static study. At reference network geographical density and diameter, the overlay routing application creates a little path overhead compared to AODV, which does not become greater than 25 %.

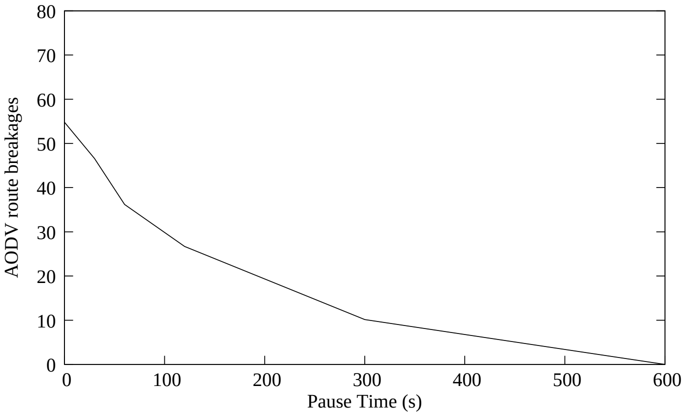


(a) NR = 2

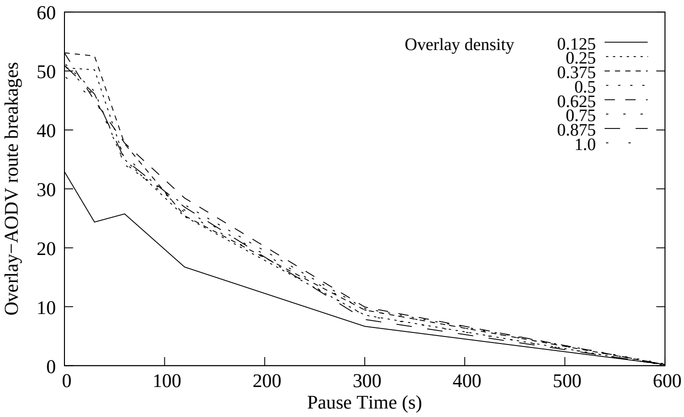


(b) NR = 3

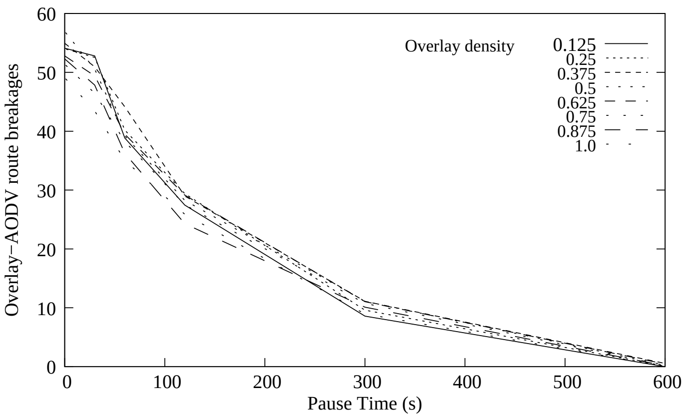
Figure 3.14: Dynamic study: Overlay/Native AODV path length comparison



(a) Native AODV



(b) Overlay routing application - NR = 2



(c) Overlay routing application - NR = 3

Figure 3.15: Dynamic study: AODV and Overlay-AODV routes stability

Routes stability

Figure 3.15 shows the number of breakages observed during the simulations. Their amount obviously increases with the degree of mobility. For a given pause time, because the path length is very low, the route stability is quite the same for AODV and for the overlay routing application, at all overlay density and neighbourhood range.

The only exception occurs with the lowest overlay density and neighbourhood range because many route requests fail. In this case, there are a few routes built. Consequently less route breakages are observed.

Delivery delay

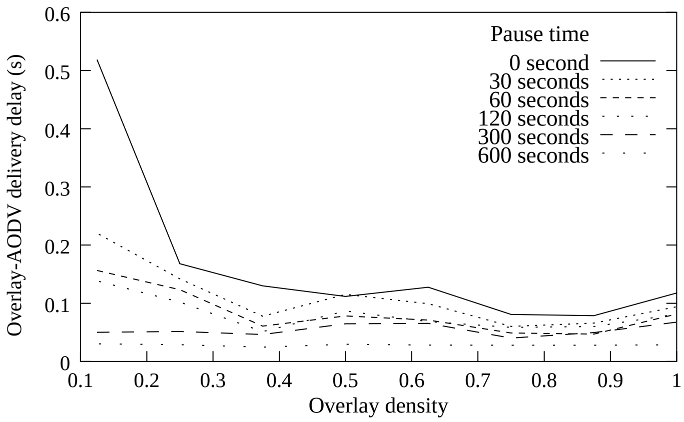
In Fig. 3.16, the delivery delay values represented are averaged on all messages received. Note that these results must not be compared to the ones obtained in the static study, and drawn on Fig. 3.10. As stated above, the route discovery delay in a static network constitutes the main part of the delivery delay. The situation is very different in a mobile network. Most of the time, when the overlay constant bit rate application has data to send, the route the message will follow is already known. The route discovery delay is null and the total delay reduces to its path component.

With these tests, the path delay is almost constant because the geographical density and consequently the path length are invariant, and because there is no congestion. Even if the route discovery delay is null most of the time, its value influences the average total delivery delay because of the order of magnitude existing between route discovery durations and path delays. It is the route discovery component of the total delivery delay which varies with the overlay density and the mobility level.

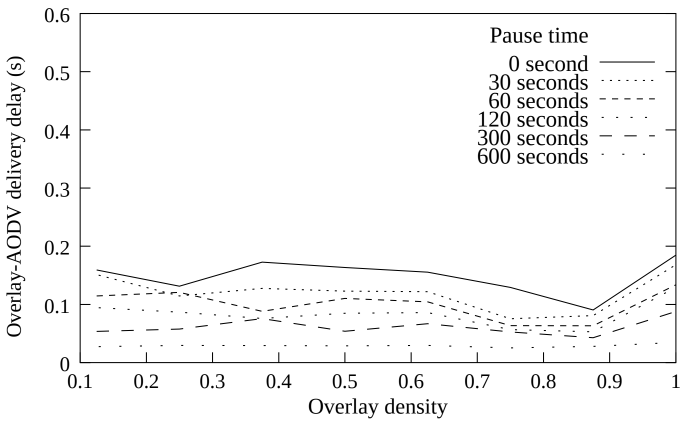
The longer delays observed when the pause time lessens is due to the increase of route breakages. When the source is informed that a route is no more valid, it stores the next overlay messages in a routing queue until a new route is discovered. This increases their route discovery delay. This is particularly visible in the experience with a neighbourhood range of two, at lowest overlay density and highest mobility level. In this area, a neighbourhood range of two is often insufficient to find an overlay route. Moreover, once a route is discovered, its lifetime is limited. Many of the received overlay messages are sent in burst from the routing queue. Their route discovery delay is long and so is their path delay because they contend with each other on the path to the destination[2]. With a neighbourhood range of three, the average routing queue length is lower because the source and the destination are more often connected at the overlay level.

In a large panel of densities and mobility conditions, the total delivery delay, compared to AODV, is about two times longer with the overlay rout-

[2]The capacity of a chain of node is studied in [LBD$^+$01]



(a) NR=2



(b) NR=3

Figure 3.16: Dynamic study: Overlay-AODV delivery delay

ing application, for both neighbourhood range values. If the overlay density is very low and the mobility level high, it can be three to four times longer. If the neighbourhood range value is appropriate, it never exceeds four times the AODV delay.

3.5 Conclusion

Our starting hypothesis is that applying a customised routing strategy could be profitable to a group of ad hoc users.

We adopted an overlay approach. This allows a transparent coexistence between the overlay members and other network nodes, without the use of a new packet header nor any modification of the ad hoc routing protocol used by all nodes.

We introduced the notion of neighbourhood range, defined as the maximum number of hops between two neighbouring overlay nodes. Its value implicitly defines the overlay structure at any time. Consequently, the overlay nodes can run a reactive overlay routing application without discovering a set of overlay neighbours first.

We defined an elementary reactive overlay routing application and tested it in a variety of conditions, including the network and overlay densities.

The reactive overlay routing application has been developed with the following objectives in view :

- dynamic reactive routing between overlay nodes
- transparent coexistence of overlay members and other network nodes
- operation in geographically dense and sparse networks
- operation with high and low ratio of overlay nodes in the network

Only two architectural modifications are required to the conventional model of ad hoc nodes. Firstly, in order to limit the control traffic overhead, the ad hoc nodes do not forward a broadcast packet if they run an application listening to the type of message encapsulated in the packet. In other words, the overlay broadcast messages must be grabbed by their listening application. Secondly, the overlay link break detection mode is possible if and only if AODV notifies the overlay routing application when it detects the breakage of an active route. This is the only modification we impose to AODV, and it is necessary only in this overlay routing mode.

We observed that the neighbourhood range used has a crucial impact on the throughput and on the amount of control traffic. If it is too low, communication between overlay nodes can be impossible, even if the network is connected. If it is too large, there is a big waste of control messages.

However, when the neighbourhood range used is appropriate, the tested reactive overlay routing application achieves good performance, even if the overlay density is no more than 12.5 % and the mobility level important.

Chapter 4

The Critical Neighbourhood Range Theoretical Study

4.1 Study motivation and overview

Consider an ad hoc network and a given community, i.e. a subset of its nodes that are susceptible to communicate more together than with outside nodes. Assume that this community wishes to organise itself as an overlay. Overlay routing favours routes passing through numerous community members. These may be longer, to a limited extent, than the shortest path, but there is a potential gain each time a packet traverses a community member on its way to the destination.

In previous chapter, we defined a reactive overlay routing protocol, Overlay-AODV, using the reactive overlay approach. Simulations confirmed that, on ad hoc networks, an overlay routing protocol can work without pre-establishing the overlay topology. The performance study evidenced the impact of using an appropriate value for the *neighbourhood range*, defined as the maximum number of hops between two overlay neighbours.

The reactive overlay approach is made possible by the fact that, at any time, the neighbourhood range implicitly defines the overlay topology. In this framework, we can expect, as for Overlay-AODV, that the efficiency of many overlay routing protocols will greatly depend on whether the right neighbourhood range value is used or not.

The neighbourhood range must be sufficiently high to obtain a connected overlay but as low as possible to limit the amount of messages generated in the network by overlay nodes communication.

The **critical neighbourhood range problem** consists of determining the minimum neighbourhood range value that generates a connected overlay. We adopt this terminology because of its obvious resemblance with the critical transmission range problem, presented in Section. 2.2.1.

We first demonstrate that in connected networks, as the network gets denser or larger, the shortest path between any pair of nodes draws close to the straight line. This property, that we call the asymptotic path length theorem, helps us to demonstrate the necessary and/or sufficient conditions on the neighbourhood range to achieve asymptotic connectivity of the overlay almost surely, i.e. connectivity when the number of nodes in the underlay tends to infinity (so-called dense networks) or when the size of the field tends to infinity (so-called sparse networks).

The main reason for addressing asymptotic connectivity is its mathematical tractability. We build on several asymptotic results on basic geometric graphs to derive properties of the overlays. Our asymptotic results can be seen as approximations of finite (real) networks either when the number of nodes (resp. the field) is large enough. Nevertheless our mathematical conditions already shed some light on the relation linking the number of nodes, the field size, the radio transmission range, the overlay density and the neighbourhood range to get a connected overlay.

We do take into account the potential use of a homogeneous topology control algorithm at the underlay level and allow the overlay density to evolve with the network size. In particular, if the overlay density diminishes, our results show how a compensation in neighbourhood range can keep the overlay still connected. They also point out that a more efficient topology control algorithm of an ad hoc network will require more traffic for the use and maintenance of overlays built on it.

In next section, we precisely define the problem studied. We then present analytical results, respectively for dense and sparse networks in Sec. 4.3 and 4.4, and discuss some of their practical implications.

4.2 Problem Definition

We are interested in the asymptotic connectivity of overlay graphs built over asymptotically almost surely (a.a.s.) connected basic graphs.

These notions are defined in the following paragraphs. We then close this section with a discussion on the implicit assumptions we make in the problem and model specification.

4.2.1 Basic and Overlay Graphs

Consider an ad hoc network of n nodes, deployed over a square field of length ℓ, and where each node is assigned a transmission range of length r, the unit used for measuring r and ℓ being identical. This network is modelled by a random geometric graph denoted $g(n, r, \ell)$ which has the following properties. The vertices of g are uniformly and independently distributed on a square of size $\ell \times \ell$. They can either have been disseminated following the uniform distribution of n points or by a spatial homogeneous Poisson point process of mean n. There exists an edge between each pair of vertices if and only if the Euclidean distance between them is not greater than r. An example of basic graph is given by Fig. 4.1(a).

Let then $g(n, r, \ell)$ be a connected graph, D be a real number with $0 \leq D \leq 1$ and R be an integer with $R \geq 1$. An overlay graph $G(n, r, \ell, D, R)$ denotes a graph with the following properties. The D parameter represents the overlay nodes density. The number of vertices of G equals a proportion D of the number of vertices of g. These are randomly and uniformly selected in the vertices set of g, which is called its basic graph. For example, on Fig. 4.1(b), the vertices of the overlay graph, grey-shaded, have been selected out of the 20 vertices of the basic graph given by Fig. 4.1(a) with a density of 25%.

The R parameter symbolises the neighbourhood range. There exists an edge between a pair of vertices (v_1, v_2) if and only if the shortest path in g from v_1 to v_2 contains less than or exactly R hops. For example, the third subfigure of Fig. 4.1 shows the shortest paths between the vertices of the overlay graph, and the last three subfigures represent the three overlay graphs obtained respectively with $R = 1$, $R = 2$ and $R = 3$.

In the following, in conjunction with the ad hoc networks terminology, the vertices of an overlay graph will be referred to as overlay nodes and the vertices of its basic graph as nodes.

4.2.2 Asymptotic Connectivity

We use two models for studying the asymptotic connectivity of random geometric graphs, that we qualify as dense and sparse.

Dense Model

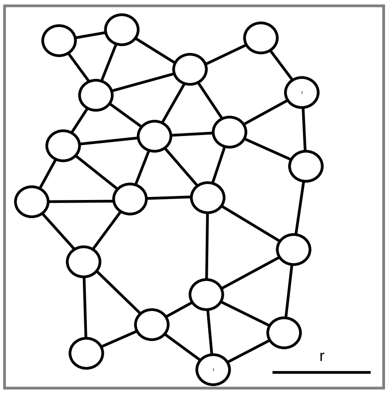
Definition The field length ℓ is a constant. All other parameters (r,D and R) are functions of the number of nodes. For example, $r(n)$ can be decreasing when n increases, which is a desired behaviour for minimising the capacity loss due to interferences.

Notations In this context, the ℓ parameter is assumed to be set to one and can be omitted in the notations. A basic graph can be denoted by $g(n, r(n))$ and an overlay graph by $G(n, r(n), D(n), R(n))$ or $G(g, D(n), R(n))$.
We may generally simply write $g(n, r)$, $G(n, r, D, R)$ or, if $g(n, r)$ is given, $G(g, D, R)$.

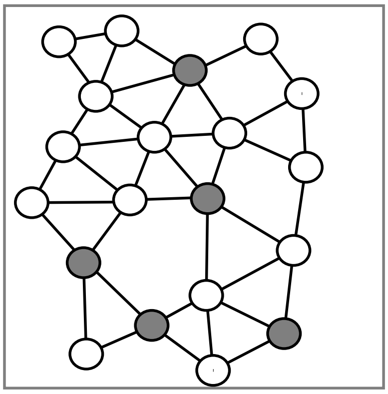
Asymptotic Connectivity A dense graph is connected asymptotically almost surely if and only if the probability that it is connected tends to one as its *number of vertices* tends to infinity.

Dense graph $\mathcal{G}$ is connected a.a.s.

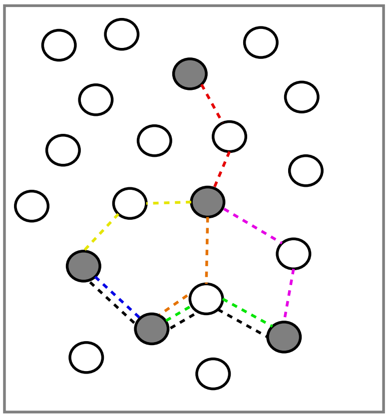
$$\Longleftrightarrow$$



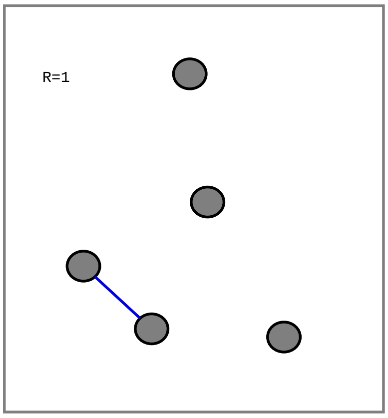
(a) A connected basic graph with $n = 20$, r given on the graphic



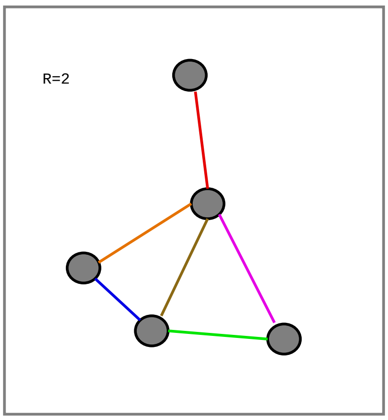
(b) Overlay nodes distributed with $D = 0.25$



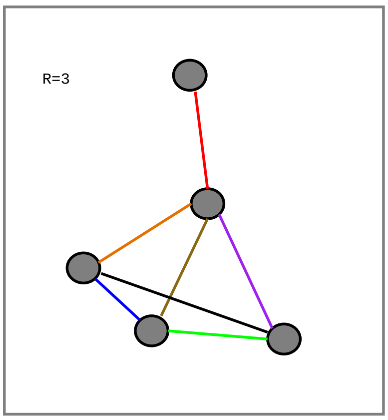
(c) Shortest paths between each pair of overlay nodes



(d) Overlay graph obtained with $R = 1$ (not connected)



(e) Overlay graph obtained with $R = 2$ (connected)



(f) Overlay graph obtained with $R = 3$ (connected)

Figure 4.1: Example of a basic graph and of overlay graphs obtained on it with $D = 0.25$, and with increasing values assigned to R

$\lim_{n\to\infty}$ P[$\mathcal{G}$ is connected]=1.

Note that for overlay graphs, the vertices are the overlay nodes. This means that $D(n)$ must be such that $\lim_{n\to\infty} D(n)n = +\infty$.

Use This model is only suited for studying the asymptotic behaviour of dense networks because the vertices density $\frac{n}{\ell^2}$ tends to infinity as n does.

Sparse Model

Definition All parameters (n,r,D and R) are functions of the field length ℓ. This was the model used in [SB03] to study the critical transmission range in sparse networks.

Notations A basic graph should be denoted by $g(n(\ell), r(\ell), \ell)$ and an overlay graph by $G(n(\ell), r(\ell), \ell, D(\ell), R(\ell))$ or $G(g, D(\ell), R(\ell))$. We may also simply write $g(n, r, \ell)$, $G(n, r, \ell, D, R)$ or, if $g(n, r, \ell)$ is given, $G(g, D, R)$.

Asymptotic Connectivity A sparse graph is connected asymptotically almost surely if and only if the probability that it is connected tends to one as the *field length* tends to infinity.

Sparse graph $\mathcal{G}$ is connected a.a.s.

$\Longleftrightarrow$

$\lim_{\ell\to\infty}$ P[$\mathcal{G}$ is connected]=1.

Use In this context, the node density $\frac{n}{l^2}$ might either converge to 0, or to a constant $c > 0$, or diverge as the size of the deployment region grows to infinity, depending on the relative values of r, n, and l. This model is thus more general than the dense one and suited for studying the asymptotic behaviour of sparse as well as of dense networks.

Notice that the sparse appellation, which was used to present this model in [SB03], could be a little confusing. However, as exposed in the following, the theorems based on the dense model are more convenient than the ones based on the sparse one. For this reason, the latter should only be used to analyse sparse networks. This justifies the model denomination.

4.2.3 Problem and model discussion

Connected basic graph

We consider only connected basic graphs. This seems reasonable to us as a disconnected basic graph will not provide connected overlays, whatever the neighbourhood range is, unless all the overlay nodes are concentrated in a connected part of it.

Asymptotic study

Many asymptotic properties of random geometric graphs have been demonstrated. In particular, we mentioned in Section 2.2.1 several studies of the asymptotic connectivity of ad hoc networks, while the connectivity probability of a finite network, because of its complexity, has been the subject of very few analytical studies.

Homogeneous transmission range assignment

The transmission range is represented as a function of the number of nodes, directly in the dense case and indirectly, via its dependence to ℓ, in the sparse case. This allows us to model a possible topology control protocol running on the ad hoc network, which would reasonably reduce the transmission range as the number of nodes increases, in order to conserve energy and global network capacity. We however implicitly limit ourselves to homogeneous topology control protocols, i.e. protocols which assign the same transmission range to all nodes.

This assumption greatly simplifies further mathematical developments and seems realistic in the context of our study. A common transmission range at each node provides some appealing features, that can be consulted in [KK05], such as the creation of bidirectional links only. Moreover, it is shown in [GK00] that, under a homogeneous spatial distribution, choosing a common transmission range can decrease capacity at most by a factor of $\sqrt{\ln n}$, where n is the number of nodes, in comparison to allowing the flexibility of a different power level for each packet at each node [KK05]. This means that asymptotically a common power is nearly optimal in terms of network capacity [NKSK02]. Finally, as we use a uniform distribution of nodes and study an asymptotic property, more sophisticated topology control algorithms would intuitively lead to transmission range values converging in probability to a common function $r(n)$. All these reasons make us believe that a homogeneous transmission range assignment is both general and adapted.

4.3 Dense Networks

4.3.1 Known Results on Basic Graphs

Consider a basic graph $g(n, r)$. Let us build a graph $g\prime(n, r\prime)$ that has the same nodes set as g and such that there is an edge between every pair of nodes. Let M_n denote the longest edge length of the minimal spanning tree built on $g\prime$. In [Pen97], Penrose demonstrated that the graph $g(n, r)$ is connected if and only if $r \geq M_n$ and [1]

$$\forall \alpha \in R: \lim_{n \to +\infty} P[n\pi {M_n}^2 - \ln n \leq \alpha] = \exp(-e^{-\alpha}) \quad (4.1)$$

This implies directly the following theorem.

Theorem 4.3.1 *(Asymptotic connectivity of dense basic graphs)*
A graph $g(n, r)$ with

$$\pi r^2 = \frac{\ln n + k(n)}{n}$$

is connected a.a.s. if and only if $\lim_{n \to +\infty} k(n) = +\infty$.

The same result was demonstrated by Gupta and Kumar for a uniform distribution of nodes over the unit disk [GK99].

Note that for dense networks, a unique condition has been demonstrated to be both sufficient (when it is fulfilled, the graph is a.a.s. connected) and necessary (when it is not fulfilled, the graph is not a.a.s. connected).

This theorem is illustrated by Figure 4.2. On the upper sub-figure, we drew the function $r(n) = \sqrt{c \frac{\ln n}{\pi n}}$ for different values of parameter c. This corresponds to $k(n) = (c-1)\ln n$ in the theorem. On the lower sub-figure, we drew the percentage of connected dense basic graphs $g(n, r(n))$ obtained on 200 experiments with 400, 800 and 1200 nodes. For all $c > 1$, we have $k(n) \to +\infty$ and observe that the connectivity probability increases with the number of nodes. We can presume that it converges to 1 when the number of nodes grows to infinity. For $c = 1$, we have $k = 0$ and observe that the connectivity probability does not increase. We also conducted experiments with lower values (0.5, 0.75 and 0.9) of parameter c but did not draw the corresponding curves because none of them provided any connected graph.

4.3.2 Minimal Neighbourhood Range

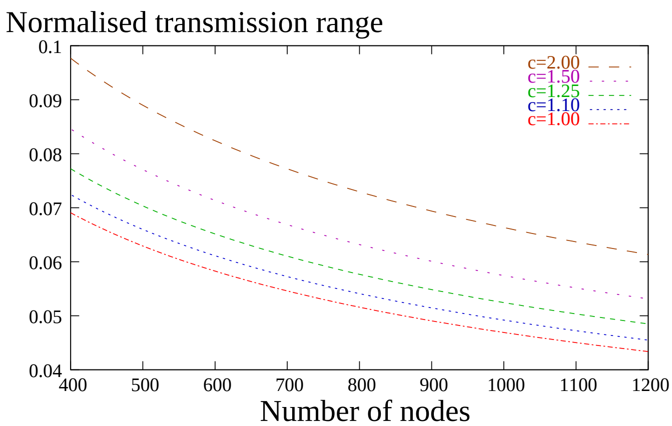
Theorem 4.3.2 *(Necessary condition for the asymptotic connectivity of dense overlay graphs)*
An overlay graph $G(n, r, D, R)$ with

$$\pi(Rr)^2 = \frac{\ln(\lceil Dn \rceil) + K(n)}{\lceil Dn \rceil} \quad (4.2)$$

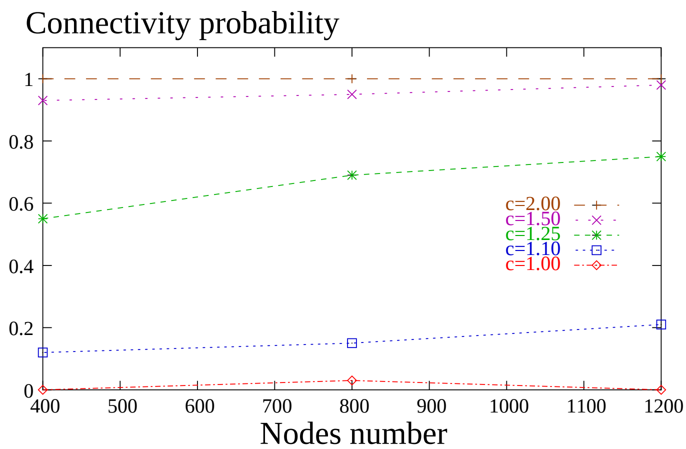
is not a.a.s. connected if $\lim_{n \to +\infty} K(n) \neq +\infty$.

Proof: Let $G(n, r, D, R)$ be an overlay graph. Consider a graph $g\prime(\lceil Dn \rceil, Rr)$ such that the vertices sets of G and $g\prime$ are identical. By definition, if there exists an edge in G between two vertices v_1 and v_2 then the

[1] Note that the theorem in [Pen97] is more general. We isolated here the results that are of direct interest to us.



(a) Communication range



(b) Connectivity probability

Figure 4.2: Relationship between the communication range and connectivity probability for dense graphs

shortest path between them contains less than or exactly R hops. As the distance between two consecutive nodes on a path cannot be longer than the transmission range r, the maximal distance between v_1 and v_2 is thus Rr and this edge also exists in $g\prime$. Consequently, the edges set of G is included in the edges set of $g\prime$. If $g\prime$ is not connected, then G neither is.

Applying Theorem 4.3.1 to a graph $g\prime(\lceil Dn \rceil, Rr)$, we obtain a necessary condition for the asymptotic connectivity of an overlay graph $G(n, r, D, R)$.

■

4.3.3 Sufficient Neighbourhood Range

We start with the following lemma.

Lemma 4.3.3 *Let X_S be a random variable designating the number of nodes on a surface S with $0 \leq S \leq 1$. For the uniform distribution of n nodes, as for the Poisson two-dimensional spatial distribution of mean n on the unitary square, $P[X_S = 0] \leq \exp(-nS)$.*

Proof: [Lemma 4.3.3]
We start with the uniform distribution.

If n nodes are distributed uniformly and independently on the unitary square, then the probability that a node lies on a surface $S \leq 1$ equals S.

Let X_S be a random variable designating the number of nodes on a surface S with $0 \leq S \leq 1$.

$$P[X_S = k] = S^k (1 - S)^{(n-k)}$$

Thus the probability that there is no node on S is

$$\begin{aligned} P[X_S = 0] &= (1 - S)^n \\ &= \exp(n \ln(1 - S)) \end{aligned}$$

with

$$\ln(1 - x) = -x(1 + \frac{x}{2} + \frac{x^2}{3} + \frac{x^3}{4} + ...)$$

$$\Rightarrow P[X_S = 0] = \exp(-nS(1 + \frac{S}{2} + \frac{S^2}{3} + \frac{S^3}{4} + ...)) \leq \exp(-nS)$$

Let us now focus on the Poisson distribution. The spatial Poisson point process has mean n.

The probability of having k nodes on a surface S is

$$P(X_S = k) = \frac{(nS)^k}{k!} \exp(-nS)$$

$$\Rightarrow P(X_S = 0) = \exp(-nS)$$

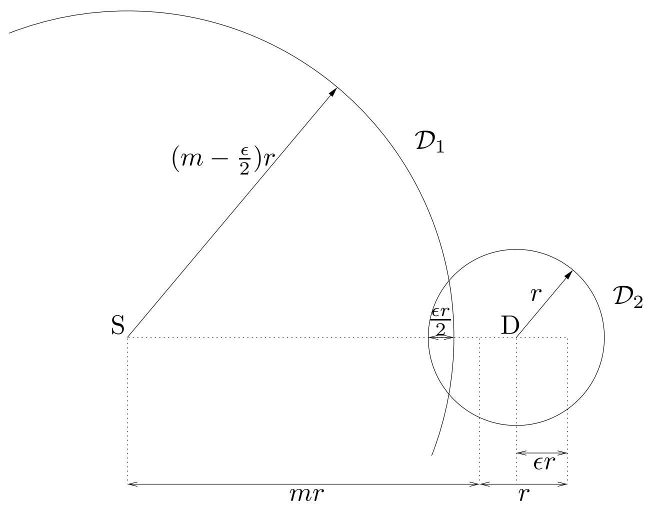


Figure 4.3: Drawing for the proof of the asymptotic path length theorem

Thus, for both distributions

$$P[X_S = 0] \leq \exp(-nS) \tag{4.3}$$

■

Exploiting this lemma, we can derive the following theorem on the asymptotic path length.

Theorem 4.3.4 *(Asymptotic path length)*
Let g be an a.a.s. connected graph and m be a strictly positive integer. Let n_1 and n_2 be two nodes of g. If the Euclidean distance between n_1 and n_2 is strictly less than mr, then there exists a.a.s. a path between them composed of less than or exactly m hops.

Proof: (the asymptotic path length theorem in the context of dense graphs)

We adopt an inductive approach over m. Assume that n_1 is located at point S and n_2 at D. If $m = 1$, then the Euclidean distance between S and D, denoted by $|SD|$, is strictly less than r. The nodes n_1 and n_2 are thus neighbours and there exists a path of one hop between them; the property is valid.

Let us now prove that if the property is valid for an integer m, then it is also valid for the integer $m + 1$.

Assume that $|SD| = (m + 1 - \epsilon)r$ with $m > 0$ and $0 < \epsilon \leq 1$.

Let us draw a disk $\mathcal{D}_1$ centered on S and of radius $(m-\frac{\epsilon}{2})r$ and another disk $\mathcal{D}_2$ centered on D and of radius r, as in Fig. 4.3. The disks have a non-empty intersection, that we denote $\mathcal{I}$.

If the three following conditions are all satisfied, then there is a path of at most $m+1$ hops between n_1 and n_2 :

1. there is a node n_i in $\mathcal{I}$,
2. there is a path of at most m hops between n_1 and n_i,
3. there is a path of length 1 between n_i and n_2.

In terms of probabilities, this can be written:
$P[path\_len(n_1, n_2) \leq m+1]$
$\geq P[some\, n_i\, in\, \mathcal{I}]$
$\times P[path\_len(n_1, n_i) \leq m \,|\, some\, n_i\, in\, \mathcal{I}]$
$\times P[path\_len(n_i, n_2) = 1 \,|\, some\, n_i\, in\, \mathcal{I}]$
By geometric construction, if there is a node in $\mathcal{I}$ then this node is a neighbour of n_2.
Thus the third probability equals one and, asymptotically, we have:

$$lim_{n\to\infty} P[path\_len(n_1, n_2) \leq m+1]$$

$$\geq lim_{n\to\infty} P[some\, n_i\, in\, \mathcal{I}] \times P[path\_len(n_1, n_i) \leq m \,|\, some\, n_i\, in\, \mathcal{I}] \ . \quad (4.4)$$

Figure 4.3 reveals that the value of r is only a scaling factor; the area of $\mathcal{I}$, $A(\mathcal{I})$, is proportional to r^2, the proportional factor being a function of m and ϵ only. This can also be checked by using the circle-circle intersection area formula, that we can for example find in [Wei99].
Let $A(\mathcal{I}) = C(m, \epsilon) r^2$.
As $g(n, r)$ is by hypothesis an a.a.s. connected graph, we know by Theorem 4.3.1 that there exists a function $k(n)$ such that $\pi r^2 n = \ln n + k(n)$ and $\lim_{n\to+\infty} k(n) = +\infty$.
Thus $A(\mathcal{I}) = C(m, \epsilon) r^2 = C(m, \epsilon) \frac{\ln n + k(n)}{\pi n}$
Lemma 4.3.3 $\Rightarrow$
$\lim_{n\to+\infty}\ P[no\ node\ in\ \mathcal{I}]$
$\leq \lim_{n\to+\infty} \exp(-nA(\mathcal{I}))$
$\leq \lim_{n\to+\infty} \exp(-\frac{C(m,\epsilon)}{\pi}(\ln n + k(n))) = 0$
Hence, whatever the value of m and ϵ,

$$lim_{n\to\infty} P[some\, n_i\, in\, \mathcal{I}] = 1 \ . \quad (4.5)$$

Moreover, as the Euclidean distance between n_1 and n_i is strictly less than mr, by inductive hypothesis we have:

$$lim_{n\to\infty} P[path\_len(n_1, n_i) \leq m \,|\, some\, n_i\, in\, \mathcal{I}] = 1 \ . \quad (4.6)$$

Equations 4.4, 4.5 and 4.6 $\Rightarrow$

$$lim_{n\to\infty} P[path\_len(n_1, n_2) \leq m + 1] = 1$$

The property is thus verified for $m + 1$.
By induction, the property is valid for any integer.

■

Intuitively, this theorem means that as a network gets denser and becomes a.a.s. connected, the shortest path between any pair of nodes draws close to a straight line. This interpretation is illustrated by Figure 4.4.

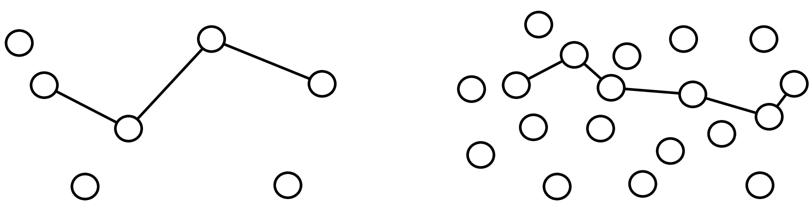


Figure 4.4: Interpretation of the asymptotic path length theorem

Using the asymptotic path length theorem, we can derive the main result of this section. It extrapolates the theorem of asymptotic connectivity for dense basic graph to overlay graphs, as shown by the table and drawing on Figure 4.5.

Basic graph	**Interpretation**	**Overlay graph**
n	number of vertices	Dn
r	maximal edge length	Rr

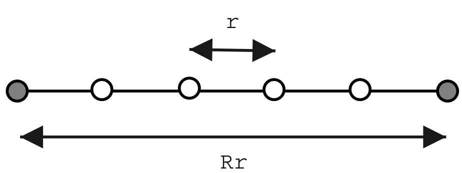


Figure 4.5: Extrapolation of theorems about asymptotic connectivity for overlay graphs

Theorem 4.3.5 *(Sufficient condition for the asymptotic connectivity of dense overlay graphs)*

Consider an overlay graph $G(g, D(n), R(n))$. *Assume* $g(n, r(n))$ *is a.a.s. connected and* $\lim_{n\to+\infty} Dn = +\infty$. *If*

$$\pi(Rr)^2 = \frac{\ln(\lceil Dn \rceil) + K(n)}{\lceil Dn \rceil} \tag{4.7}$$

with $\lim_{n\to+\infty} K(n) = +\infty$ *then* G *is a.a.s. connected.*

Proof: Let us build a graph $g\prime(\lceil Dn \rceil, Rr)$ such that the vertices set of $g\prime$ and G are the same.
Consider an edge of $g\prime$ linking two nodes n_1 and n_2.
By definition, the distance between n_1 and n_2, denoted $|n_1 n_2|$, is less than or equal to Rr.
Let us first assume that $|n_1 n_2| < Rr$. By Theorem 4.3.4, as g is a.a.s. connected, the maximum number of hops between n_1 and n_2 is a.a.s. less than or equal to R. Hence, asymptotically, any edge of $g\prime$ of length strictly less than Rr also exists in G.
Let us now assume that $|n_1 n_2| = Rr$. We can draw two disks of radius Rr respectively centered on n_1 and n_2. Let $\mathcal{I}$ denote the disks intersection and $A(\mathcal{I})$ its area. As shown in Fig. 4.6, $A(\mathcal{I})$ is minimal when n_1 and n_2 are both located on a border of the field.

Using the circle-circle intersection area formula [Wei99], we obtain $A(\mathcal{I}) \geq C(Rr)^2$ with $C = \frac{1}{12}(4\pi - 3\sqrt{3})$, wherever n_1 and n_2 are located. By lemma

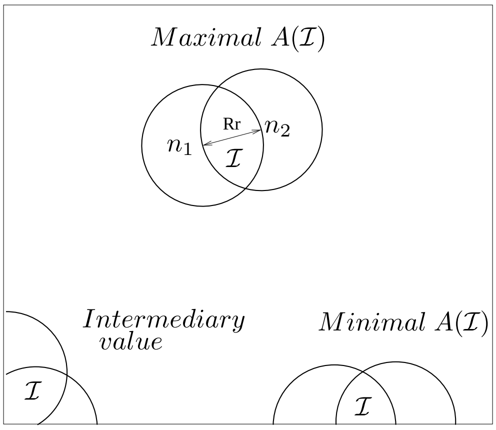


Figure 4.6: Drawing for the proof of the sufficient condition for asymptotic connectivity of dense overlay graphs

4.3.3, the probability that there is no overlay node in $\mathcal{I}$ is less than or equal to $exp(-C(\lceil Dn \rceil - 2)(rR)^2)$.

Assume that $\pi(Rr)^2 = \frac{\ln(\lceil Dn \rceil)+K(n)}{\lceil Dn \rceil}$ with $\lim_{n\to+\infty} K(n) = +\infty$.

Asymptotically, there exists almost surely an intermediary overlay node $n_i \in \mathcal{I}$.

The distances $|n_1n_i|$ and $|n_in_2|$ are strictly less than Rr thus, by Theorem 4.3.4 with $m = R$, there exists a.a.s. two edges (n_1, n_i) and (n_i, n_2) in G.

Thus, for any edge (n_1, n_2) of $g\prime$, there exists a path between the corresponding nodes in G.

As their vertices sets are the same, the asymptotic connectivity probability of G is greater than or equal to the asymptotic connectivity probability of $g\prime$.

We assumed that $\pi(Rr)^2 = \frac{\ln(\lceil Dn\rceil)+K(n)}{\lceil Dn\rceil}$ with $\lim_{n\to+\infty} K(n) = +\infty$. Consequently, by Theorem 4.3.1, $g\prime$ is a.a.s. connected.

Thus G is also a.a.s. connected. ■

4.3.4 Discussion

The following corollaries are meant to give an insight about the relationship between the neighbourhood range and the overlay density. For both of them, we consider an overlay graph $G(g, D, R)$ and make the assumptions that g is a.a.s. connected and that $\lim_{n\to+\infty} Dn = +\infty$.

As we will extensively use the notations for the asymptotic behaviour of functions in the following, we recall them in Appendix B.

The first observation we can make about Theorems 4.3.2 and 4.3.5 is that, as for basic graphs, the necessary condition for the asymptotic connectivity of overlay graphs is also sufficient.

Consider a basic graph $g(n, r)$ a.a.s. connected. An overlay graph $G(n, r, D, R)$ is a.a.s. connected if and only if $g\prime(\lceil Dn\rceil, Rr)$ is a.a.s. connected. Homogeneous topology control algorithms reduce the transmitting range of ad hoc nodes in order to improve the overall capacity of the network. Shorter transmission ranges can be used when the number of nodes per unit area increases. Likewise for overlay graphs, the more overlay nodes (Dn), the lower product of neighbourhood and transmission ranges (Rr) can be used. In particular, the first corollary presented below states that $DR^2 = 1$ is sufficient in all cases to obtain the overlay graph connectivity a.a.s.

Corollary 4.3.6 *If $DR^2 \geq 1$ then G is a.a.s. connected.*

Proof: If basic graph g is a.a.s. connected then there exists a function $k(n)$ such that $\pi r^2 n = \ln n + k(n)$ and that $\lim_{n\to+\infty} k(n) = +\infty$.

Thus $DR^2\pi r^2 n \geq \pi r^2 n = \ln n + k(n) \geq \ln(Dn) + k(n)$. By Theorem 4.3.5, the overlay graph is a.a.s. connected. ■

The sufficient condition $R \geq \frac{1}{\sqrt{D}}$ shows that a decreasing overlay density does not necessarily make the overlay graph a.a.s. disconnected. We can for example have $D = \frac{1}{\ln n}$ and $R = \sqrt{\ln n}$. It also confirms the intuitive idea that the lower D is, the larger R must be.

The advantage of the previous corollary is that we do not need any information about the basic graph, except that it is a.a.s. connected. It states that $DR^2 = 1$ is sufficient in all cases to obtain the overlay graph connectivity a.a.s.

However, lower values for the neighbourhood range could be obtained if the relationship existing between n and r is known.

Corollary 4.3.7 *Let $\pi r^2 n = \ln n + k(n)$ with $k(n) \gg 1$. Assume D is constant and R is an integer with $R \geq 1$.*

1. *If $k(n) \gg \ln n$ then G is a.a.s. connected for any R.*

2. *If $k(n) \geq a \ln n$ with $a > 0$ then G is a.a.s connected for any $R > \frac{1}{\sqrt{D(1+a)}}$.*

3. *If $k(n) \ll \ln n$, G is a.a.s. connected if and only if $R \geq \frac{1}{\sqrt{D}}$.*

Proof: Let $K(n) = \pi r^2 n D R^2 - ln(Dn)$. G is connected if and only if $K(n) \gg 1$.

1. By definition, if $k(n) \gg \ln n$ then for every $M > 0$ and n sufficiently large, $k(n) \geq M \ln n$. By hypothesis, $R \geq 1$ thus $K(n) \geq [D(1+M) - 1] \ln n - \ln D \geq [D(1+M) - 1] \ln n$. Let $M = \frac{1}{D}$. For n sufficiently large, $K(n) \geq D \ln n$. By definition, $D > 0$ thus $K(n) \gg 1$.

2. If there exists $a > 0$ such that $k(n) \geq a \ln n$, then $K(n) \geq [DR^2(1 + a) - 1] \ln n$. If $R > \frac{1}{\sqrt{D(1+a)}}$ then $K(n) \gg 1$.

3. For $k(n) \ll \ln n$, we know by Corollary 4.3.6 that $DR^2 \geq 1$ assures the asymptotic connectivity of G. Assume $DR^2 < 1$. By definition, if $k(n) \ll \ln n$ then for any $\epsilon > 0$ and n sufficiently large, $k(n) < \epsilon \ln n$. This gives $K(n) < [(1+\epsilon)DR^2 - 1] \ln n - \ln D$. Let $\epsilon = \frac{1-DR^2}{DR^2}$. By hypothesis, $DR^2 < 1$ thus $\epsilon > 0$. For $DR^2 < 1$, there exists $\epsilon > 0$ such that for n sufficiently large, we have $K(n) < -\ln D \Rightarrow \lim_{n \to +\infty} K(n) \neq +\infty$.

■

Concerning a basic graph, a function $k(n)$ that grows quickly just accelerates the convergence of the connectivity probability [SB03]. This function has a stronger impact on the neighbourhood range needed for connectivity. For example, for a constant overlay density D, it decides if R can take any value or must be greater than a fixed threshold.

In particular, if the transmission range r is kept constant while the number of nodes grows, we have $k(n) \gg \ln n$ which implies that $R = 1$ is sufficient to obtain an a.a.s. connected overlay. The overlay nodes do not need other intermediary nodes to forward their packet for communicating. The subnetwork composed of the overlay nodes only is a.a.s. connected. In fact, there is no need for building an overlay in this case. The overlay nodes can directly use their own routing protocol, with customised packet format.

Oppositely, if a topology control protocol is used for optimising the transmission range, $R = 1$ can be too small to make the overlay a.a.s. connected. In this case, the subnetwork composed of the overlay nodes only is a.a.s. disconnected. It is necessary for some overlay nodes to communicate through intermediary non overlay nodes. Overlay techniques are required; the overlay nodes control and data packets must be encapsulated in packets that can be routed by all nodes.

On Figure 4.7, from top to bottom, decreasing transmission range values are set for the same distribution of 6 nodes. The corresponding underlay topologies become sparser. On the upper sub-figure, it is a full-mesh, while on the lowest one, the number of edges is minimal. The two grey-shaded nodes are overlay members. The dashed edges form a path, or overlay link, between them. On the upper topology, only one packet is required for the forwarding of an overlay message from one overlay node to the other. On the intermediate topology, two packets are needed. On the sparsest topology, three packets are necessary. We thus observe that the critical neighbourhood range increases from the full-mesh to the minimal homogeneous underlay topology. We deduce from corollary 4.3.7 and from this simple example that the more efficient the underlay topology control algorithm is, the more traffic is needed for the construction, maintenance and usage of an overlay.

4.4 Sparse Networks

4.4.1 Known Results on Basic Graphs

Combining results from [SB03] and [MP03], we can state the following conditions on the connectivity of a basic graph $g(n, r, \ell)$. Refer to Sect. 4.2 for notations.

Theorem 4.4.1 *(Necessary condition for the asymptotic connectivity of sparse basic graphs)*
Let r be strictly less than $\sqrt{2}\ell$. If $r^2 n = O(\ell^2)$ then $g(n, r, \ell)$ is not a.a.s. connected. If $r = O(\ell^\epsilon f(\ell))$ with $0 \leq \epsilon < 1$ and $f(\ell)$ a function that grows strictly slower than any function of type ℓ^γ where $\gamma > 0$ and if $r^2 n < \frac{1}{2}(1-\epsilon)\ell^2 \ln \ell$ then $g(n, r, \ell)$ is a.a.s. not connected.

Theorem 4.4.2 *(Sufficient condition for the asymptotic connectivity of sparse basic graphs)*
If $r \geq \sqrt{2}\ell$, then $g(n, r, \ell)$ is a.a.s. connected. If $r = \Omega(\ell)$ and $r^2 n = \Omega(\ell^2 \ln \ell)$, then $g(n, r, \ell)$ is a.a.s. connected. If $r = \Omega(\ell^\epsilon f(\ell))$ with $0 \leq \epsilon < 1$ and $f(\ell)$ a function that grows strictly slower than any function of type ℓ^γ where $\gamma > 0$ and if $r^2 n \geq 4(1-\epsilon)\ell^2 \ln \ell$ then $g(n, r, \ell)$ is a.a.s. connected.

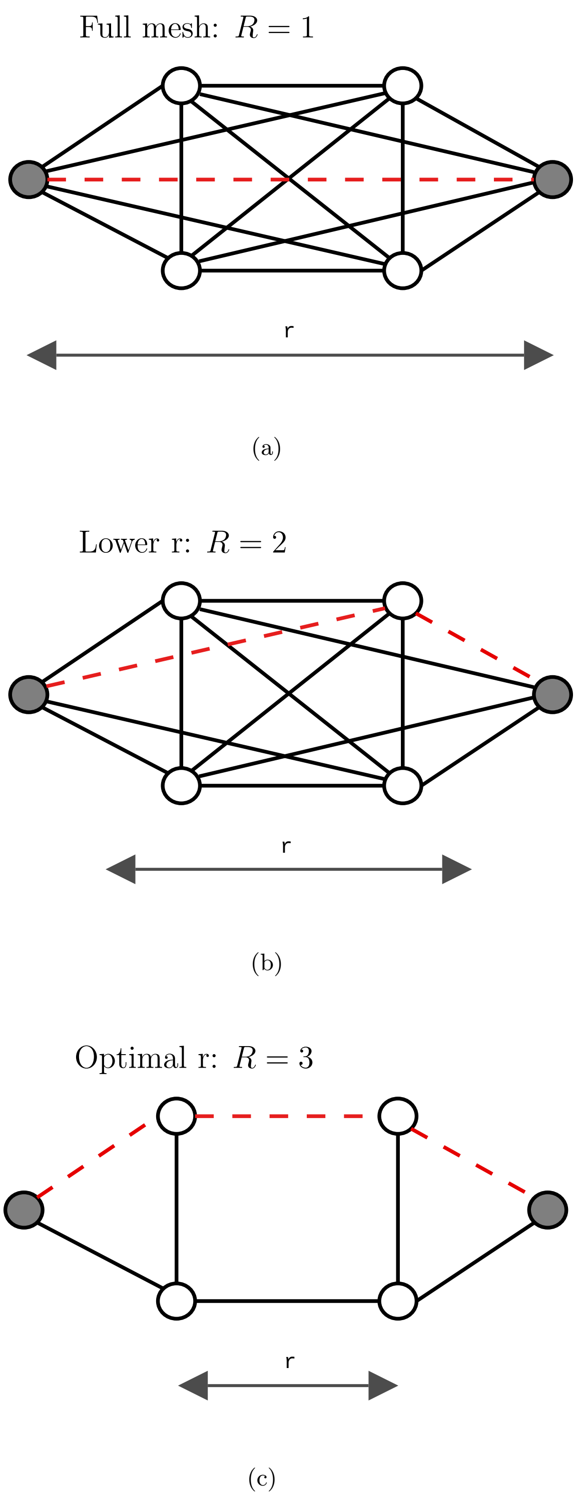


Figure 4.7: Impact of underlay topology control protocols on the overlays control and data traffic

4.4.2 Minimal Neighbourhood Range

Theorem 4.4.3 *(Necessary condition for the asymptotic connectivity of sparse overlay graphs)*
Let Rr be strictly less than $\sqrt{2}\ell$. If $(Rr)^2\lceil Dn\rceil = O(\ell^2)$ then $G(g, D, R)$ is not a.a.s. connected. If $Rr = O(\ell^\epsilon f(\ell))$ with $0 \le \epsilon < 1$ and $f(\ell)$ a function that grows strictly slower than any function of type ℓ^γ where $\gamma > 0$ and if $(Rr)^2\lceil Dn\rceil < \frac{1}{2}(1-\epsilon)\ell^2 \ln \ell$ then $G(g, D, R)$ is a.a.s. not connected.

Proof: As for dense networks, if a graph $g\prime(\lceil Dn\rceil, Rr, \ell)$ is not connected a.a.s., $G(n, r, \ell, D, R)$ cannot be connected a.a.s. Applying Theorem 4.4.1 to a graph $g\prime(\lceil Dn\rceil, Rr, \ell)$, we obtain the above conditions on the asymptotic connectivity of G. ■

4.4.3 Sufficient Neighbourhood Range

We use the same techniques as for dense graphs, the only difference being that the probability for a node to be located on a surface S equals $\frac{S}{\ell^2}$, instead of S.

We first demonstrate that Theorem 4.3.4, called the asymptotic path length theorem and stated in Sect. 4.3, still holds for sparse graphs.

Proof: (the asymptotic path length theorem in the context of sparse graphs)
If $r \ge \sqrt{2}\ell$, every overlay node can reach any other overlay node in one hop. Assume $r < \sqrt{2}\ell$ and let P_S denote the probability that there is no node on a surface $S = c\pi r^2$. For any constant $c > 0$, by lemma 4.3.3, $\lim_{\ell\to+\infty} P_S \le \lim_{\ell\to+\infty} \exp(-n\frac{c\pi r^2}{\ell^2})$. Graph g is a.a.s. connected, thus, by Theorem 4.4.1, $r^2 n \gg \ell^2$. This implies that $\lim_{\ell\to+\infty} P_S = 0$ and, using the same technique as for dense graphs (see proof of Theorem 4.3.4 in Sect. 4.3), we can demonstrate that if $|n_1 n_2| < mr$ then there exists a path between n_1 and n_2 composed of at most m hops. ■
Using this theorem, we can derive the main result of this section.

Theorem 4.4.4 *(Sufficient condition for the asymptotic connectivity of sparse overlay graphs)*
Let $g(n, r, \ell)$ be a.a.s. connected. If $r \ge \sqrt{2}\ell$, then $G(g, D, R)$ is a.a.s. connected. If $Rr = \Omega(\ell)$ and $(Rr)^2\lceil Dn\rceil = \Omega(\ell^2 \ln \ell)$, then $G(g, D, R)$ is a.a.s. connected. If $Rr = \Omega(\ell^\epsilon f(\ell))$ with $0 \le \epsilon < 1$ and $f(\ell)$ a function that grows strictly slower than any function of type ℓ^γ where $\gamma > 0$ and if $(Rr)^2\lceil Dn\rceil \ge 4(1-\epsilon)\ell^2 \ln \ell$ then $G(g, D, R)$ is a.a.s. connected.

Proof: If $r \ge \sqrt{2}\ell$, every overlay node can reach any other overlay node in one hop and G is connected whatever parameters D and R are. The hypotheses imply, by Theorem 4.4.2, that a graph $g\prime(\lceil Dn\rceil, rR, \ell)$ is a.a.s. connected.

As for dense graphs, exploiting Theorem 4.3.4, we can prove that if a graph $g\prime(\lceil Dn\rceil, rR, \ell)$ is a.a.s. connected then G is a.a.s. connected.

Consequently, the hypotheses imply that G is a.a.s. connected. ■

4.4.4 Discussion

For sparse networks, no condition for the asymptotic connectivity has been demonstrated to be both necessary and sufficient. Note however that the bounds for basic graphs given by Theorems 4.4.1 and 4.4.2 are asymptotically tight and that they have remained close for overlay graphs.

As for dense graphs, values for the neighbourhood range can be obtained if the relationship existing between ℓ and r is known. For example, if $r = \sqrt{a\ell \ln \ell}$ with $a > 0$ and $n = 2\ell$, a sufficient condition for the overlay graph to be a.a.s. connected is: $R \geq 1$ and $(Rr)^2 Dn \geq 4(1 - \frac{1}{2})\ell^2 \ln \ell$, which is fulfilled if $R \geq \lceil \frac{1}{\sqrt{aD}} \rceil$.

Corollary 4.4.5 *Let $g(n, r, \ell)$ be an a.a.s. connected graph. Assume D is constant. If $r^2 n \gg \ell^2 \ln \ell$ then G is a.a.s. connected for any $R \geq 1$.*

Proof: If D is constant, $R \geq 1$ and $r^2 n \gg \ell^2 \ln \ell$ then $(Rr)^2 Dn \gg \ell^2 \ln \ell$ and, by Theorem 4.4.4, G is a.a.s. connected. ■

In particular, if the node density is kept constant while the field length grows, and if the transmission range is such that $r \gg \sqrt{\ln \ell}$, $R = 1$ is sufficient to obtain an a.a.s. connected overlay. As explained in Sect. 4.3.4, there is no need for building an overlay in this case.

Oppositely, if a topology control protocol is used for optimising the transmission range, the basic graph can be a.a.s. connected with a transmission range only proportional to $\sqrt{\ln \ell}$, while the subnet composed of the overlay nodes only can be a.a.s. disconnected. As also explained in Sect. 4.3.4, overlay techniques are then required.

4.5 Conclusions

We presented and analysed the critical neighbourhood range problem.

We demonstrated that in connected networks, as the network gets denser ($n \to +\infty$) or larger ($\ell \to +\infty$), the shortest path between any pair of nodes draws close to the straight line. This sets an upper bound on the number of hops between any pair of nodes, knowing the distance between them and the nodes transmission range r.

This property, that we called the asymptotic path length theorem, and known works on the critical transmission range problem, allowed us to derive an analytical solution to the critical neighbourhood range problem for dense networks. For a large class of sparse networks, we determined asymptotically tight bounds.

The mathematical conditions obtained do take into account the potential use of a homogeneous topology control algorithm and allow the overlay density D to evolve with the network size (n or ℓ). In particular, if D diminishes, they show how a compensation in R can keep the overlay still connected.

The analysis of these results provides, among others, the following properties for overlays built on ad hoc networks.

Whatever the characteristics of the underlying network are, an overlay built on a dense connected network with $DR^2 \geq 1$ is asymptotically almost surely connected. We conjecture that this still holds in sparse networks.

In many cases, if the relationship between n and r is known, one can set R to a lower value than $\lceil \frac{1}{\sqrt{D}} \rceil$ and still obtain asymptotic overlay connectivity.

For constant D, depending on the network degree of connectivity, the minimal value of R for asymptotic overlay connectivity can either be equal to one, or to a higher fixed threshold, or be an unbounded function of the network size.

In particular, in dense networks, if D and r are kept constant while the number of nodes increases, the overlay nodes can asymptotically use their own routing protocol, bypassing the network routing protocol common to all nodes. This is also the case in sparse networks if the node density is kept constant and r increases with the field length ℓ so that $r \gg \sqrt{\ln \ell}$. However, this strategy has a negative impact on the network capacity, and a topology control algorithm is more likely to be applied. In this case, the network composed only of the overlay nodes can be asymptotically disconnected. It is necessary to resort to the overlay technique, and to use a higher overlay neighbourhood range, in order to build an asymptotically connected overlay.

Chapter 5

The Critical Neighbourhood Range Heuristic Study

5.1 Study motivation and overview

In Chapter 3, we showed the feasibility of the Reactive Overlay approach. An overlay routing protocol can work without pre-establishing the overlay topology. However, good performance are obtained only if an appropriate value is assigned to the neighbourhood range.

In Chapter 4, we derived analytical solutions to the critical neighbourhood range problem in the asymptotic case, i.e. in networks with an infinite number of nodes.

In this chapter, we first show that the asymptotic critical neighbourhood range value is not sufficient for building connected overlays with a high probability in finite networks. Hence, we explore heuristics for estimating the critical neighbourhood range (CNR). On this basis, we present and evaluate a simple protocol, ReactiveOtc, that provides an appropriate neighbourhood range to overlay routing protocols.

5.2 Connectivity Study

In this Section, we compare several heuristics for estimating the critical neighbourhood range:

1. Use the asymptotic value, determined in previous chapter.
2. Use a fixed number of hops, determined empirically so as to ensure connectivity with a high probability.
3. Increase the neighbourhood range until K overlay nodes have been reached, with K a target number of neighbours determined empirically so as to ensure connectivity with a high probability.

We first discuss the best heuristic for the static case. We will study its convenience to mobile situations in Sec. 5.3.

5.2.1 Model

We model the ad hoc network by a random geometric graph.

The ad hoc nodes are represented by the graph's vertices. We randomly and uniformly distribute these on a unitary square field. We vary their number from 50 to 1000 and the overlay density from 10 to 90%. Overlay nodes are randomly and uniformly distributed on the set of ad hoc nodes.

The edges assignment is a bit more challenging. As presented in Section 2.2.1, there exist two classes of topology control (TC) algorithms, namely homogeneous and non-homogeneous. We showed in previous chapter that for a given set of ad hoc nodes and communication links, the more efficient underlay topology control algorithm we use, the more traffic is needed for the construction, use and maintenance of overlays built on top of its resulting logical topology. Hence, in order to test our discovery process in a stringent environment and to represent both types of underlay topology, we employ the logical topologies obtained after the use of the two following TC models:

1. An ideal homogeneous TC technique which assigns the same value r to each node's radio transmission range, r being the minimal value that makes the underlay connected, and

2. An efficient non-homogeneous TC technique which links every node to its m nearest neighbours, m being the minimal number of symmetric neighbours needed for connectivity.

For each number of nodes, we build 50 random geometric graphs. Then, for every overlay density, we count the number of connected overlays obtained with each heuristic.

5.2.2 Algorithms

Graphs representation We wrote a Java library for random geometric graphs. The library is derived from JGraphT [JGr], a free Java graph library that provides mathematical graph-theory objects and algorithms. JGraphT has a clean and simple API, and is designed to support high-performance and large-scale applications. However, it does not provide specific handling for geometric, nor random graphs. We copied the simple undirected graphs API, and adapted its functions for random geometric graphs.

The random geometric graph class has two direct important subclasses for basic and overlay graphs, as defined in Section 4.2. The basic graphs are further divided following the homogeneous and non-homogeneous topology control models. A diagram of these main classes is presented on Fig. 5.1.

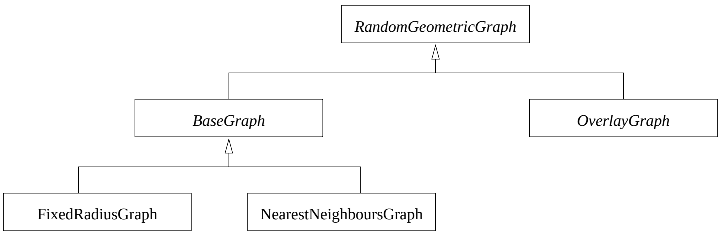


Figure 5.1: Main classes of the random geometric graph library

For each random geometric graph, an important instance variable is its list of nodes. Each node instance, in turn, has its own list of neighbours. If the node is a member of the overlay, a list of overlay neighbours is also instantiated. This completely defines the random geometric graph topology. This representation of graphs is known as the adjacency list data structure. It is preferred when the number of edges is low in comparison to the maximal number of edges that could be built with the same nodes set. It fits to ad hoc networks, as a high number of edges impairs the global capacity. Figure 5.2 shows our implementation of the adjacency list, for basic and overlay graphs.

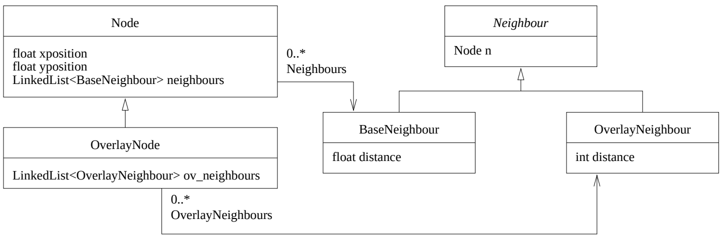


Figure 5.2: Edges are represented by an adjacency list

Computation of the basic graphs edges Each node is assigned a position on the unitary square. Homogeneous basic graphs $g(n, r)$ are calculated with a scanning method, from the computational geometry field. This avoids to test the distance between each pair of node, which has a complexity $O(n^2)$. The nodes are handled by increasing order of abscissa x_i. The distance between a pair of nodes (n_j,n_k) is computed if and only if $0 \leq x_k - x_j \leq r$. The algorithm is illustrated by Fig. 5.3. During its execution, the gray-shaded zone Z moves from left to right. The shift stops any time the right border of Z crosses a node n, for computing the neighbourhood of n positioned on its left. All these physical neighbours belong to Z, which greatly reduces the number of distances to be calculated and compared with the radio transmission range. Program 1 gives a sketch of the algorithm.

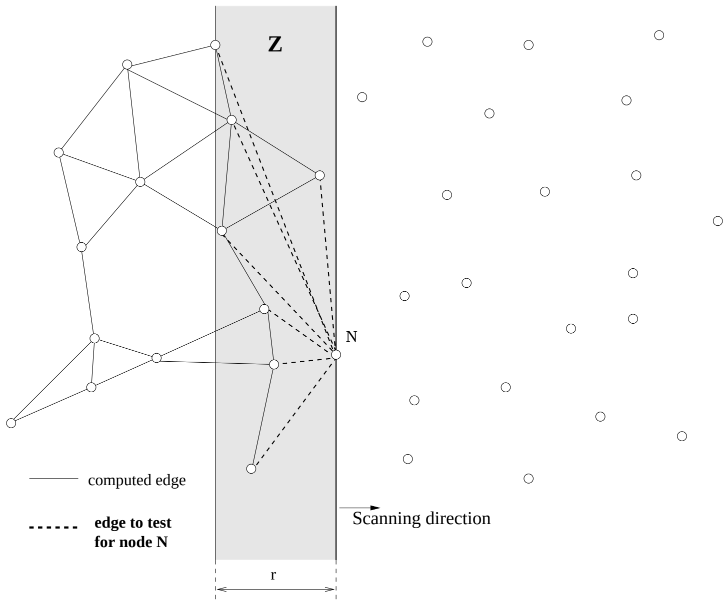


Figure 5.3: The edges of a basic graph $g(n, r)$ are computed by a scanning method

The code for the non-homogeneous basic graphs has also been designed with care for processing time, but is not noticeable.

Computation of the overlay graphs edges For calculating the overlay edges, we use modified versions of the Dijkstra algorithm [Dij59]. This algorithm builds the shortest path from a given node, the root, to all other nodes in increasing order of distance. All nodes initially receive a temporary label indicating that they are located at an infinite distance from the root. At each step, the node with the temporary label indicating the smallest distance to the root node is set definitive. If this node offers to some of its neighbours a path with a smaller length than the distance indicated by their label, their distance to the root is relaxed. Its complexity, when using a binary heap data structure for storing the visited nodes during the execution, has a complexity of $(m + n)\log(n)$, with m the number of edges.

When we need to locate all overlay nodes covered by a given neighbourhood range R, we run the algorithm until we get a definitive path strictly longer than R. Any overlay node that was previously marked with a definitive label is an overlay neighbour.

When we need to locate the K nearest overlay neighbours, we run the algorithm until the distance to K overlay nodes is computed.

With the grabbing messages architecture, described in Sec. 3.2.2, the discovery of overlay neighbours hidden by a closer overlay neighbour is avoided. This behaviour can also be obtained with the Dijkstra algorithm by prevent-

Algorithm 1 Scanning algorithm for computing the edges of a basic graph $g(n, r)$

```
Node[] nodes = getNodesSortedByAbscissa();

int nn = nodesNumber();
int firstElmtIndex, lastElmtIndex, j;
Node n1, n2;
float x1, x2;

firstElmtIndex = 0;

for (lastElmtIndex=0;lastElmtIndex<nn;lastElmtIndex++) {

  n2 = nodes[lastElmtIndex];
  x2 = n2.getX();

  for (j=firstElmtIndex;j<lastElmtIndex;j++) {

    n1 = nodes[j];
    x1 = n1.getX();

    if (x2 - x1 > transmission_range) {
      firstElmtIndex++;
    } else {
      if (getDistance(n1,n2) <= transmission_range)
        addEdge(n1,n2);
    }
  }
}
```

ing it to relax the distance of an overlay node's physical neighbours when its distance is set as definitive.

For calculating the optimal neighbourhood range, we progressively run a Dijkstra algorithm for each overlay node. At first step, each Dijkstra process computes the set of overlay nodes located at one hop from its root. The corresponding overlay edges are added to the overlay graph and the soft state of each Dijkstra process is stored. If the overlay obtained is not connected, we let the Dijkstra algorithm progress for every overlay node until all pairs of overlay neighbours separated by two hops have been computed. If the overlay obtained is not yet connected, we add the pairs of overlay nodes located at three hops from each other, and so on.

The procedure is similar for calculating the optimal neighbourhood car-

D	0.1	0.3	0.5	0.7	0.9
$\frac{1}{\sqrt{D}}$	3.16	1.83	1.41	1.20	1.05
NR	4	2	2	2	2

Table 5.1: Asymptotic CNR for various overlay densities

dinality.

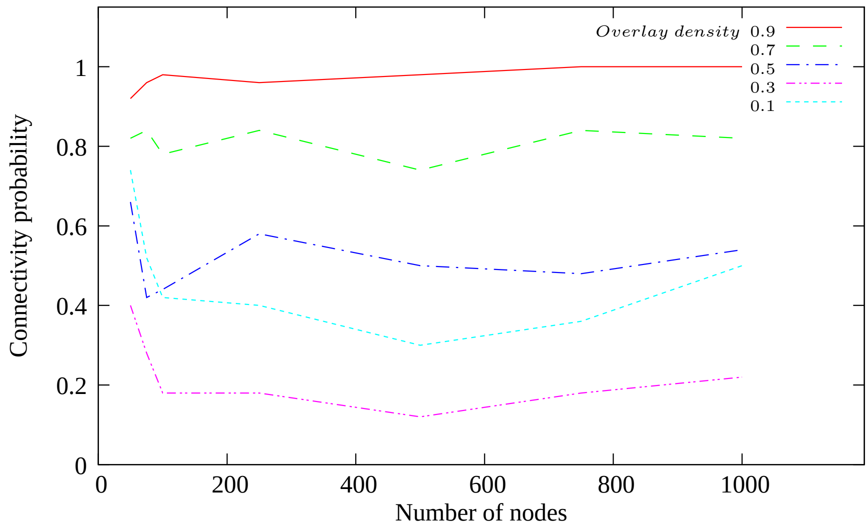
Overlay connectivity test We build an undirected weighted graph G with the overlay nodes as vertices. The weight of an edge is set to the distance calculated between its two end vertices on the underlay during the overlay edges computation. We then run the Dijkstra algorithm from an arbitrary vertex of G. The overlay graph is connected if and only if all its vertices have been visited at the end of the execution.

5.2.3 Asymptotic neighbourhood range

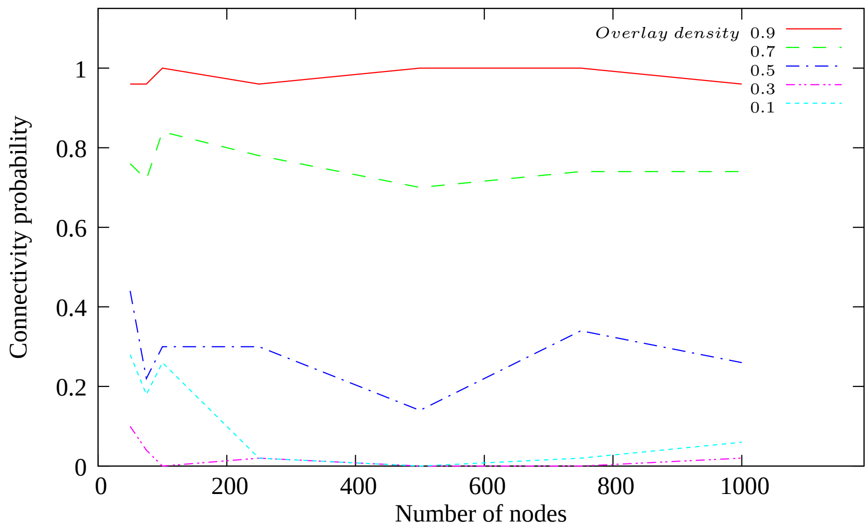
The first heuristic sets the neighbourhood range (NR) to $\lceil \frac{1}{\sqrt{D}} \rceil$. We demonstrated in Sect. 4.3.4 that this value is sufficient for building an overlay graph connected a.a.s. when the basic graph is itself connected with a high probability. These tests evaluate how the asymptotic value fits to the finite case. The overlay densities examined and their respective neighbourhood ranges are given by Table 5.1. We first observe that this heuristic is coarse. It provides a neighbourhood range of 2 for the large interval $0.25 \leq D < 0.81$. Its non-linearity makes it more adaptive for lower overlay densities. Nevertheless, we see on Fig. 5.4 that the percentage of connected overlay graphs is much too low, except for the highest overlay density.

5.2.4 Empirical neighbourhood range

For each underlay topology and overlay density, we calculate the critical neighbourhood range R_{opt}. Its average value for various overlay densities and number of nodes is compared on Fig. 5.5 to the asymptotic critical neighbourhood range value, defined as $R_\infty = \lceil \frac{1}{\sqrt{D}} \rceil$. The 95%-confidence intervals are drawn. The curves of the theoretical and experimental CNR have the same shape, but R_∞ is under R_{opt} for all overlay densities below 70%. On homogeneous underlays, the average CNR is a real value comprised between 2 and 6, while on non-homogeneous underlays, it belongs to the interval $(2, 8)$. The curves for different network sizes are close to each other, while a larger CNR is needed at low overlay densities. Its variance is also larger for lower overlay densities. The CNR thus varies more with the overlay density than with the number of nodes. Without any information about the overlay density nor the underlay type, these results do not provide any more



(a) Homogeneous underlay topology



(b) Non-homogeneous underlay topology

Figure 5.4: Percentage of connected overlays obtained with $NR = \lceil \frac{1}{\sqrt{D}} \rceil$

accurate estimation than setting the neighbourhood range to 8. Yet, the performance study we presented in Chapter 3 indicates that this strategy is not sustainable.

5.2.5 Empirical neighbourhood cardinality

An alternative heuristic consists in increasing the neighbourhood range of each overlay node until it gets linked with a target number of overlay neighbours, that we call the *critical overlay neighbourhood cardinality* or, shorter, the critical neighbourhood cardinality and denote CNC. As presented in Section 2.2.1, this type of strategy has been used in *k-Neigh* TC protocol for setting locally the transmission range of every node such as building globally a connected network with a high probability.

Reduction and extension rules

Let L_U^K denote the set of K nearest overlay neighbours of U. Overlay nodes U and V are *K-symmetric neighbours* if and only if $U \in L_V^K$ and $V \in L_U^K$. Figure 5.6 shows an example with $K = 1$.

Many MANET routing protocols assume bidirectional links. Moreover, using unidirectional links in route searches only provides an incremental benefit because of the high overhead needed to handle them [MD02]. We thus fix as an objective to build overlay topologies where the neighbourhood relation is symmetric.

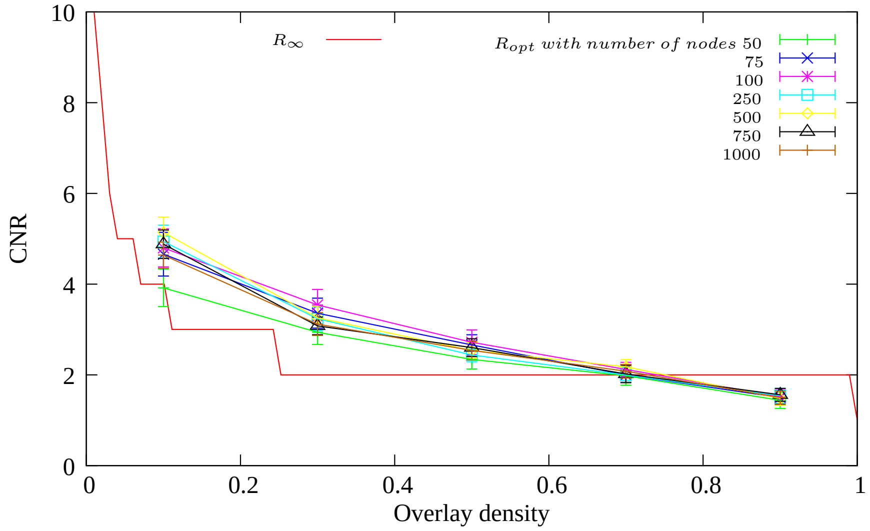
Let L_U denote the set of overlay neighbours selected by overlay node U. For each pair of overlay nodes U and V, there could be two rules to ensure symmetry of the overlay topology:

1. Reduction rule: $V \in L_U$ iff $U \in L_K^V$ *AND* $V \in L_K^U$,
2. Extension rule: $V \in L_U$ iff $U \in L_K^V$ *OR* $V \in L_K^U$ (graph symmetric closure)

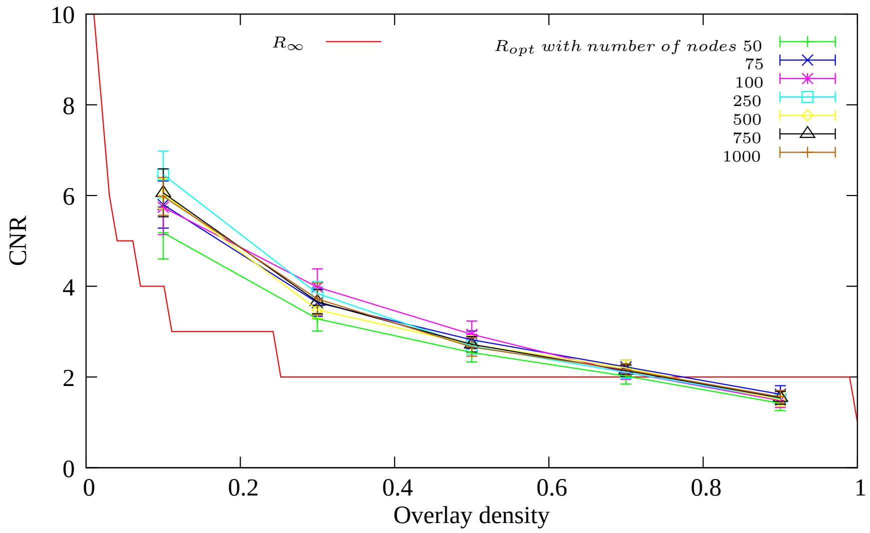
With the reduction rule, only the symmetric K-neighbours of a node are included in its neighbourhood. With the extension rule, asymmetric K-nearest neighbours are also considered. For a given value K, the topology obtained with the extension rule is a super-graph of the topology obtained with the reduction rule. Its connectivity probability is thus higher. An example is given on Fig. 5.7.

Results

For each underlay topology and overlay density, we calculate the CNC with the reduction and extension rules. We denote them respectively K^{red} and K^{ext}. Their average value for various numbers of nodes and overlay densities are given respectively on Figs. 5.8 and 5.9.



(a) Homogeneous underlay topology



(b) Non-homogeneous underlay topology

Figure 5.5: Critical neighbourhood range (CNR)

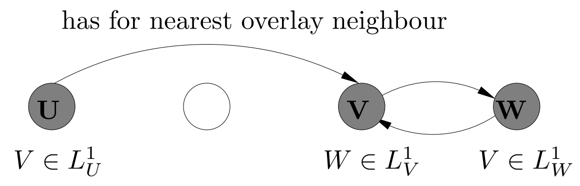
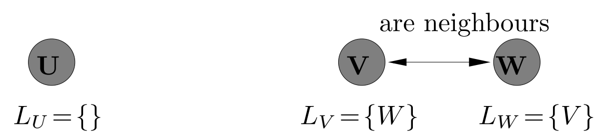
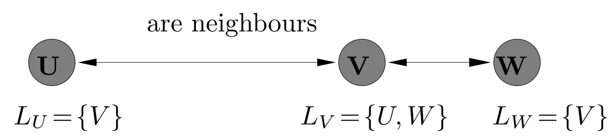


Figure 5.6: With $K = 1$, V is a K-asymmetric neighbour of U. V and W are K-symmetric neighbours.

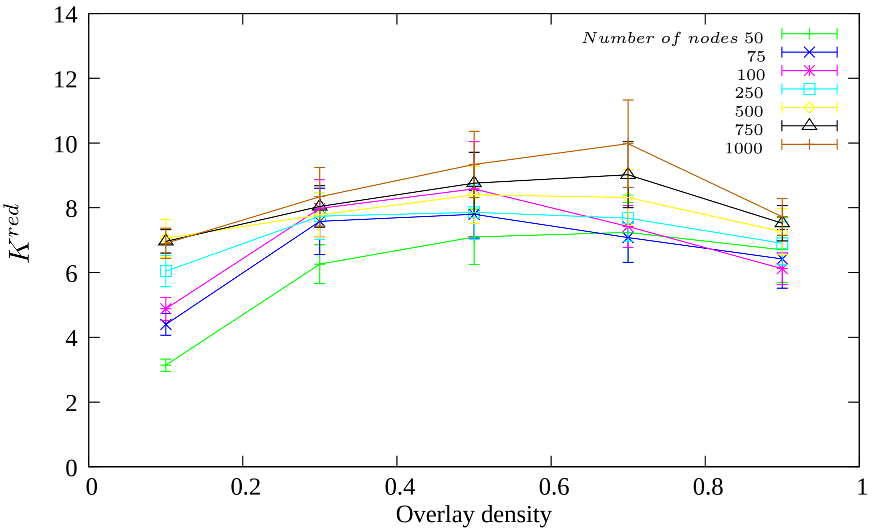


(a) The overlay topology obtained with the reduction rule for $NR = 1$ is not connected

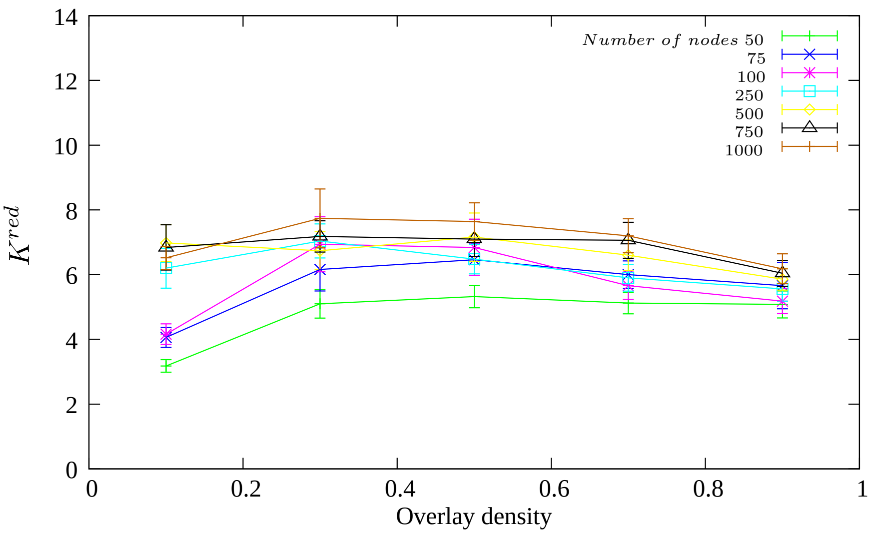


(b) The overlay topology obtained with the extension rule for $NR = 1$ is connected

Figure 5.7: Result of the reduction and extension rules on the same example topology, with asymmetric neighbours

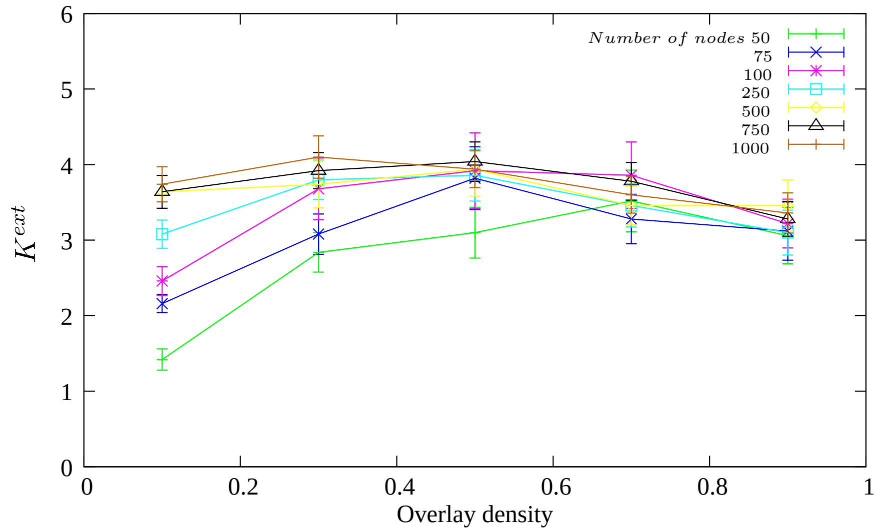


(a) Homogeneous underlay topology

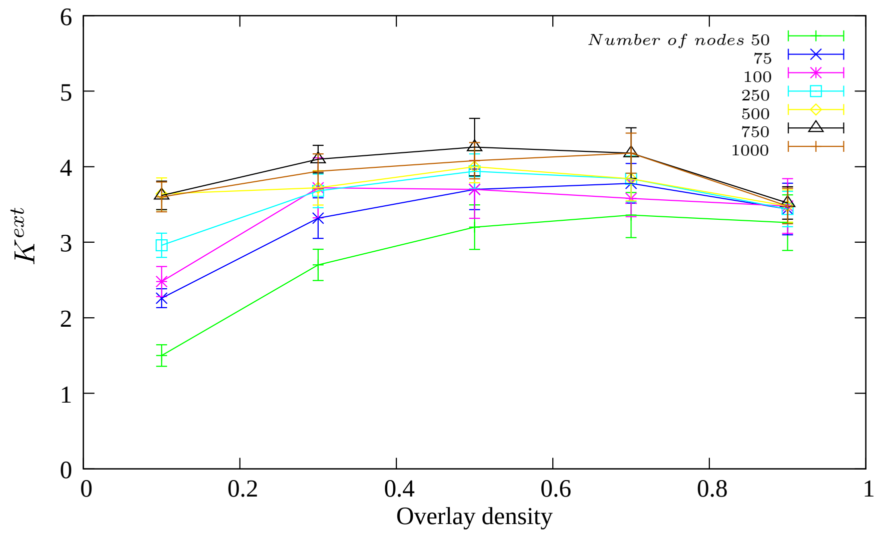


(b) Non-homogeneous underlay topology

Figure 5.8: Critical neighbourhood cardinality with the reduction rule



(a) Homogeneous underlay topology



(b) Non-homogeneous underlay topology

Figure 5.9: Critical neighbourhood cardinality with the extension rule

The average K^{red} is higher for the homogeneous underlay type than for non-homogeneous one. Its maximum value equals 10 in the former case, and 8 in the latter.

Oppositely, for all network size, overlay density and underlay type, the average K^{ext} is less than 5. Its variance is of the same order on both underlay types and for all overlay densities.

A further advantage of the extension rule over the reduction rule is shown on Fig. 5.10.

It shows the evolution of the percentage of overlays that are connected, for 200 tests with 500 nodes, as a function of the number of nearest overlay nodes (K) for both rules. The dashed horizontal lines are drawn at a 0.95 probability of connectivity.

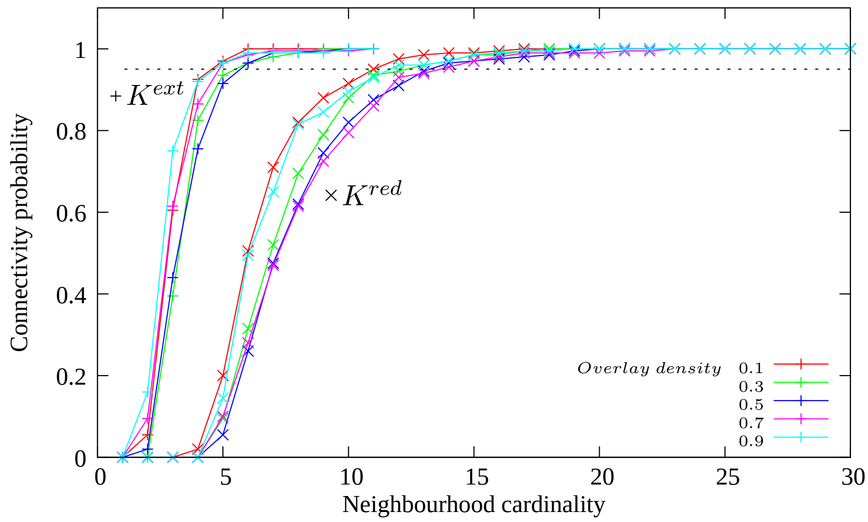
The five lowest curves are obtained with the reduction rule and the five highest with the extension rule. With both rules, there is a phase where the connectivity probability is very low and a phase where it is very high.

Let us denote K_{95} the number of overlay neighbours needed to obtain a connected overlay with a probability higher or equal to 95%. This value is much lower with the extension rule, that is if we do include the K-asymmetric neighbours, than with the reduction rule. With the extension rule, setting the neighbourhood cardinality to 6 is sufficient at all overlay densities and both underlay types, for providing 95% of connected overlays, without over-estimating the $K95$ of more than two overlay nodes. Note that at high overlay densities, these two unnecessary overlay neighbours could not require to increase the neighbourhood range. Moreover, as the transition from the low-probability phase to the high-probability one is sharper with the extension rule, the $K95$ value is more reliable in this latter case.

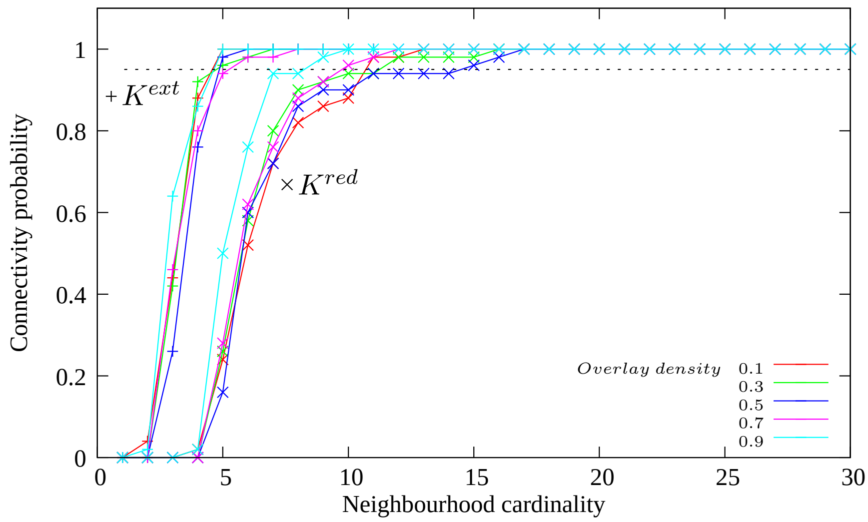
The same experiment has been conducted for nodes ranging from 50 to 1000. In all cases, for a given overlay density, the curves obtained for the different underlay sizes were very close from each other. In other words, we observed that the percentage of connected overlays is more influenced by the overlay density than by the number of nodes.

Table 5.11 gives the minimum number of nearest overlay neighbours that must be considered for obtaining 190 connected overlay topologies over 200, for 1000 nodes and different overlay densities. We respectively denote K_{95}^{red} and K_{95}^{ext} this value for overlays built with the reduction and with the extension rule. All results show that the value of K_{95}^{ext} is far less than K_{95}^{red}. The maximal value of K_{95}^{ext} on our whole set of experiments equals 8, while the maximal value of K_{95}^{red} gets to 30. This table also indicates the neighbourhood range (resp. R_{95}^{ext} and R_{95}^{red}) that must be admitted in order to reach the corresponding K_{95}^{red} and K_{95}^{ext} number of overlay neighbours. For each overlay density, the needed neighbourhood range is one to three hops longer, which is not negligible. The diffusion of an overlay message will thus consume less bandwidth if we accept the K-asymmetric neighbours.

We conducted the same experiment with increasing neighbourhood ranges.



(a) Homogeneous underlay topology (500 nodes)



(b) Non-homogeneous underlay topology (500 nodes)

Figure 5.10: Connectivity obtained with a fixed neighbourhood cardinality

Ov. density	K_{95}^{ext}	R_{95}^{ext}	K_{95}^{red}	R_{95}^{red}
0.1	5	5	11	8
0.3	7	3	17	5
0.5	8	3	19	4
0.7	8	2	25	4
0.9	6	2	15	3

Figure 5.11: Neighbourhood cardinality needed for a connectivity probability equal to 0.95 for the extension and reduction rules (1000 nodes)

Results, given on Fig. 5.12, confirm that K_{95}^{ext} is much less dependent of the overlay density than the neighbourhood range value required for obtaining a connected overlay with a probability higher or equal to 95% (R^{95}). The phase transition is also sharper for the overlay neighbourhood cardinality with extension rule than for the overlay neighbourhood range. Hence, the value K_{95}^{ext} is more reliable than R_{95}.

As a conclusion, the best heuristic studied is the empirical neighbourhood cardinality with the extension rule. The empirical value $K_{95}^{ext} = 8$ covers all network sizes from 50 to 1000 nodes, overlay densities from 10 to 90% and both underlay models studied.

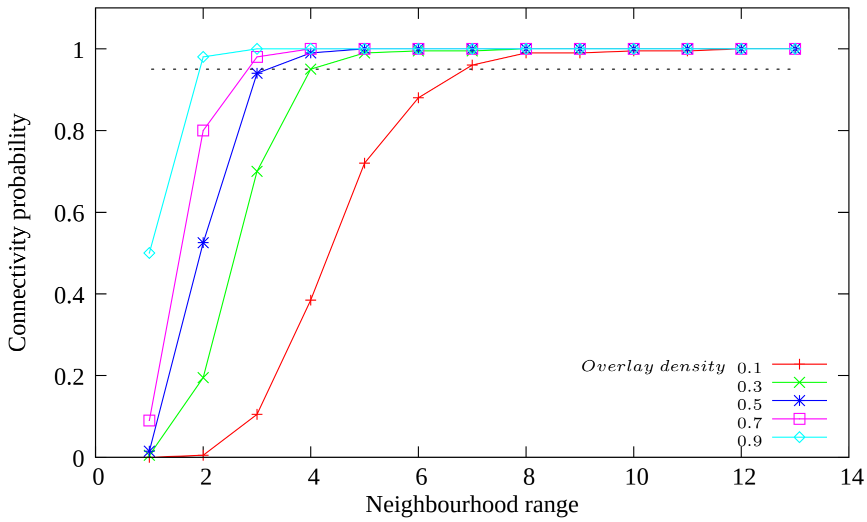
We would like to point out that the principles of this study is not restricted to the simple underlay model used in these simulations, which are only presented as illustrations. The important information they bring is not the particular value of $K_{95}^{ext} = 8$ but how it can be determined and why it is preferable to use the symmetric closure.

5.3 ReactiveOtc: A simple protocol for estimating the needed neighbourhood range

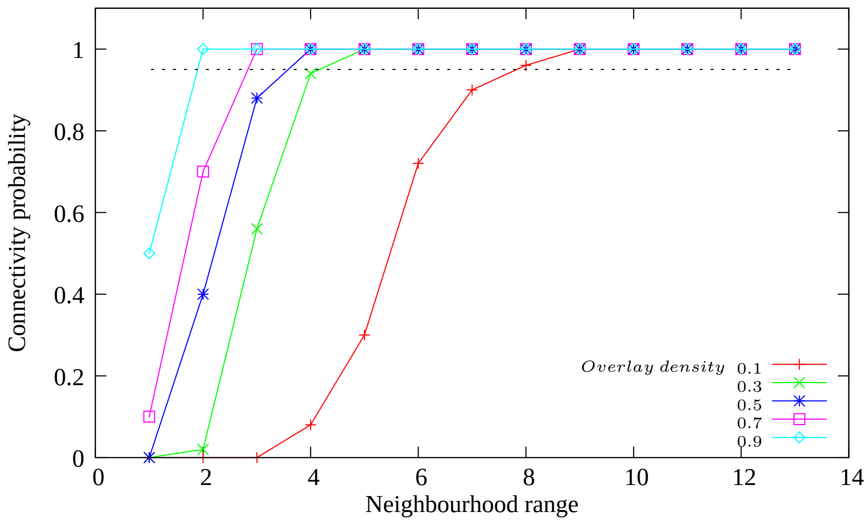
We thus propose an algorithm that uses a target number of overlay neighbours equal to 8 for estimating the critical neighbourhood range. In the following, we only present results obtained with the homogeneous TC model because all observations were similar in both cases. This corroborates our belief that the concepts exposed are valid for various topological models, and in particular for real networks.

5.3.1 Algorithm: Broadcast hellos on an increasing ring

The following algorithm, that we call *ReactiveOtc*, is an increasing ring announcement method with a different stop criteria for high and low overlay density. Its convergence relies on a uniform distribution of the overlay nodes in the ad hoc network. The overlay density may evolve with time but we



(a) Homogeneous underlay topology (500 nodes)



(b) Non-homogeneous underlay topology (500 nodes)

Figure 5.12: Connectivity obtained with a fixed neighbourhood range

assume that there is no region obviously more densely populated by overlay nodes than others.

Each overlay node regularly emits an overlay hello message. It also grabs the hello messages sent by other nodes, and systematically stores the sender identifier and its hop distance in its overlay neighbour table. An expiration time is associated with every overlay neighbour entry. Out of date information are regularly purged.

When a node enters the overlay, it sets its local variable *range* to 1 and enters the following cycle:

1. send an overlay hello message in a broadcast packet with $TTL = range$

2. listen to overlay hello messages and update the overlay neighbour table during a period Δ

3. recalculate *range* using its current value and the overlay neighbour table content

4. go to step 1

Algorithm 2 shows the computation done at step 3.

Assume that the number of overlay neighbours is equal or greater than *TARGET_NEI_NB*. The function *getSufficientRange(K)* returns the distance at which the *Kst* nearest overlay neighbour is located. The idea behind line *sufficient_range = getSufficientRange(TARGET_NEI_NB)* is that if an overlay node receives a sufficient number of neighbour advertisements from overlay nodes located at *range* hops, it should advertise himself to these overlay nodes. Hence, it sends hellos at this distance.

If the number of overlay neighbours is strictly less than *TARGET_NEI_NB*, this may indicate that the algorithm has not yet converged or that the overlay density is decreasing. In these cases, the *range* value is incremented. However, it may also occur, at high overlay density, that the algorithm converged but that the nearest overlay neighbours grab all the hello messages, preventing the reception of hello messages from farther overlay nodes. This possibility is pointed out by comparing the relative values of *range* and *d=farthestNeighbourDistance(TARGET_NEI_NB)*. The function *farthestNeighbourDistance()* returns the maximal distance over all neighbour entries. When *range* equals $d+2$, this means that no overlay neighbour has been heard farther than distance d during the listening time corresponding to at least two increasing ring searches. The overlay density is thus assumed to be high and the *range* frozen to $d+2$. Note that our performance study indicates that the value of two rings is suitable for all overlay densities above or equal to 10%. For lower values, it could happen that no overlay neighbour is hidden but that there are really two successive empty rings because of the few number of overlay nodes.

Algorithm 2 CNR estimation by ReactiveOtc

```
void
ReactiveOtc::timeout() {
  int nei_nb = getNeighboursNumber();
  if (nei_nb == 0) { // No neighbour
    if (range < MAX_RANGE)
      range++;
  } else if (nei_nb < TARGET_NEI_NB) { // Not enough neighbours
    int longest_dist = farthestNeighbourDistance();
    if (range < longest_dist+2) {
       if (range < MAX_RANGE)
  range++;
    } else {
       range = longest_dist+2;
    }
  } else { // enough neighbours
    int sufficient_range = getSufficientRange(TARGET_NEI_NB);
    if (sufficient_range <= range)
      range = sufficient_range;
    else
      range++;
  }
  sendHello(range); // sends and reschedules hello
  return;
}
```

The API that provides the CNR estimation to the user routing application is given by Algorithm 3. If the overlay node has knowledge of at least the target number of neighbours K, it returns the distance at which the Kst nearest overlay node is located. Else, it returns the minimum value between its current range estimation and the hop distance of its farthest known overlay neighbour.

5.3.2 Evaluation

Performance criteria

The objective of the overlay creation and maintenance is to offer a logical communication structure between the overlay nodes which allows the deployment of efficient overlay routing protocols. From this angle of view, the quality of an overlay is strongly linked to desired properties of overlay routing protocols. We translate this in terms of the following objectives.

1. Bandwidth: as routing control traffic is often generated by flooding,

Algorithm 3 Public interface of ReactiveOtc

```
int
ReactiveOtc::getRange() {
  int r = 0;
  int nei_nb = getNeighboursNumber();
  if (nei_nb >= TARGET_NEI_NB) {
    r = getSufficientRange(TARGET_NEI_NB);
  } else if (nei_nb > 0) {
    int longest_dist = farthestNeighbourDistance();
    if (longest_dist < range)
      r = longest_dist;
    else
      r = range;
  }
  return r;
}
```

the bandwidth necessary to send a message from one overlay nodes to all other ones by using a simple flooding procedure must be as low as possible.

2. Diffusion time: in order to quickly compute valid routes, the overlay control traffic must be flooded rapidly.

3. Delivery percentage: in order to find routes, the overlay control traffic must be received by all overlay nodes.

4. Stretch: the average cost of the shortest overlay path between any pair of overlay nodes must be as close as possible to its value in the underlay. Its maximal cost must also be kept reasonable. We use the hop metric. Other metrics, as for example the path delay, could be considered.

Simulations description

All simulations are realised with ns-2.29.

Overlay nodes are randomly chosen in a set of 100 ad hoc nodes, which are randomly and uniformly distributed on a square field. The length of the field is of 1200 meters and the radio transmission range of the nodes equals 250 meters. All experiments are made for overlay densities of 10, 50 and 90%.

In order to study the performance obtained under mobility, three sets of scenarios are built by the random waypoint scenarios generator provided with the ns distribution, as in [CGT05]. All results with label **slow** on the x-axis, correspond to scenarios generated with a pause time uniformly distributed in $[0, 10]$ seconds and a speed in $[1, 5]$ meters per second. The label **fast** indicates that the pause time is uniformly distributed in $[0, 5]$ seconds and the speed in $[5, 15]$ meters per second. We generate 10 different topologies for each mobility level. The simulation duration is of 150 seconds.

The ReactiveOtc protocol starts on every overlay node at the beginning of the simulation. After 40 seconds, the source overlay node starts to emit 100 overlay messages of 64 bytes, at the average rate of one message per second, and the performance log also commences. The flooding mechanism is the same as described in Sect. 3.2.2.

The underlay routing protocol is AODV. Note however that the ReactiveOtc protocol never sends any unicast message. Hence, as AODV is an on-demand routing protocols, there is no underlay routing traffic.

Results

Performance are presented on Fig. 5.13. The 95%-confidence intervals are specified. As a reference, the performance obtained by the diffusion of a broadcast packet in the whole ad hoc network is shown by the curve with label **underlay**.

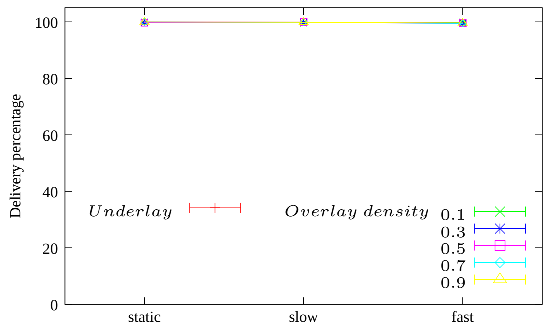
Diffusion delivery percentage For all overlay nodes, except the source of the overlay broadcast, we determine how many overlay messages out of the hundred sent are received before the end of the simulation.

The delivery percentage is excellent.

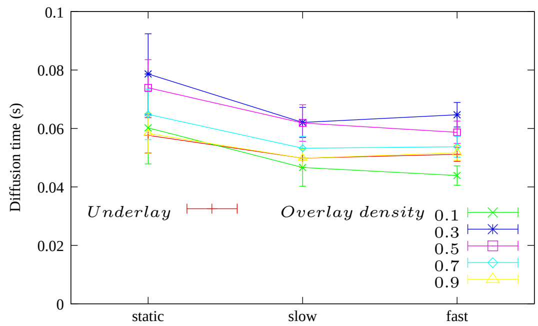
Diffusion time and path stretch For each overlay message flooded on the overlay, we log the interval of time elapsed between its emission and the moment at which its first copy is received by the last overlay node. We also compute the ratio of the number of hops it has passed through since its emission and of the shortest path length from the source. This defines the path stretch.

The overlay diffusion time has the same shape as the underlay diffusion time. It lowers a little when the degree of mobility increases, because mobile nodes speed up the geographic dispersion of broadcast messages. The average overlay diffusion time never exceeds by more than 40% the average diffusion time of a broadcast message on the full underlay.

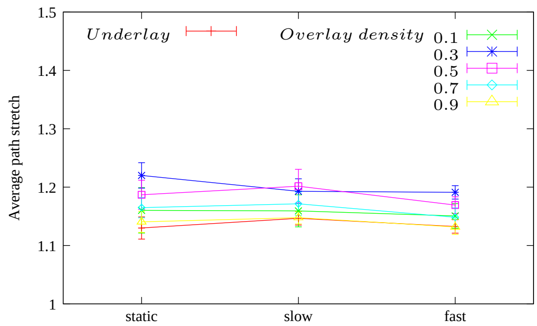
At the lowest overlay density, the overlay diffusion time is even lower than the underlay diffusion time. The overlay message emitted by the overlay source is encapsulated in a broadcast packet with a limited TTL. If it were not forwarded by other overlay nodes, the number of packet collisions provoked by the emission of an overlay message would be much less than by the emission of one broadcast packet with unlimited TTL. However, it would also not be received by all overlay nodes. The overlay nodes thus re-emit



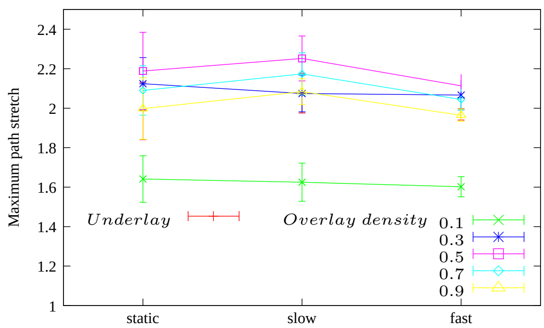
(a) Delivery percentage



(b) Diffusion time



(c) Average path stretch



(d) Maximum path stretch

Figure 5.13: Flooding on the overlay topology built by ReactiveOtc: Performance

the overlay message, on the first time they receive it. The forwarded overlay message is encapsulated in a new broadcast packet with limited TTL and a new broadcast packet identifier. At intermediate overlay densities, the bandwidth consumed may be huge and the collisions frequent. However, for low overlay densities, the overlay message retransmissions are few and may occur simultaneously on different geographic zones. This reduces the number of collisions, the overlay path stretch and the diffusion time of the overlay message.

The average overlay path stretch is shown on Fig. 5.13(c). It is lower than 1.5 for all cases studied. Note that the path stretch for a simple flooding of broadcast packets on the whole ad hoc network is also greater than 1.0 because of collisions.

The maximum overlay path stretch is shown on Fig. 5.13(d). The very low value for the lowest overlay density is noticeable, and due to the creation of different local collisions zones, as explained above.

Bandwidth consumption The bandwidth used, shown by Fig. 5.14, increases for low overlay densities, because the required neighbourhood range is large. It lessens a bit when mobility raises, as the diffusion time, because mobile overlay nodes may relay the information farther than static ones. It is a decreasing function of the overlay density. ReactiveOtc thus provides a NR value adapted to the overlay density. With a constant NR, the bandwidth would have presented a clock shape, with a maximum at middle overlay densities, as observed in our dynamic study of the Overlay-AODV application on Chapter 3 (see Fig. 3.13).

The total bandwidth consumption by constant overlay flooding is huge, compared to the reference, underlay, broadcast. However, the decomposition on control and data traffic indicates that the ReactiveOtc only represents 20% of the bandwidth consumption.

We may thus expect that if the overlay nodes do not require new routes frequently, overlay routing on the topology implicitly defined by ReactiveOtc will provide good performance. Effectively, in this case, the overlay routing protocol does not invoke the flooding procedure too often, and the other performance criteria show that overlay routes will be found with a high probability and the data delivered with a reasonable delay.

5.4 Conclusion

The asymptotic critical neighbourhood range, determined in previous chapter, is not adequate for finite networks. It does not build a connected overlay with a high probability and its computation necessitates the knowledge of the overlay density.

Hence, we explored heuristics for estimating the CNR. Setting a target

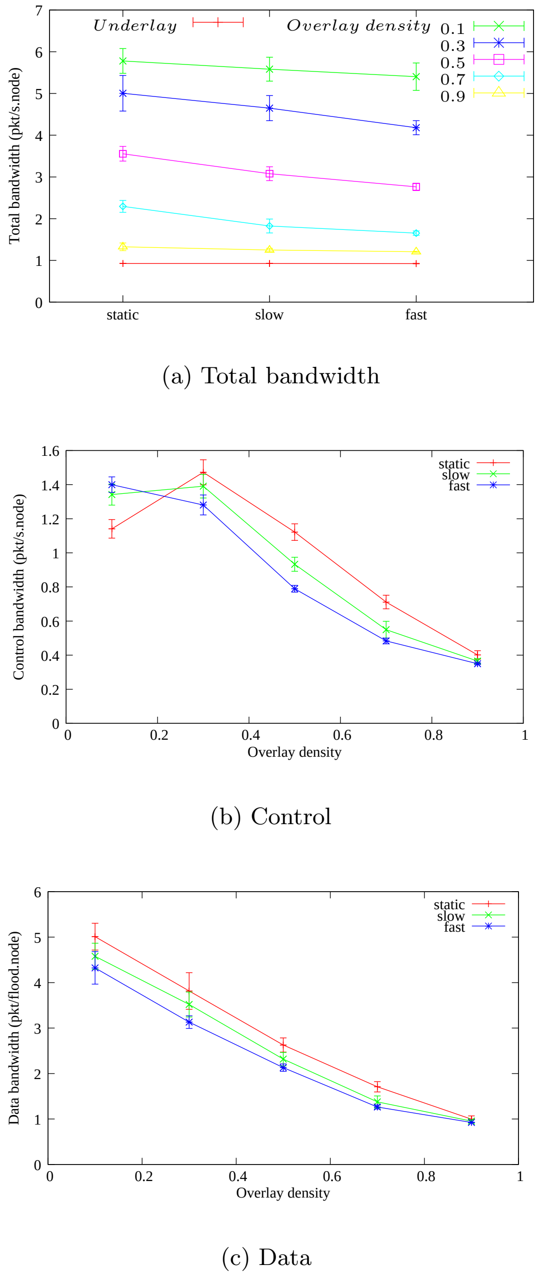


Figure 5.14: Flooding on the overlay topology built by ReactiveOtc: Bandwidth usage analysis

number of overlay neighbours is more reliable than setting a fixed number of hops. It is also less sensitive to the overlay density and to the underlay type.

In order to obtain an overlay where the neighbourhood relation is symmetric, the symmetric closure of the K-nearest neighbour graph is preferable to its reduction. The extension method is expected to consume less traffic for the diffusion of overlay messages. It is also more reliable because the neighbourhood cardinality required for obtaining a connected overlay with a given (high) probability depends less on the number of ad hoc devices and overlay density with the extension than with the reduction rule.

We then presented and evaluated ReactiveOtc. This simple protocol estimates an appropriate neighbourhood range for overlay routing applications. It consists of an increasing ring announcement with a different stop criterion for low and high overlay densities. The stop criterion for low overlay densities is based on a target number of overlay neighbours.

ReactiveOtc can be used in finite ad hoc networks, without any information on the underlay topology nor on the overlay density. The overlay density may evolve with time. However, it is assumed that the local overlay density is nearly constant on the whole ad hoc network. Another assumption is the use of the broadcast overlay messages grabbing architecture, necessary for the stop criterion at high overlay density.

Our performance evaluation utilises general criteria based on overlay flooding. The delivery percentage of broadcast messages represents the probability of finding an overlay route with a reactive overlay routing protocol such as Overlay-AODV, introduced on Chapter 3. The time duration of flooding on the overlay and its path stretch are indicators for data transmission time on the overlay routes found. Finally, as the diffusion of overlay route requests constitutes the major part of the Overlay-AODV traffic, we examined the bandwidth required for overlay flooding as well as the bandwidth consumed by ReactiveOtc.

Chapter 6

The Proactive Overlay Approach

6.1 Study motivation and overview

In previous chapters, we introduced a reactive approach for performing overlay routing. With the reactive approach, overlay neighbours are discovered on-the-fly, during the diffusion of overlay route requests. These are encapsulated in broadcast packets with a limited TTL, called the overlay neighbourhood range (NR). The reactive overlay approach is made possible by the fact that, at any time, the neighbourhood range of each overlay node implicitly defines the overlay topology.

The main drawback of the reactive overlay approach is the amount of bandwidth consumed during the flooding of overlay route requests. The only exception is when the overlay density is high, under the condition that overlay broadcast messages are grabbed by the overlay nodes.

We now consider a different mechanism for the diffusion of overlay messages. Assume that an overlay node must send a broadcast overlay message.[1] It employs the following flooding technique:

1. For all overlay neighbours located only one hop away, it emits a single overlay message, which is actually broadcast in the underlay with a *Time To Live* (TTL) field set to one.

2. For every overlay neighbour located further away, an individual overlay message is created, which will be unicast to it by the underlay routing protocol.

Compared to the overlay flooding technique employed in previous chapters, this mechanism may save a lot of bandwidth, especially when the overlay

[1] The technique is the same for a node that must forward a broadcast message received from an overlay neighbour N, except that the message is not resent to N

density is not high. It also has the advantage of not relying on overlay messages grabbing. However, each overlay node must know the identity and the distance of each of its overlay neighbours. In other words, the overlay topology must be built before the emission of an overlay broadcast message.

We thus now adopt a different approach, where the overlay topology is proactively maintained. Particular attention is given to the emission of a small amount of packets during the diffusion of broadcast overlay messages. We first study the performance of various overlay topologies in the static case. In particular, we present an optimisation technique that selects efficient overlay links, without impairing the overlay connectivity.

We then describe the Overlay Topology Control (OTC) protocol. It maintains, in a mobile context, the overlay topology as close as possible to the overlay topology evaluated as the best.

6.2 Defining a target overlay topology

6.2.1 Methodology

We consider a connected underlay and assume that a routing protocol that builds the shortest symmetric paths is available to all nodes. Overlay nodes are randomly and uniformly distributed on the set of ad hoc nodes. As defined previously, the proportion of overlay nodes is called the overlay density.

Fundamental properties of the overlay topologies studied

The overlay topologies we compare in the following are strongly connected, i.e. there exists a path on the overlay graph between any pair of overlay nodes, at least with a high probability. They can be built by a fully distributed algorithm. We also take care of locality: The topology can be built even if each overlay node is allowed to exchange only a few messages with a limited number of nearest overlay nodes. Locality is an important feature in ad hoc networks because of the limited bandwidth available. We cannot allow the overlay messages to travel along overlay paths much longer than the shortest path in the underlay.

Topologies computation

We first compute underlay topologies with the ad hoc network model and the Random Geometric Graphs (RGG) library presented in Sec.5.2. We only present results obtained with the homogeneous TC model because all observations were similar with the non-homogeneous one.

The ad hoc nodes are randomly and uniformly distributed on a square field. We vary their number from 50 to 250. Overlay nodes are randomly chosen in the set of ad hoc nodes. All experiments are made for overlay densities ranging from 10 to 90%. For the sake of brevity, we only show graphics for the 50% overlay density. Analysis is identical for all overlay densities.

Overlay topologies evaluation

In order to compare the quality of various overlay topologies, we use the performance criteria defined in Sect. 5.3.2, namely the delivery percentage of broadcast overlay messages, the bandwidth and the time consumed for their diffusion and the overlay path stretch. The interference level is not directly addressed. We let the task of reducing interferences to the underlay topology control algorithm and assume that reducing the number of packets emitted per flood is an efficient way to pace collisions due to the overlay use.

The underlay and overlay topologies, calculated offline with the RGG library, are provided as input to the ns-2 simulator. A source node emits 23 overlay messages of 64 bytes, at the rate of one message per second. The underlay routing protocol used is AODV. The performance study ignores the period elapsed during the transmission of the first 3 messages. Over AODV, their flooding necessitates the building of paths between the overlay neighbour pairs. Consequently, the AODV traffic is heavier at the beginning of the simulations and the diffusion time of the first overlay messages is higher than for the following messages. When there is no congestion, the latter must be forwarded on AODV paths that are already up. Each point on the graphics is a mean calculated on 20 trials.

6.2.2 Building topologies that fulfil the locality and connectivity properties

Ropt: The critical neighbourhood range

One simple way to select the nearest neighbours, and thus to respect the locality principle, is to fix the maximal hop distance between overlay neighbours, called in previous chapters the *neighbourhood range*. As discussed in Chapter 4, for any underlay and subset of overlay nodes, one can compute the critical neighbourhood range, that is the minimal neighbourhood range R_C such that the overlay is connected. We denote *Ropt* (R optimal) a topology obtained when each overlay node considers as a neighbour any overlay node that is located at a distance less than or equal to R_C.

Kopt: The critical neighbourhood cardinality

Another simple way to respect locality is to fix the maximal number of overlay neighbours. For any underlay and subset of overlay nodes, one can compute the critical number of overlay neighbours, that is the minimal neighbourhood cardinality K_C such that the overlay is connected. We denote $Kopt$ (K optimal) a topology obtained when each overlay node considers as a neighbour its K_C nearest neighbours, the distance metric being the number of hops. Let k_i be the number of overlay nodes located at i hops from a given overlay node U. If there exists an integer j such that $\sum_{i=1}^{i=j} k_i < K$ and $\sum_{i=1}^{i=j+1} k_i > K$, the required number of overlay neighbours is randomly picked in the set of overlay nodes located at distance $j+1$ from U.[2]

Ropt and Kopt delivery percentage

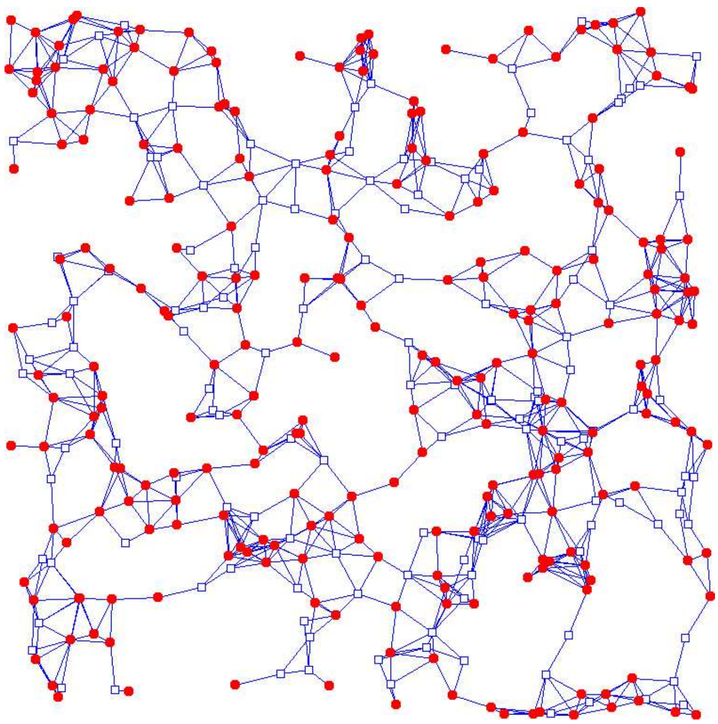


Figure 6.1: An example of Ropt topology ($R_C = 3$)

Figures 6.1 and 6.2 show respectively an example of the Ropt and Kopt overlay topologies for the same underlay. There are 500 nodes and the overlay density equals 50%. The 250 overlay nodes are represented with disks (filled with red). The remaining nodes, represented with empty (blue) squares, are drawn if and only if they are on the shortest path between a pair of overlay neighbours. For this particular underlay and assignment of overlay nodes, the critical neighbourhood range equals 3 and the critical neighbourhood cardinality equals 4. This figure also illustrates that the Ropt

[2]We evaluated some more sophisticated policies, but none provided significantly better performance

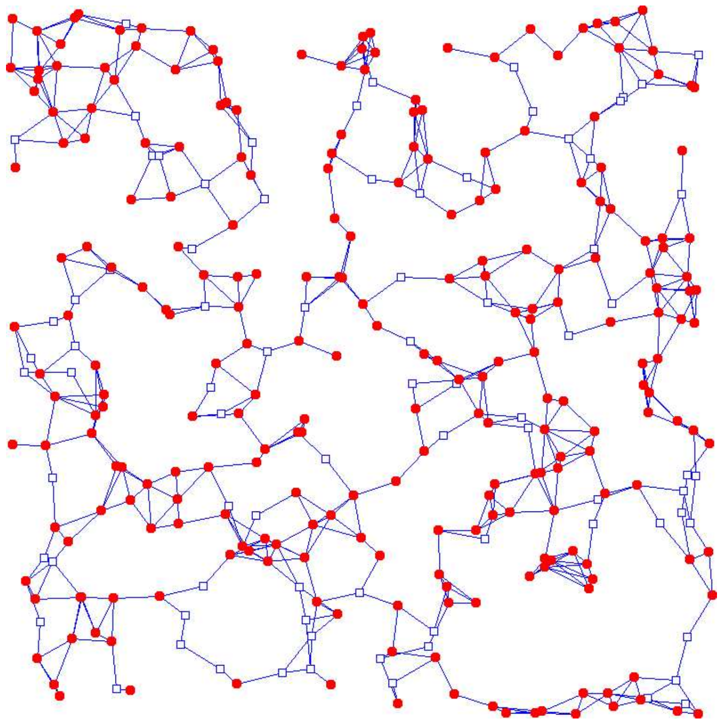


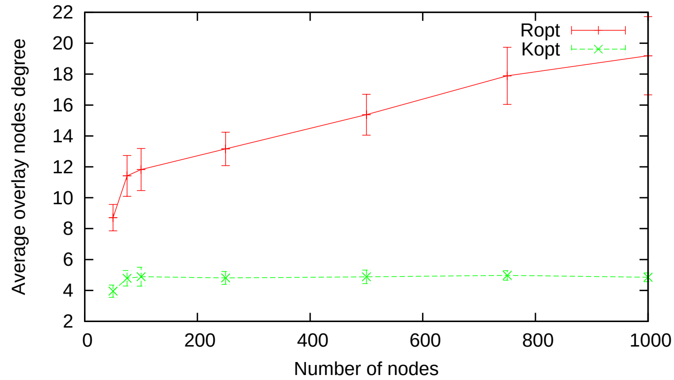
Figure 6.2: An example of Kopt overlay topology ($K_C = 4$)

overlay topologies are much denser than the Kopt ones. This is confirmed on Figure 6.3(a) which shows the average number of overlay neighbours per overlay node, that we call the overlay nodes degree.[3] The high overlay nodes degree of *Ropt* topologies explains their weaker delivery percentage for flooded messages, as illustrated by Fig. 6.3(b) for an overlay density equal to 50%. Congestion problems arise for a moderate amount of overlay nodes. These become severe on top of AODV because paths are maintained by intermediate nodes if and only if they are regularly traversed by data. After the loss of consecutive messages, a new message arriving at an intermediate overlay node could trigger a discovery procedure, and amplify the network load.

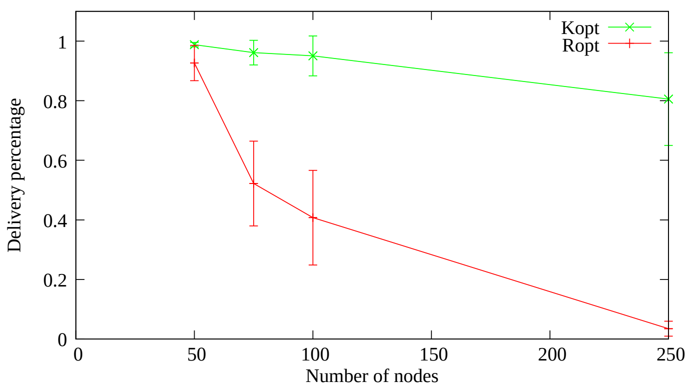
KNN: The empiric neighbourhood cardinality

$Kopt$ topologies provide better delivery percentages but are difficult to build in practice. As discussed in Chapter 5, there is no analytical function that gives the optimal number of overlay neighbours needed for connectivity in finite networks. One could imagine a distributed algorithm that determines K_C. For example, the algorithm employed in [NKSK02] for electing the best

[3]The average overlay nodes degree is above $K_C = 4$ because we apply the extension rule, as defined on Sec. 5.2.5.



(a) Average overlay nodes degree



(b) Average delivery percentage

Figure 6.3: Overlay flooding delivery percentage and average overlay nodes degree, for Ropt and Kopt topologies

radio transmission range could be adapted. However, this would require the exchange of a lot of information in the whole network. We reject this solution because of its high bandwidth demand.

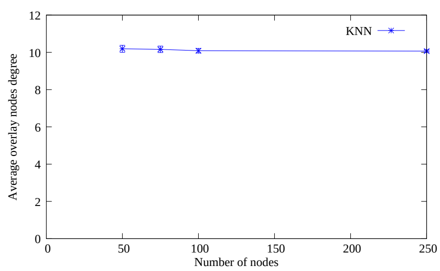
In previous chapter, we determined that setting a target number of overlay neighbours is a good heuristic for building local overlay topologies, connected with a high probability. An extensive set of simulations allowed us to determine empirically a parameter K^{95} that assures with a probability higher or equal to 95% the overlay connectivity for a wide range of ad hoc network sizes and overlay densities. We determined that setting a target number of overlay neighbours is more reliable than setting a fixed number of hops. This heuristic is also less sensitive to the overlay density and to the underlay type.

We denote KNN the symmetric closure of the K-nearest neighbour graph. It is the overlay topology obtained with the extension rule defined on Sec. 5.2.5, i.e. when each overlay node N considers as overlay neighbours:

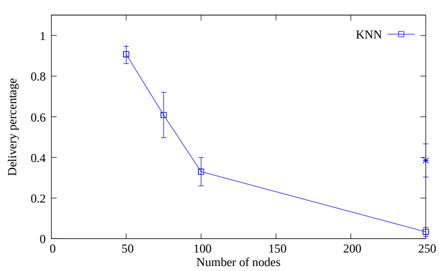
- its K nearest overlay nodes, and
- every overlay node that has N in its K-nearest overlay nodes set.

The overlay nodes degree of KNN topologies is shown on Fig. 6.4(a). It is obviously higher than the one of $Kopt$ topologies. The corresponding delivery percentage is given by Fig. 6.4(b). The KNN topologies, as the *Ropt* topologies, are too dense. The delivery percentage obtained on these overlay topologies with the flooding technique considered in this Chapter is too low.

The KNN overlay topologies present however several advantages. They are local, connected with a high probability, and can be built with a sustainable volume of control traffic. We thus explore in the next section methods for eliminating edges while preserving the connectivity property.



(a) Average overlay nodes degree

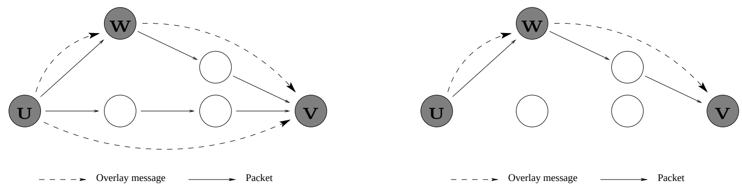


(b) Delivery percentage

Figure 6.4: Average overlay nodes degree and overlay flooding delivery percentage for KNN topologies

6.2.3 Optimising the topologies for overlay routing

Shortest Path Pruning



(a) Kopt topology without SPP (b) Kopt topology with SPP

Figure 6.5: Motivation for the Shortest Path Pruning. Thick arrow = overlay message, thin arrow = packet.

Figure 6.5 shows an example of the flooding of an overlay message from U and the corresponding underlay packets emitted. The overlay nodes (U, V and W) are grey-shaded. In fig. 6.5(a), the Kopt overlay topology is used; it is composed of the three edges (U, V), (V, W) and (U, W). The flooding of the overlay message on this Kopt topology generates 6 packets on the underlay. However, as illustrated in fig. 6.5(b), the propagation from U to V, followed by the forwarding from V to W would have been sufficient for all overlay nodes to receive the messages and would have generated only 3 packets. The longest edge of the triangle is thus unnecessary.

Hence, we introduce the following *Shortest Path Pruning*. Consider three overlay nodes U, V and W, and a distance metric d. The distance metric can be the hop count, the path average delay or any other real positive and symmetric function. Assume that the edge (U, W) is the longest. We have $d(U, V) \leq d(U, W)$ and $d(V, W) \leq d(U, W)$. The Shortest Path Optimisation sets aside the edge (U, W) if and only if $d(U, V) + d(V, W) \leq d(U, W)$. It preserves the connectivity of any overlay graph because an overlay edge is suppressed if and only if an alternative path exists on the overlay.

Maximal Pruning

Shortest Path Pruning improves the delivery percentage of flooded messages on KNN topologies. However, this pruning method is not sufficiently selective. It can be generalised by setting aside any overlay edge (U, W) such that $d(U, V) + d(V, W) \leq \alpha d(U, W)$, with $\alpha >= 1$. Connectivity is still preserved.

The higher value is assigned to α, the more edges are pruned. We call this parameter the *pruning selectivity*. Maximal Pruning is reached when any edge (U, W) is suppressed as soon as there exists two shorter edges (U, V) and (V, W). This behaviour is already obtained for $\alpha = 2$. The inequality $d(U, V) + d(V, W) \leq 2d(U, W)$ is always satisfied because the edge (U, W) is the longest.

Let us make the distinction between the one-hop overlay neighbours, or *broadcast neighbours*, and the overlay neighbours located farther, the *unicast neighbours*. The emission of only one broadcast packet is sufficient for an overlay flooded message to reach all the broadcast neighbours. Thus, keeping all broadcast neighbours does not increase the bandwidth consumed per overlay flooding. On the other hand, it increases the density of the final overlay, without increasing the number of unicast neighbours of any overlay node. The consequence is a lower diffusion time and stretch. It also improves the overlay resilience. We thus modify a little the generalised rule in order to maintain as neighbours every pair of overlay nodes located at one hop from each other.

Therefore we finally define the following generic pruning rule.
Consider three edges $E_1 = (U, V)$, $E_2 = (V, W)$ and $E_3 = (U, W)$, a distance metric d, and a real number α with $1 \leq \alpha \leq 2$. Assume E_3 is the longest edge.
Edge E_3 is pruned if and only if:

1. E_3 is longer than one hop, and
2. $d(E_1) + d(E_2) \leq \alpha d(E_3)$.

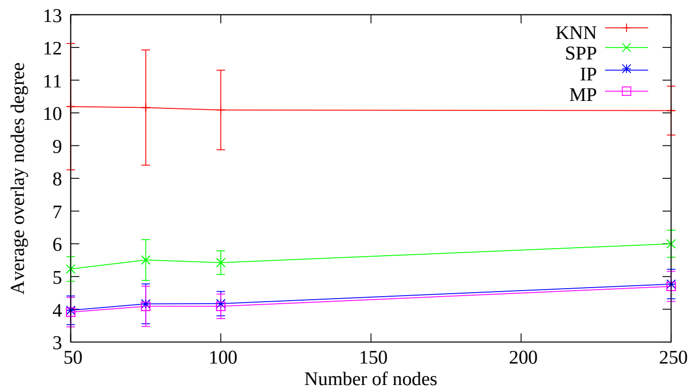
Figure 6.6 shows the overlay nodes degree and delivery percentage for various pruning selectivity on KNN overlay graphs. The distance metric used is the hop count. For the intermediate pruning selectivity, denoted by IP, parameter α is set to 1.5. The delivery percentage increases with the selectivity of the pruning method. It is correlated with the average number of overlay neighbours. Flooding an overlay message consumes much bandwidth. Congestion is avoided on sparse overlay graphs.

The average overlay nodes degree of KNN overlay topologies with Maximal Pruning is above 4, with a tight 95%-confidence interval. Maximal Pruning thus preserves some resilience on KNN overlay topologies. Note that resilience is also provided by the underlay topology and routing protocol. The underlay often offers several different paths between each pair of overlay nodes, and a new route can be built when a path between two overlay neighbours breaks.

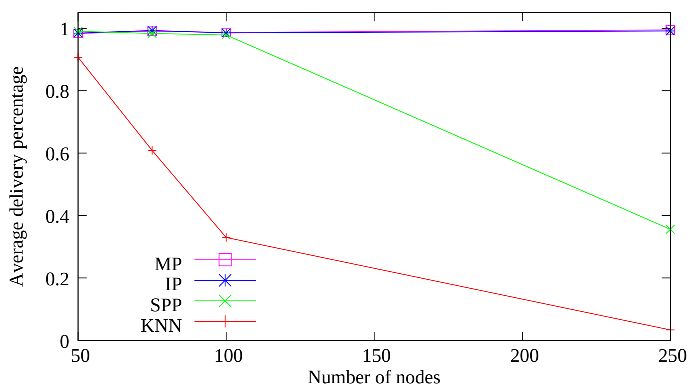
6.2.4 Final comparison of overlay topologies

A brief comparison with XL-Gnutella

We do not criticise the XL-Gnutella protocol, which is intended to be used for P2P data search, not for overlay routing applications. The point here is



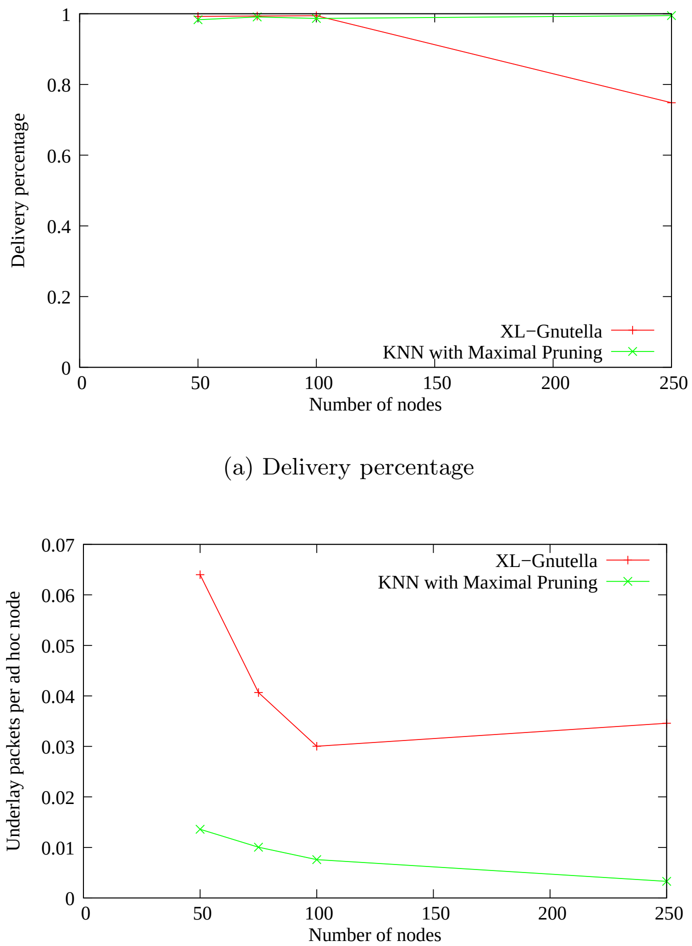
(a) Overlay degree



(b) Delivery percentage

Figure 6.6: The average overlay nodes degree and delivery percentage of KNN overlay topologies pruned with various selectivity factors

to show the utility of our own work in the context of overlay routing.



(a) Delivery percentage

(b) Bandwidth

Figure 6.7: XL-Gnutella overlay topologies are intended to be used in a P2P networking context, not for overlay routing

XL-Gnutella is an optimisation of the Gnutella protocol for ad hoc networks. In order to remain fully compatible with the legacy Gnutella protocol, an overlay edge selection algorithm maintains the number of neighbouring peers between 4 and 8.

We compare on Fig. 6.7(a) the average delivery percentage of flooded messages on XL-Gnutella and on KNN with Maximal Pruning overlay topologies. Recall that the homogeneous underlays we use for our simulations are computed with the model presented in Sec.5.2. The radio transmission range used is thus, for each underlay, the minimal value that makes it connected.

On these very sparse underlays, forcing every overlay node to reject neighbours once the overlay node degree has reached the highest watermark of 8, as XL-Gnutella does, leads to a lower connectivity probability than for KNN overlay topologies, for which such restriction does not exist. For the same reason, some overlay edges are longer in XL-Gnutella than KNN topologies. This increases a lot the bandwidth required per overlay message flooding (Figure 6.7(b)). We also expect, when the underlying routing protocol is reactive, the discovery of XL-Gnutella topologies to consume much more bandwidth than the discovery of KNN topologies, again because some overlay neighbours are selected very far away. In the XL-Gnutella paper, authors use a proactive routing protocol, OLSR, and a cross-layer architecture that allows the P2P middleware to be aware of every overlay node identity and distance, with a low bandwidth consumption. They mention that experiences were also successful with AODV, but that results are better with OLSR.

Comparison of Kopt and KNN with Maximal Pruning

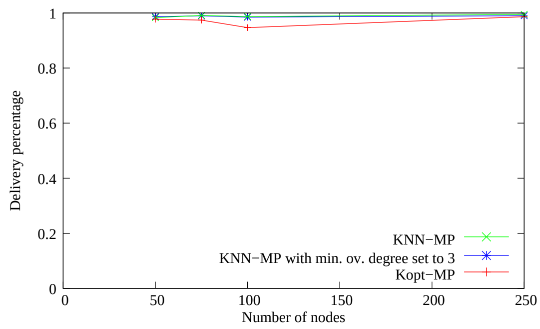
The performance of flooding a message on KNN and Kopt with Maximal Pruning topologies are compared on Figure 6.8. These are similar, which indicates that the use of the empirical value $K = 8$ before optimisation, common for all simulations, instead of the exact minimal number of nearest neighbours needed for overlay connectivity K_C, which value must be determined for each simulation, is not a handicap.

We can also observe that the flooding of an overlay message, which can collect and propagate interesting information for the overlay routing applications, does not consume much more bandwidth than the flooding of a packet on the underlay (exactly 1 packet per node). Note also the reasonable value of the overlay path stretch.

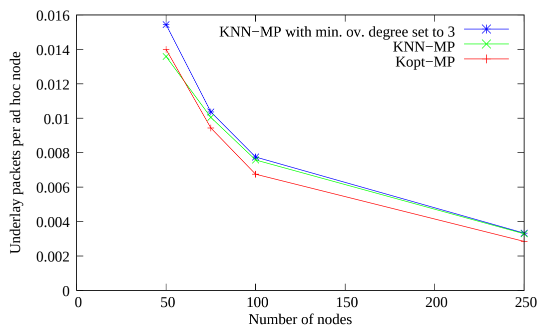
Improving resilience

One could use an intermediate value for α instead of Maximal Pruning, for the purpose of improving the overlay topology resilience. Performance obtained on the KNN topologies pruned with $\alpha = 1.5$ and $\alpha = 2$ for instance are very close (their delivery percentage is compared on fig. 6.6). However, the gain in resilience is difficult to quantify.

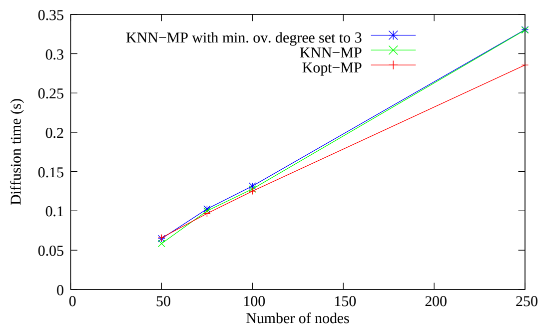
Setting a minimum overlay node degree is another way to increase the redundancy of the overlay, is easier to evaluate and simple to implement. In some cases, it is even required. This is the case, for example, when one wants to deploy multipath routing on the overlay. A minimal number K_{min} of overlay neighbours is easily guaranteed by reading the nearest overlay nodes list in increasing order of distance and beginning to apply the pruning rule only at the $K_{min} + 1$ element. On Figure 6.9, we also compare



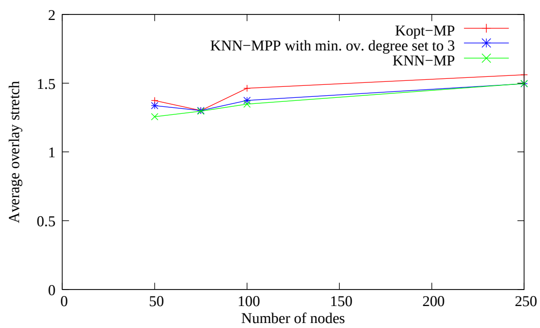
(a) Delivery percentage



(b) Bandwidth



(c) Diffusion time



(d) Stretch

Figure 6.8: After pruning, flooding a message on KNN overlay topologies provides similar performance results than on Kopt ones.

the performance obtained with Maximal Pruning on KNN topologies when applying the pruning rule to the 3 nearest overlay nodes and when systematically keeping them in the final neighbourhood.

6.3 The Overlay Topology Control protocol

In the first part of this chapter, we have compared the performance of various overlay topologies in the static case. We noticed, in particular, the good properties of the K-Nearest Neighbours overlay topology with Maximal Pruning and a minimal bound on the overlay node degree.

In this section, we present the Overlay Topology Control protocol (OTC) that keeps, in a mobile environment, a set of overlay links as close as possible to this target overlay topology.

6.3.1 Description

Assumptions and protocol overview

We consider a connected ad hoc network, the underlay. The underlay routing protocol is supposed to provide short paths, but not necessarily the shortest ones and may build asymmetric paths. We do not assume that the underlay routing protocol is able to inform the above layer about the length of the available paths. We make however the following, weaker, assumption: When a node receives a packet, it is able to know how many hops the packet has traversed since its emission.

The OTC algorithm is fully distributed and local, i.e. each overlay node exchanges only a few messages with a limited number of nearest overlay nodes. It avoids logical long-range neighbours because of their prohibitive maintenance cost. The overlay topologies built are connected, at least with a high probability.

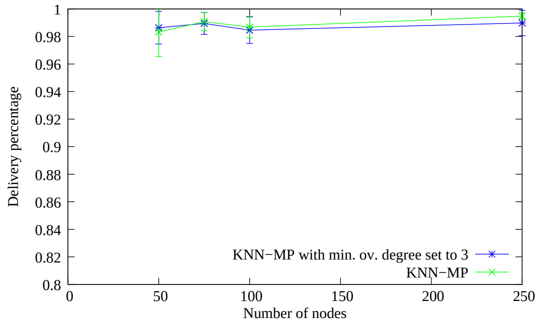
We differentiate broadcast and unicast overlay neighbours, as in Sec. 6.2.3. The former are also physical neighbours, i.e. there exists a direct radio communication link between them, while the distance between the latter is at least of two hops.

Each overlay node U:

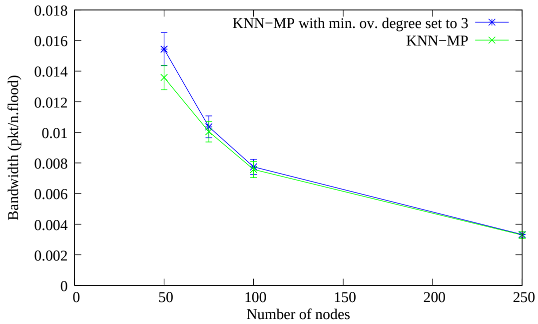
1. Collects in its neighbour candidates list L_U the identifier and the shortest distance to its K closest overlay nodes.

2. Also inserts in L_U any overlay node V such that $U \in L_V$ (thus turning the neighbourhood relation into a symmetric relation).

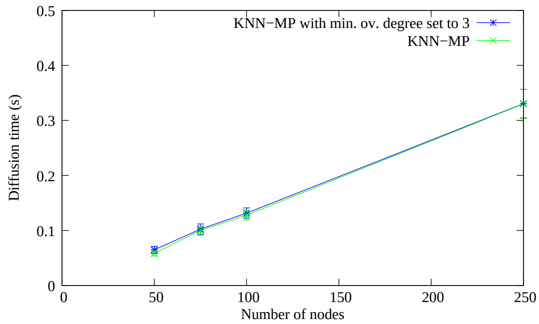
3. Selects its active neighbours in L_U by applying the pruning rule described in Sec. 6.2.3.



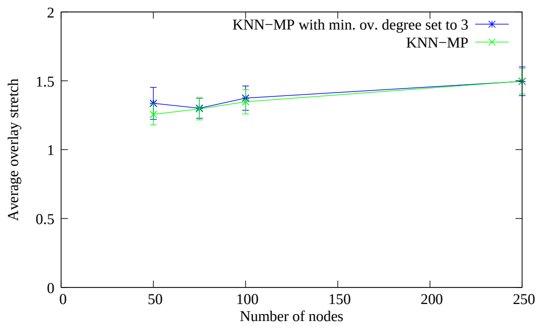
(a) Delivery percentage



(b) Bandwidth



(c) Diffusion time



(d) Stretch

Figure 6.9: Setting the minimal overlay degree to 3 does not modify significantly the performance obtained on KNN with Maximal Pruning.

Table 6.1: OTC messages emitted by a node U (to a node V, for unicast messages)

Message type	Emission mode	IP TTL	Message content
OTC_HELLO	Broadcast	1	U
OTC_REQUEST	Broadcast	$R_U - 1$ ([4])	U
OTC_REPLY	Unicast	$NETD$	U
OTC_ADVERTISE	Unicast	$NETD$	U, L_U, m_V
OTC_DELETE	Unicast	$NETD$	U

U = overlay identifier of the sender
R_U = current overlay range of the sender
$NETD$ = estimated network diameter
L_U = neighbours list of the sender
m_V = monitoring state for the overlay link between the sender and the receiver V

4. For resilience, sets all broadcast neighbours active, and does not prune the overlay links established with the 3 nearest overlay nodes.

We observed in Sec. 6.2.2 that setting K to the value of 8 was sufficient to guarantee connectivity of overlay topologies with a probability higher than 95% for up to 1000 underlay nodes and overlay densities ranging from 10 to 90%.

The data messages only flow onto active neighbours, i.e. on active overlay links. Oppositely, OTC control messages are continuously exchanged with pruned neighbours as well as with selected ones.

Each overlay node must update its neighbour candidates list as soon as feasible when nodes move. Possible updates are adding, sorting and deleting elements. The overlay nodes must also determine if a neighbour candidate must be selected as neighbour or pruned.

We begin the OTC protocol description with the simple maintenance procedure of broadcast neighbours. Table 6.1 describes the control messages used by OTC.

The discovery and maintenance of broadcast neighbours

Each overlay node regularly emits an OTC_HELLO message, encapsulated in a broadcast packet with the Time To Live (TTL) field set to one. If a node U receives an OTC_HELLO message from a node V, it adds V to its neighbour candidates list L_U. Every broadcast neighbour is automatically selected as overlay neighbour. Broadcast neighbours are purged if no hello message has been received during a given time interval.

If a node U has less than K broadcast overlay neighbours, its neighbour candidates list must be supplemented by overlay neighbours located further.

The necessary unicast overlay neighbours will be selected among them. The rest of this section is devoted to this more complicated part of the OTC algorithm.

The discovery of new unicast neighbour candidates

As soon as a node enters the overlay, it regularly emits an OTC_REQUEST in broadcast packets. The Time To Live (TTL) field of these packets is set to increasing values, beginning from 2, until L_U gets sufficiently long (at least K neighbour candidates).

An overlay node V that receives an OTC_REQUEST from U responds with a unicast packet containing an OTC_REPLY if and only if node U is not already in L_V.

The neighbour candidates list is sorted by increasing distance and, when distances are equal, by increasing identifier. When node U receives the OTC_REPLY from node V, it calculates at which position it would insert V in L_U, using the number of hops the OTC_REPLY has passed through and on the identifier of V. If the position is less than or equal to K, it inserts V in its neighbour candidates list, sets its monitoring state for V on, and sends a unique OTC_ADVERTISE message to V. At this point of the protocol, this message is used by U for forcing V to create the (U, V) pair of neighbour candidates. This ensures the symmetry of the neighbourhood relationship.

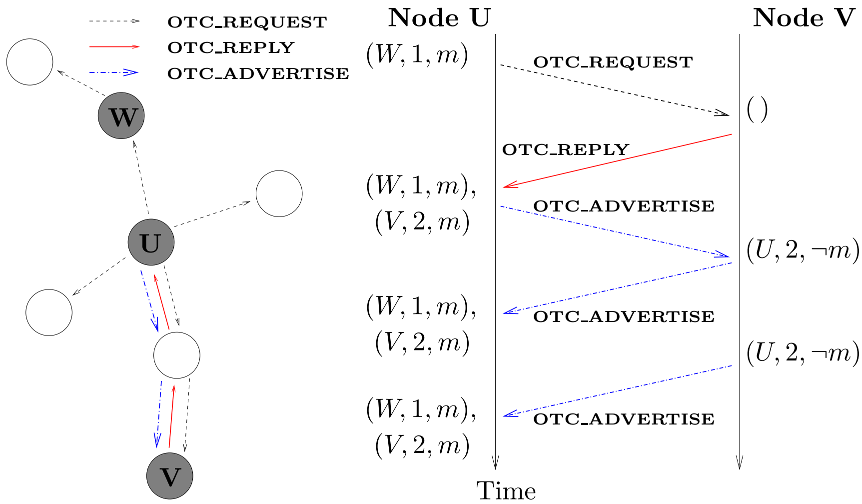
Node U maintains a monitoring state for each of its neighbour candidates. If node U monitors node V (i.e. its monitoring state for node V is on), this means that node U is responsible for estimating the distance $d(U, V)$, and communicating any change to V. The complete distance update process is described below, on Section 6.3.1.

If $U \notin L_V$ when node V receives the first advertisement from U, node V inserts U in its neighbour candidates list, and registers the distance $d(U, V)$ announced in L_U, contained in the advertisement. It sets its monitoring state for U off and begins to send OTC_ADVERTISE messages to U at regular intervals. The reception of these frequent advertisements by node U makes him capable of monitoring the distance between U and V, by observing the number of hops they traversed. At stability, there is one and only one end node per candidate overlay link that monitors its length.

A simple discovery process is illustrated on Fig. 6.10. At the beginning, $L_U = ((W, 1, m))$, which means that U only knows one broadcast neighbour W and monitors it[5]. At the end, U and V have been inserted in the neighbour candidates list of each other, with distance $d(U, V) = 2$. Node U monitors node V, V does not monitor U (indicated in the Fig. 6.10(b) by the expression $\neg m$). Note that the distances registered in L_U and L_V

[5] The monitoring state for broadcast neighbours is not used by the algorithm and set on by default.

are equal because it corresponds to the number of hops traversed by the reply (and the advertisements from V to U), even if the request or the first advertisement has followed an underlay path of different length.



(a) OTC messages

(b) Corresponding evolution of the neighbours lists along time

Figure 6.10: OTC discovery procedure

The maintenance of the neighbour candidates list

Consider an overlay node U. In a mobile context, the content of its ordered neighbour candidates list, L_U, evolves continuously. The set of K closest overlay nodes of U and the distance between U and a member of L_U may change. Moreover, overlay node U may enter or leave the set of K closest overlay nodes of other overlay nodes.

New unicast overlay neighbour approaching Consider a node U that already knows at least K neighbour candidates. Let us define its current neighbourhood range R_U by the distance registered for the Kth element of L_U. If $R_U \leq 2$, interesting new neighbours are at most one hop away, and their discovery will be done through the reception of their hello messages. When $R_U > 2$, in order to spot approaching unicast overlay nodes, node U regularly emits new OTC_REQUEST messages. The TTL field of the broadcast packet containing these requests is set to $R_U - 1$. As U owns a sufficient number of neighbour candidates, new overlay nodes located R_U hops or farther are not considered of better quality than the elements of

L_U and must not be sought. The process induced by the reception of these requests is the same as described above.

Distance update The updates are made possible by the regular emission of OTC_ADVERTISE messages. An overlay node U sets an advertisement timer for each of its neighbour candidates. The advertisement timer set for a node V is reset every time U sends an advertisement to V or receives an advertisement from V.

An overlay node assigns a longer expiration time to candidates it monitors than to candidates that it does not monitor. The difference of expiration time is such that, if there is no advertisement loss nor distance change, an overlay node only receives regular advertisements from neighbour candidates that it is monitoring. If it notices a distance modification, it sends a unique OTC_ADVERTISE to the corresponding, unmonitoring, peer. The overlay nodes thus also receive asynchronous advertisement from neighbour candidates that they are not monitoring, for being informed of any modification of the corresponding candidate overlay links.

If there are losses or if both nodes of a neighbour candidates pair are in the monitoring state advertisements are still received. The latter case may appear in transient scenarios, caused by mobility or at set up. For example when overlay nodes U and V discover the existence of each other in a short interval of time, by the reception of two OTC_REQUEST messages sent in opposite direction.

When an overlay node U receives an advertisement from V on a path of h hops, it first checks its monitoring state for V and updates $d(U, V)$:

- If node U is not monitoring V, it sets its local $d(U, V)$ variable to the value indicated in the list L_V, contained in the advertisement (see Tab. 6.1).

- If node U is monitoring V, it first verifies in the advertisement that V is not also monitoring U. If U and V are monitoring each other, a tie function is applied on their identifiers in order to elect the monitoring overlay node. If U stops monitoring V, it reacts as described above. If it continues to monitor V after the check, it sets $d(U, V)$ to the value h, the number of hops traversed by the advertisement.

If $d(U, V)$ has changed, U corrects the position of V in L_U. If it is monitoring V, it also directly sends a new advertisement to V in order to inform it about the distance update (or, as explained in next section, a delete message indicating that the neighbour relation is no more useful).

Old unicast overlay neighbour leaving Once the distance update process is completed, node U calculates the new position it would occupy in L_V consequently to the distance update. On the basis of the updated lists

L_U and L_V, it can then detect if V is still required or not in L_U. If the position of V in L_U and of U in L_V are both greater than K, then the neighbourhood relationship between U and V is no more necessary. Node U deletes V from L_U and cancels its advertisement timer for V. It also sends an OTC_DELETE message to V. If the delete message is lost, it will be sent again at reception of the next advertisement from V.

If node V remains in L_U, node U can then determine if it must select V as a neighbour or prune it.

The pruning of neighbours in the list Consider an overlay link (U, V), with U elected as the monitoring node. Node U receives at regular intervals an advertisement that contains L_V. It prunes the overlay link (U, V) if and only if there exists a third overlay node W that appears before V in L_U and before U in L_V, and such that $d(U, W) + d(W, V) \leq \alpha d(U, V)$. Parameter α has been defined in Sec. 6.2.3 as the *pruning selectivity*. It is a real number in the interval $[1, 2]$.

The idea is, for every pair of overlay neighbours (U, V) to prune each other if they share a common better neighbour, under the condition that the remaining overlay path that links them is not stretched by more than a factor α[6]. The pruning rule does not affect the overlay connectivity. It reduces significantly the bandwidth consumption during flooding and decreases contention, while increasing the overlay path stretch and diffusion time by an acceptable amount.

The pruned neighbours are not deleted from the candidates list. OTC control messages are continuously exchanged with pruned neighbour candidates as well as with selected ones. Oppositely, the overlay data will only flow on the underlay paths that link selected pairs of overlay neighbours.

Summary of the advertisements role

Advertisements are sent asynchronously anytime an overlay link is modified. Their purpose is to give to the two end nodes a consistent view of the (local) overlay topology, so as to make their decisions consistent. They contain the local neighbours list, sorted by increasing distance and, when distances are equal, by increasing identifier. Modification events are the creation of the overlay link, and update of its length or state. An overlay link may be in the active or pruned state..

The behaviour of a node U when it receives an OTC_ADVERTISE from V is sketched by Program 6.11.

[6] Note however that the final maximal overlay stretch may be higher than α because the pruning rule is also applied to (U, W) and (W, V), and so on.

```
OTC::recvAdvertisement(adv,sender,hops) {

  V = neighbour_lookup(sender);
  if (V == NULL) {
    /* 0. Special behaviour if sender is not
       in the neighbour candidates list */
      ...
  } else {
    /* I. Check monitoring state */
    // Does V monitor U ?
    bool monitored =  readMonitoringInfo(adv);
    // Does U monitor V ?
    bool monitoring = V->isMonitored();
    assert(monitored || monitoring);
    if (monitored && monitoring) {
      if (sym_select(U,V) == V)
         unsetMonitoring(V);
    }

    /* II. Update distance in local list */
    rcvd_dist = adv->readDistanceInfo(U);
    if (V->isMonitored()) {
      updateDistance(V,hops);
    } else {
      updateDistance(V,rcvd_dist);
    }

    /* III : purge check */
    purge = purgeProcess(V,adv);
    if (!purged) {

      /* IV : pruning or selection of V */
      bool state_modified = pruningProcess(V,adv);

      /* V : inform V about any modification */
      local_dist = V->getDistance();
      if ((local_dist != rcvd_dist) ||
          (state_modified))
         sendAdvertisement(V);

      /* VI : reschedule next advertisement later */
      rescheduleAdvertisementTimer(V);
    }
  }
}
```

Figure 6.11: Behaviour of node U when it receives an OTC_ADVERTISE from V

Topologies		**Scenarios**		
Number of nodes	100	**RWP**	Speed (m/s)	Pause Time (s)
Field length	1200 m	Static	0	150
Communication range	250 m	Slow	$[1, 5]$	$[0, 10]$
Overlay density	0.1, 0.5, 0.9	Fast	$[5, 15]$	$[0, 5]$

Table 6.2: Simulation parameters

6.3.2 Evaluation

We evaluate the OTC with the same methodology that we used for ReactiveOtc, detailed in Sec. 5.3.2.

As OTC is not application-specific, its evaluation uses indirect performance criteria based on overlay flooding. The diffusion of a message can be seen as a worst-case scenario for group communication. It is also a key component of many unicast route discovery mechanisms in MANETs.

We use the ns-2.29 simulator and its random waypoint (RWP) scenarios generator. Parameters used are given in Tab. 6.2. The OTC protocol starts at the beginning of the simulation, with a pruning selectivity set to 1.5. After 30 seconds, a source overlay node starts to emit 100 broadcast overlay messages of 64 bytes, at the average rate of one message per second.

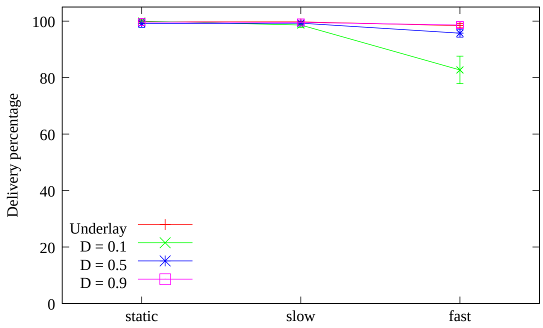
We present the performance of OTC over the Optimised Link State Routing protocol (OLSR) [JMC+01], a proactive routing protocol. We also ran simulations over the Ad-hoc On-demand Distance Vector routing protocol (AODV). Results proved that the OTC protocol can be applied over reactive as well as over proactive routing protocols. However, we observed that the performance obtained over AODV rapidly degrades when the nodes move, except for the lowest overlay density. The reason is that AODV cannot sustain the maintenance of many paths in a mobile context. Proactive routing protocols are less efficient than AODV when the traffic pattern is light, because the paths between each pair of ad hoc nodes is maintained even if only of few of them are required by the users. However, their routing load is far less sensitive to the number of user flows. When these are numerous, proactive routing protocols outperform reactive ones.

The average performance obtained when flooding an overlay message on the topologies built by OTC over OLSR are shown on Fig. 6.12. The 95%-confidence intervals are specified.

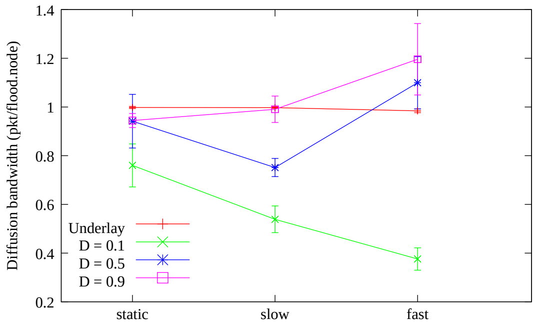
Diffusion delivery percentage

For all overlay nodes, except the overlay broadcast messages source, we determined how many overlay messages out of the hundred sent were received before the end of the simulation.

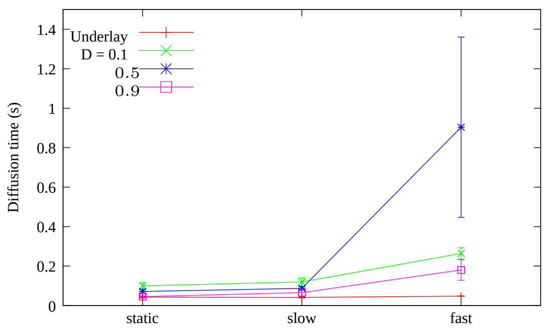
The delivery percentage over OLSR is very good in the static and slow



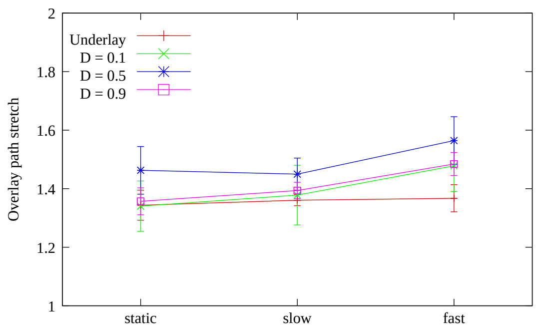
(a) Delivery percentage



(b) Data bandwidth



(c) Diffusion time



(d) Path stretch

Figure 6.12: Overlay flooding performance over OLSR

case, for all overlay densities. When nodes move a lot and rapidly, the delivery percentage is also very good (above 94%), except for the lowest overlay density (only 83% of the overlay nodes receive the flooded message). Noticeably, in the latter case, we obtained very good performance over AODV because the number of overlay links is low. In such situation, it has been shown in [CJV02] that AODV resists better to mobility than OLSR. Hence, we assume that the performance drop is due to OLSR and not to OTC.

Bandwidth consumption

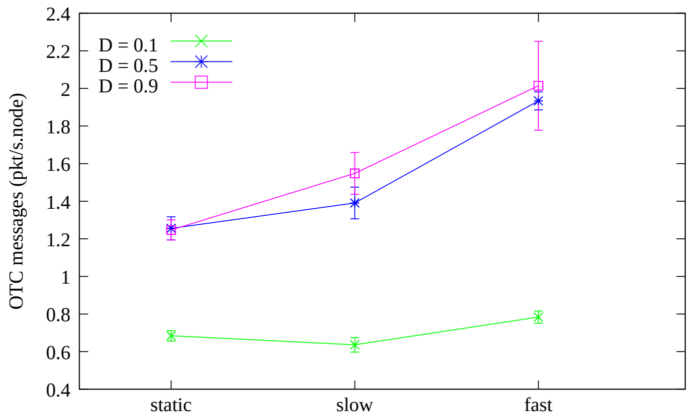
We logged the number of packets emitted during the simulation. We classified them into three categories: the packets containing the 64 bytes of data, the OLSR control messages and the OTC control messages. The control bandwidth includes both OLSR and OTC messages. The metric used on Fig. 6.12 is the number of packets emitted per flood and per ad hoc node.

The total bandwidth amount increases reasonably with the degree of mobility. At very low and intermediate overlay densities, the flooding of a message on the overlay, which is able to propagate useful information for overlay routing, costs less than one packet per node. Hence, it does not consume more bandwidth than a legacy flooding on the underlay of a broadcast packet (whose cost equals approximately one packet per broadcast and per node). For all cases studied, we observed that the OTC traffic occupies less than 40% of the total amount of control traffic. Figure 6.13 shows that maintaining an overlay with OTC over OLSR costs less than 60% the bandwidth needed by OLSR for maintaining underlay routes (the maximal value on the y-axis is higher for the right sub-figure). The metric used is the number of packets emitted per second and per ad hoc node.

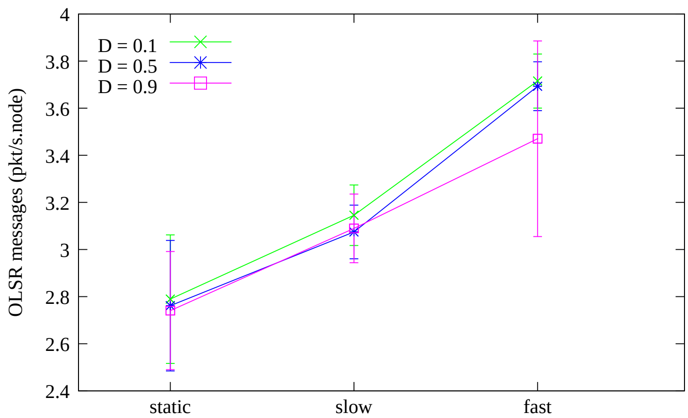
Overlay diffusion time

For each overlay message flooded on the overlay, we logged the interval of time elapsed between its emission and the moment at which its first copy was received by the last overlay node. This measures the maximum time that would be needed for an overlay route request to reach any destination. As a reference, we also quantified, for every mobility case, the diffusion time on the underlay, i.e. the interval of time elapsed between the emission of a broadcast packet on the underlay and the reception of its first copy by the last ad hoc node. It is represented by the curve identified by the label "underlay".

The overlay diffusion time raises with the mobility degree. The sharper increase is observed at the middle overlay density because of our flooding policy, where each overlay node must emit a new message for each of its unicast neighbours. At a high overlay density, most neighbours are located at one hop (they are broadcast neighbours). At low density, there are a few



(a) OTC



(b) OLSR

Figure 6.13: Comparison of the bandwidth consumed by OTC and OLSR

overlay links, thus the time required for accessing the media is lower.

Overlay path stretch

Each time an overlay message was received, we also computed the ratio of the number of hops it has passed through since its emission and of the shortest path length from the source. This defines the overlay path stretch. Its average value is under 1.6 for all cases studied. Note that the path stretch of broadcast packets flooded on the whole ad hoc network is also greater than 1.0 because of collisions.

6.4 Conclusion

We first discussed what kind of overlay topology should be pro-actively built before an overlay routing protocol enters a route search process on top of it. We then described and evaluated a protocol for building and maintaining them.

We introduced a family of optimisation rules of the K-Nearest Neighbours topologies, based on a pruning rule. As flooding is a key component of many route discovery mechanisms in MANETs, our performance study focusses on the delivery percentage, bandwidth consumption and time duration of flooding on the overlay.

A first set of simulations illustrated the gain in performance when flooding a message on pruned topologies, in the static case. The most selective rule, Maximal Pruning, suppresses any overlay edge such that there exists an alternative path in the overlay graph, while preserving from pruning any pair of overlay neighbours that are in the direct communication range of each other. We also considered the overlay path stretch and the overlay nodes degree as respective indicators for the data transfer transmission time and overlay resilience. Maximal Pruning does not increase a lot the path stretch, but can have an undesired effect on the overlay resilience. It can be easily adapted so as to impose a minimum overlay node degree K_{min}. For reasonable values of K_{min}, the performance remains very close to the one obtained with the primary Maximal Pruning rule.

We then presented the Overlay Topology Control protocol (OTC). It keeps, in a mobile environment, a set of overlay links as close as possible to the target K-Nearest Neighbours overlay topology with Maximal Pruning and a minimal bound on the overlay nodes degree. We used a general approach. This overlay maintenance protocol is not application-specific. It also avoids as much as possible cross-layer optimisations and assumptions about the underlay topology or routing protocol.

The final performance utilises the same generic performance criteria based on overlay flooding. It tests OTC in the dynamic case as well as in the static one.

On top of OLSR, the maintenance of the overlay generates an acceptable volume of OTC messages. The flooding of a message on overlay topologies built by OTC always shows up good performance, except when the overlay density is very low and the mobility degree high.

Chapter 7

Conclusions

7.1 Summary of contributions

We have studied the feasibility of overlay routing. In particular, our investigations focused on unstructured, topologically-aware overlay topologies, offering a good ground for efficient overlay routing. We summarise our contributions by answering the five following questions.

Is it mandatory, in ad hoc networks, to maintain an overlay topology before running an overlay routing process ?

We show in Chapter 3 how members of a community can avoid the expensive process of building an overlay topology, before using their customised routing application. The rationale exploits the broadcast nature of ad hoc networks, and is qualified as a *Reactive Overlay Approach*. We detail an elementary reactive overlay routing application, that discovers overlay neighbours on-the-fly, during the diffusion of overlay route requests. We test it by simulations, in a variety of conditions, including the network and overlay densities.

The performance study evidences the impact of using an appropriate value for the *neighbourhood range*, defined as the maximum number of hops between two overlay neighbours. The neighbourhood range must be sufficiently high to obtain a connected overlay but as low as possible to limit the amount of messages generated in the network by overlay nodes communication. We thus supplement the technical solution designed in Chapter 3 by presenting in Chapter 5 a simple protocol which estimates an appropriate neighbourhood range for overlay routing applications.

The performance study also shows the feasibility of overlay routing applications developed according to the Reactive Overlay Approach. It must however been pointed out that the simulations have been conducted with a reactive underlay routing protocol. When the nodes have no data to send, there is no routing control traffic as well. Our experimental overlay maintenance and routing protocols generate a lot of broadcast packets. Though we have not tested them over a proactive underlay routing protocol, nor in heavy load conditions, we are quite sure the results would not have been so convincing. We expect that the broadcast packets would often collide, eventually perturbing the proactive underlay routing protocol, that some overlay routes would not be found, and that some data would be lost during its forwarding.

In summary, when the underlay routing protocol is flat and belongs to the reactive family, there is no need for building an overlay topology before running a reactive overlay routing protocol.

When the underlay routing protocol is proactive, we advise to maintain an overlay structure. For this purpose, we propose the OTC protocol, detailed in Chapter 6.

We thus designed solutions for the two types of flat ad hoc routing protocols. How these would behave and/or should be adapted to hierarchical networks is an open issue.

How can we model an overlay topology ?

In Chapter 4, we extend the mathematical abstractions used in [Pen97] and [SB03]. They model the topology of an homogeneous ad hoc network and are based on the random geometric graphs theory. For modelling homogeneous overlays, we add the overlay density and the neighbourhood range as parameters.

We also detail the critical neighbourhood range problem, which, in short, consists in determining the minimal neighbourhood range value that generates a connected overlay. We solve it in the asymptotic case, i.e. when the number of nodes in the underlay, or the size of the field, tends to infinity.

The mathematical results clarify some interactions between various typical characteristics of an overlay topology built on a connected ad hoc network. Namely, these parameters are the number of ad hoc nodes, the field size, the radio transmission range, the overlay density and the neighbourhood range.

We also exploit this model in Sec. 6.2 for injecting various overlay topologies in a network simulator and comparing them.

How can we evaluate the quality of an overlay topology with respect to overlay routing ?

All along our researches, we have been trying to extract generic results. The quality of an overlay topology with respect to a specific overlay routing protocols could be simply evaluated by quantifying the performance of this protocol over the topology. However, our goal was different. We aimed at describing overlay topologies that would be appropriate for many overlay routing protocols.

A primary test, that can yet be done in the static case, is to inspect if the overlay topology is connected, or at least to evaluate with which probability it will be. We conducted such a static connectivity study in Sec. 5.2. The results obtained were a good starting point for the design of dynamic overlay maintenance protocols, both for the reactive overlay approach, in Sec. 5.3, and for the proactive approach, in Chapter 6.

When it has been established that an overlay topology is connected with a sufficient probability, we suggest to further evaluate it with four general performance criteria based on overlay flooding. These are the delivery percentage, bandwidth consumption, time duration of flooding on the overlay, and overlay path stretch. The delivery percentage of broadcast messages represents the probability that overlay routing control messages reach any overlay node. The consumption of resources of the overlay routing protocol is mainly due to the bandwidth used during the diffusion of its control messages. The time duration of flooding on the overlay and its path stretch are indicators for the data transmission time.

Before implementing protocols in a simulator and testing them in a dynamic environment, a preliminary static study can be indicative. A simple methodology consists in injecting the overlay topology and flooding overlay messages on it. This allows us to rapidly eliminate the overlay topologies that do not provide good results, even in this simplified case, without mobility nor any overlay construction costs. We used this method in Sec. 6.2 in order to define a target overlay topology.

Once the target is defined, one could design the protocol that will maintain the overlay topology as close as possible to it in a dynamic case, as for example in Sec. 6.3. Then, for evaluating the overlay topology, including its maintenance costs and in a mobile case, an overlay node emits overlay messages and the four performance criteria are measured again.

Which type of overlay topology seems the best ?

This question is at the core of Chapters 5 and 6.

In Sec. 5.2, we explored heuristics for estimating the minimal neighbourhood range locally at each overlay node, so as to build a connected overlay with a high probability.

In Sec. 6.2, we compared the performance of various overlay topologies in the static case. In particular, we presented an optimisation technique that selects efficient overlay links, without impairing the connectivity of a given overlay.

In summary, the basic overlay structures we studied are the *K-Nearest Neighbours* overlay topologies, connecting every overlay node to its K nearest peers. These overlays can be established with respect to the locality principle, whatever the underlay routing protocol type. This feature is necessary for providing a sustainable building and maintenance cost of the overlays. Parameter K must be empirically tuned. In order to obtain an overlay where the neighbourhood relation is symmetric, the symmetric closure of the K-nearest neighbour graph is preferable to its reduction. The extension method is expected to consume less overlay topology control traffic and is also more reliable, as the corresponding K_{ext} value depends less on the number of ad hoc devices and overlay density. The extension rule lets the overlay node degree unlimited. However, an optimisation of the resulting overlay topology cancels this drawback.

We introduced a family of optimisation rules of the K-Nearest Neighbours topologies, based on a pruning rule. Simulations illustrated the gain in performance when flooding a message on pruned topologies. The most selective rule, Maximal Pruning, suppresses any overlay edge such that there exists a redundant overlay path between its two ends. As the only exception, for the purpose of resilience, we preserve from pruning any pair of overlay neighbours that are in the direct communication range of each other.

We also considered the overlay path stretch and the overlay nodes degree as respective indicators for the data transfer transmission time and overlay resilience. Maximal Pruning does not increase a lot the path stretch, but can have an undesired effect on the overlay resilience. It can be easily adapted so as to impose a minimal overlay node degree K_{min}. For reasonable values of K_{min}, the performance remains very close to the one obtained with the primary Maximal Pruning rule.

These topologies have been proposed for a general case, without targeting a specific application. They all are unstructured and close to the underlay topology. There probably exist other kinds of overlay topologies, optimised for different usages. In particular, in structured overlay solutions, the overlay topology is driven by the application. Considering the topological-awareness, avoiding long overlay links is of particular importance in ad hoc networks because of the scarcity of resources.

How to build and maintain a good overlay topology ?

Protocols have also been proposed in Chapters 5 and 6, respectively for the reactive and proactive overlay approaches.

In Sec. 5.3, we presented and evaluated ReactiveOtc. This simple protocol estimates an appropriate neighbourhood range for overlay routing applications. It consists of an increasing ring announcement with a different stop criterion for low and high overlay densities. The stop criterion for low overlay densities is based on a target number of overlay neighbours.

The overlay density may evolve with time. However, it is assumed that the local overlay density is nearly constant on the whole ad hoc network. Another assumption is the use of the broadcast overlay messages grabbing architecture, presented in Sec. 3.2.2, necessary for the stop criterion at high overlay density.

In Sec. 6.3, we presented and evaluated the Overlay Topology Control protocol (OTC). It keeps, in a mobile environment, a set of overlay links as close as possible to the target K-Nearest Neighbours overlay topology with Maximal Pruning and a minimal bound on the overlay node degree. The assumptions over the overlay density and overlay messages architecture are not necessary for OTC.

In both cases, we used a general approach. These overlay maintenance protocols are not application-specific. They also avoid as much as possible cross-layer optimisations and assumptions about the overlay density, underlay topology or routing protocol. However, as discussed above, ReactiveOtc should probably be used over a reactive underlay protocol only and OTC over a proactive one.

These protocols are not proposed as final, ready-to-use solutions. Their performance study mainly shows the feasibility of maintaining an overlay structure while using a minimal amount of information and network resources. Our propositions leave a large open space for optimisations.

7.2 Suggestions for further investigations

Our motivation for building overlays on top of ad hoc networks is based on the concept of *community*. We roughly defined it as a group of similar users, that are more susceptible to communicate with each other than with ad hoc nodes external to the group. In simulations, we randomly distributed the overlay nodes, implicitly assuming that every pair of ad hoc nodes could determine, immediately and with certainty, if they had to participate in the same overlay or not. This could be much more subtle in reality. Here follow three examples of open questions. What are the parameters that will characterise a community? How many communities should a node participate in? How does a node identify the existing communities and whether it should participate in some of them?

Several works have already pointed out the applicability and usefulness of overlays on ad hoc networks, in the context of a specific application (see Sec. 2.3.3). We took a different angle of view by deciding to tackle the problem in a more generic way. We did not identify a unique overlay application and concentrated our effort on the sub-problem of building a good overlay topology. We expect overlay applications to get a benefit from exploiting our advances in this field, but these are still to be quantified in practice.

A current objection against overlay routing on ad hoc networks is that ad hoc nodes do generally own poor resources while overlay routing consumes even more resources than native routing. This is effectively a problem that must be handled carefully. We however mentioned in previous paragraphs that the overlay maintenance procedure could be cheap enough, at least over a proactive underlay routing protocol. Moreover, we guess that the supplemented cost for the overlay routing process will be compensated by savings due to the overlay application. A typical example is overlay multicast. Routes from the source to the receivers are generally longer than the shortest path on the underlay, but the number of packets generated by the overlay routing application is much lower than emitting one flow from the source to each user. In ad hoc networks in particular, where there is no distinction between the end users and the routers, any well-designed application-layer routing procedure can save resources, without slowing down the forwarding of messages a lot. The existing literature in the active networking field could be a good source of inspiration for further applications.

While for convenience we considered only one community in the body of this dissertation, an ad hoc network may be made of various communities. Moreover, any node may belong to multiple communities, entering and leaving them dynamically. The contributions summarised above are extensible to several groups, except for the protocols. In theory, several instances of the protocols presented can be run simultaneously. However, we expect that, in their original form, running them independently would sum their individual bandwidth consumption and quickly congest the ad hoc network.

As a first response to this problem, we suggest that the level of genericity we used in this dissertation could be lowered in practical situations so as to save resources. The overlay topology control procedure could collect and exploit information about its community of users, their application(s), their hardware, the underlay conditions, and so forth. As an example, we avoided in this dissertation the use of any cross-layer optimisation. Nevertheless, the use of the cross-layer architecture presented in [CMTG04] seems very promising future work. As simulated in [CGT05], the implementation over this framework of an unstructured peer-to-peer networking protocol gives very good results. From our analysis of this paper, we infer that the type of overlay topologies we advise can be built with a very low cost over a proactive ad hoc routing protocol. With our OTC protocol, the overlay neighbours pairs exchange messages at regular interval in order to maintain a topology-aware overlay. With the cross-layer architecture, the information collected is provided by the routing layer, with a very low bandwidth cost.

Secondly, we suggest to exploit the ideas presented in [NP06] to ad hoc networks in combination with our own contributions. The authors of this paper provide a library of network primitives and services on the Internet, that are useful for overlay application designers. The library is a powerful tool for maintainning in a fast and efficient way an overlay topology useful for the application, without entering the nitty-gritty detail of such a maintenance procedure. Moreover, the results of the network primitives and services can be shared between several communities, saving a lot of resources.

We have not considered the path diversity in this dissertation. This could be added as an additional quality factor. On the Internet, building overlay routing topologies made of many independent paths provides easy and efficient recovery processes in case of routing failure [ABKM01]. As link failures occur very often in ad hoc networks, including the path diversity as an additional goal during the overlay maintenance, and exploiting it should increase the reliability of overlay routing.

Finally, the scalability of routing protocols is a big concern in ad hoc networks. Many routing protocols proposed as non location-aided, scalable solutions are based on the fisheye view idea. This consists in combining an accurate local view of the network and a more approximate knowledge of the far located nodes topology. This could be applied to our vision of overlay routing, which is a kind of *community routing*. The underlay routing protocol must be accurate, but it could be sufficient to run it only on a local scope, for reaching a few other community members which will be the overlay neighbours. In a topologically-aware unstructured overlay, these are not located far away. We then assume that long routes are only required between pairs of community members. Long routes can thus be built by the overlay routing protocol. Yet, it has been shown in [ZS05] that it can be advantageous to travel numerous routes that are relatively short and likely to be up-to-date instead of travelling on a single long and direct path. The difference of our proposal with previous works is that the routing procedure only spends energy in building routes that are susceptible to be used, and not between any pair of ad hoc nodes.

Appendix A

Application of the Reactive Overlay Approach to Active Networks

A.1 Combining the Ad hoc and Active Networks technologies

A.1.1 Incentives and limitations

Active networks allow applications to customise the processing of their packets inside the network. They can program intermediate nodes, in order to deploy new services or adapt them to their immediate requirements and internal network state. The active nodes are able to interpret and modify the data payload of packets. With the capsule-based approach, the active code is carried in every packet. With the programmable approach, programs are uploaded before the injection of data. An alternative between these two extremes it to only indicate in the data packets which functions must be used and with which parameters. The required functions may be uploaded before the data flow, or consist of the composition of a standardised set of operations that any active node should be able to perform.

All flavours open new abilities for the applications, that particularly fit to the wireless ad hoc environment. Ad hoc and active nodes have in common to work simultaneously as an end and intermediate system. They thus share the same security and resource management issues. The wireless environment is particularly volatile, presenting changing topology, link quality and bandwidth. This makes self-adaptability a strong advantage. The active technology is most helpful when the ratio between the available processing and memory capacities and the bandwidth availability is high [PS02], which is the case in most wireless situations. Finally, plenty of protocols have been proposed in the context of ad hoc networks, but few of them have been completely standardised. The programmable network provides an easy technique for deploying and testing them.

Two limitations must however be pointed out. Firstly, wireless nodes are often power limited. As described below, processing the packets may reduce the amount of data bits to carry but it also consumes power. Secondly, the migration of active code uses a part of the available bandwidth, which is often limited in ad hoc network. The advantage of programming intermediate nodes must thus be carefully evaluated as to avoid the creation or enforcement of congestion.

A.1.2 Applications

Besides the incentives described above, the applications that could benefit from the network programmability are often cited in the range of services an ad hoc network could offer. These are for example auto-configuration, self-organisation, resource discovery, novel application and content-oriented routing, dynamic service deployment [PS02].

In [KM99], a methodology is presented for introducing new protocols and studying their performance on active nodes. The authors identify protocol classes for services that an application is apt to demand from the network. They describe some problems encountered in wireless networks[1] and suggest solutions in the form of new protocols based on these classes. These problems are the limited and variable bandwidth of wireless links, their changing bit-error rate (BER), and intermittent connectivity. The bandwidth problem can be faced with filtering or transcoding protocols. The former class performs packet dropping or employ some other kind of bandwidth reduction technique on an independent per-packet basis. It includes compression protocols and the transmission of layered multimedia flows. The latter transform the user data into another form within the network. It includes image conversion protocols. Combining multiple packets that may come from the same stream of from different streams may also help. In all cases, the active nodes must run an application that monitors the link bandwidth and a second one that adapts the data to the estimated available throughput. The heterogeneity and variability of ad hoc networks result in local congested zones. Decisions such as dropping, compressing, and selecting layers must thus be taken inside the network. Moreover, only the user applications are able to describe the transmitted data, and indicate if some packets may be dropped, which type of compressing method could be used (lossy or lossless), the different layers and their importance, etc. Concerning the second problem, the BER can be continuously monitored by the active nodes, and an appropriate forward-error code (FEC) added. The strength of the FEC

[1] They do not focus on ad hoc networks. Some active networking solutions are placed at the edge between wired and wireless networks. However, several protocol classes are well-adapted to ad hoc networks.

padding depends of the BER of the link. The active technology allows to adapt it locally and to modify the payload of the packets, a feature which is not allowed in traditional routers. Lastly, the intermittent connectivity problem can be relaxed by using a content-based buffering strategy. For time-sensitive data, the active packet can indicate the maximum allowable delay. In this case, an intelligent buffering and scheduling strategy avoids the transmission of out-of-time data. Once again, the processing must occur on intermediate nodes and the type of information it requires can only be given by the user applications.

In [PW03], the authors argue that future mobile networks are the ideal target for the adoption of active networks because of their big need in flexibility. They present several applications that would benefit of this technology and, in particular, assert that activity would be useful in ad hoc networks for a context-aware choice of the most appropriate routing protocol and for integration with cellular networks. In this architecture, the programmability is applied to all layers of the mobile devices and also to their cross-layer interfaces.

Several researchers have proposed some practical use of the active technology in ad hoc networks. In [HRBB02], a network discovery mechanism using capsules improves the DSR [JM96] performances by pro-actively updating the route caches. Simulations show that route failures and control traffic are reduced and that route changes take less time. In [YHB03], the same protocol, ADSR, is improved for congestion avoidance. When visiting the nodes, the capsules observe the routing queue length and compute a new route for flows which are suffering congestion. Simulation results show that ADSR significantly improves TCP performance. In [GT01], each node observes environmental conditions and uses a fitting function to detect when it would be desirable to switch from DSR to AODV and from AODV to DSR. When this happens, it warns its neighbours and they vote to switch all together or not. The node activity is used to load code, when necessary, from a neighbour using a different protocol.

In [TGL00], the authors present a programmable infrastructure supporting different network personalities that share the route table resource. A multitude of routing protocols can thus be chosen and run in parallel. In their approach, every mobile node is turned into an active router. The forwarding functionality is separated from the setup and monitoring ones. Active packets create private forwarding entries through which passive data packets are routed without any active packet processing overhead. In order to distinguish packets belonging to different private forwarding circuitry, an additional packet header field is needed, defined by the "Simple Active Packet Format" (SAPF). All packets, active and passive, must be encapsulated in this new header which contains a selector indicating to SAPF nodes how to process the packet. The efficiency of SAPF has been shown by real tests on delay-sensitive audio traffic.

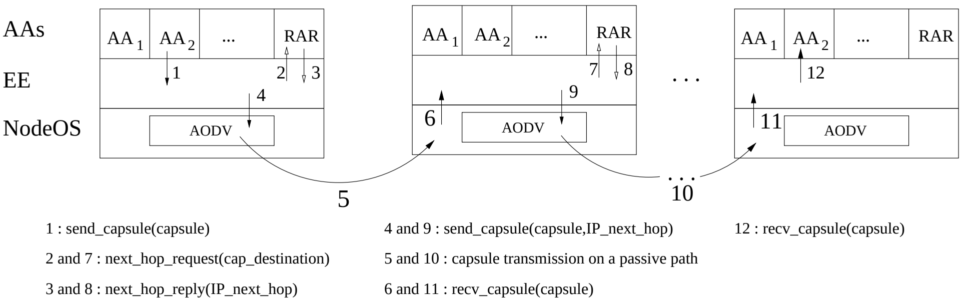


Figure A.1: Capsule transmission main steps from source to destination

A.2 Use of the Reactive Overlay Approach

All the studies summarised in Sec. A.1.2 describe interesting applications and architectures that combine the active and ad hoc technologies. They deal with fully active networks, while we consider that only a subset of the ad hoc nodes could be equipped with an active platform. Hence, we argue that an overlay approach is a helpful tool for the progressive introduction of the active technology by ad hoc users, simply upgrading software on their wireless devices.

In paper [CL04], we use the reactive overlay approach in order to let the passive and active nodes operate together, without any change in the legacy multi-hop routing protocol they use. We instantiate the general architectural model of active networks described in [Gro99] with the Overlay-AODV protocol presented in Chapter 3. In this framework, an active network consists of a set of nodes — not all of which need to be active — connected by a variety of network technologies. Users obtain end-to-end services from the active network via Active Applications (AAs). The AAs are written by the network users according to their needs and dynamically downloaded on the nodes where they are required. Each active node runs a Node Operating System (NodeOS) and one or more Execution Environments (EEs). The NodeOS is responsible for managing local resources and provides a packet-forwarding technology for communication between EEs. The EEs send the packets built by a local AA and deliver them to their peers on the appropriate distant active node thanks to the communication channels supplied by the NodeOS. Each of them also implements a virtual machine that interprets active packets that arrive at the node.

This model allows Overlay-AODV to be easily implemented as an active application. In this context, we call it *Re-Active Routing*, and denote it RAR. The general flow of packets through an active node using RAR is shown in Figure A.1. At the NodeOS level, we only assume the use of a conventional ad hoc routing algorithm for the communication between two successive EEs on the routes created by RAR. When an active packet is generated by an AA, the receiving EE asks RAR the IP address of the next active hop on the path to the final destination of the packet. When this is known, the EE gives the packet to the NodeOS which will use its legacy ad hoc routing protocol, for example AODV, to forward the packet. When an EE receives an active packet, it first looks its active destination field. If it is the local node, it delivers the packet to the appropriate AA relying on the active protocol identifier field. If it is a distant node, it asks RAR to determine the next hop as described above.

RAR may deviate packets from the shortest path from the source to the destination, at the benefit of passing through more active nodes. At each intermediate active node, the active services described in previous Section may be implemented. Globally, this may result in a gain of bandwidth, or better quality of service, despite of the higher number of hops traversed by the data. A necessary adaptation is that monitoring functionalities must not be based on a wireless link basis, but on active links, i.e. on the paths between two neighbouring active nodes. Such an *equivalent link* abstraction has been introduced for wired networks in the Protean Active Network Architecture [SHB00], where the paths characteristics between communicating active services are described by an average rate, average delay and packet loss probability.

Appendix B

Notations for the asymptotic behaviour of functions

Let f and g be functions of the same parameter x.

1. $f(x) = O(g(x))$ iff
   $\exists\, x_0,\, C > 0 :\; x \geq x_0 \Rightarrow |f(x)| \leq Cg(x)$
2. $f(x) = \Omega(g(x))$ iff $g(x) = O(f(x))$
3. $f(x) = \Theta(g(x))$ iff
   $f(x) = O(g(x))$ and $g(x) = O(f(x))$
4. $f(x) \ll g(x)$ iff $lim_{x \to +\infty} \frac{f(x)}{g(x)} = 0$
5. $f(x) \gg g(x)$ iff $g(x) \ll f(x)$

List of Abbreviations

a.a.s. .... asymptotically almost surely
BGP .... Border Gateway Protocol
CNC .... Critical Neighbourhood Cardinality
CNR .... Critical Neighbourhood Range
D .... overlay density
DHT .... Distributed Hash Table
GSM .... Global System for Mobile communications
ICMP .... Internet Control Message Protocol
IP .... Internet Protocol
ISP .... Internet Service Provider
K .... target number of overlay neighbours
K_C .... critical neighbourhood cardinality
K_{min} .... minimum overlay nodes degree
KNN .... K-Nearest Neighbours overlay topology
Kopt .... overlay topology obtained with the CNC
ℓ .... ad hoc network field length
L_U .... OTC neighbour candidates list of node U
MST .... Minimum Spanning Tree
n .... number of ad hoc network nodes
NR .... Neighbourhood Range
OSI model .... Open Systems Interconnection basic reference model
OTC .... Overlay Topology Control protocol
P2P .... peer-to-peer
P[*event*] .... probability of *event*
QoS .... Quality-of-Service
r .... transmission range of the ad hoc network nodes
R .... neighbourhood range used by the overlay nodes
R_∞ .... asymptotic critical neighbourhood range value
RGG .... Random Geometric Graph
RNG .... Relative Neighbourhood Graph
RON .... Resilient Overlay Network
Ropt .... overlay topology obtained with the CNR
RTT .... Round Trip Time
R_U .... neighbourhood range of node U

RWP .... Random Way-Point model
TTL .... Time To Live field of the IP packets header
UDP .... User Datagram Protocol

Index

Bibliography

[6Bo] 6Bone. Homepage. http://go6.net/ipv6-6bone/.

[ABKM01] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient overlay networks. In *18th ACM Symposium on Operating Systems Principles (SOSP'01)*, Banff, Canada, October 2001.

[APST05] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming the internet impasse through virtualization. *Computer*, 38(4), April 2005.

[ATS04] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4), December 2004.

[BBK02] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, August 2002.

[Ber00] S. Berson. A gentle introduction to the abone. In *Proc. of OPENSIG'2000 Workshop: Programming the Internet*, Napa Valley, CA, October 2000.

[Bey90] D.A. Beyer. Accomplishments of the darpa suran program. In *Proc. of Military Communications Conference (MILCOM'90)*, Monterey, CA, October 1990.

[Blo04] S. Blodt. Efficient end system multicast for mobile ad hoc networks. In *Proc. of the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom'04), Workshops*, Orlando,FL, March 2004.

[BLRS03] D.M. Blough, M. Leoncini, G. Resta, and P. Santi. The k-neigh protocol for symmetric topology control in ad hoc networks. In *Proc. of ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, pages 141–152, 2003.

[BLRS07] D.M. Blough, M. Leoncini, G. Resta, and P. Santi. Topology control with better radio models: Implications for energy and multi-hop interference. *Performance Evaluation*, 64(5), June 2007.

[BMJ+98] J. Broch, D.A. Maltz, D.B. Johnson, Y.C Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, Dallas, Texas, October 1998.

[Bol85] B. Bollobas. *Random Graphs.* Academic Press, London, England, 1985.

[BvRWZ04] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *Proc. of the 5th ACM Int. Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc 2004)*, Tokyo, Japan, 2004 2004.

[CCR+03] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3), 2003.

[CFSK04] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz. Characterizing selfishly constructed overlay routing networks. In *Proc. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Hong Kong, China, March 2004.

[CGT05] M. Conti, E. Gregori, and G. Turi. A cross-layer optimization of gnutella for mobile ad hoc networks. In *Proc. of 6th ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, pages 343–354, 2005.

[Cha03] Y. Chawathe. Scattercast: An adaptable broadcast distribution framework. *Multimedia Systems*, 9(1), July 2003.

[Che04] K. Chen. Structure management for scalable overlay service construction. In *Proc. of First Symposium on Networked Systems Design and Implementation (NSDI'04)*, San Francisco, CA, March 2004.

[CJV02] T. Clausen, P. Jacquet, and L. Viennot. Comparative study of routing protocols for mobile ad-hoc networks. In *In Proc. of The First Annual Mediterranean Ad Hoc Networking Workshop (Medhocnet'02)*, 2002.

[CL04] S. Calomme and G. Leduc. Performance study of an overlay approach to active routing in ad hoc networks. In *Proc. of the 3th annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'04)*, Bodrum, Turkey, June 2004.

[CL05] S. Calomme and G. Leduc. The critical neighbourhood range for asymptotic overlay connectivity in dense ad hoc networks. In *Proc. of the 4th annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'05)*, Porquerolles, France, June 2005.

[CL06a] S. Calomme and G. Leduc. The critical neighbourhood range for asymptotic overlay connectivity in ad hoc networks. *Ad Hoc & Sensor Wireless Networks journal*, 2(2), 2006.

[CL06b] S. Calomme and G. Leduc. Elaboration d'un protocole de contrle de topologie pour les overlays btis sur des rseaux ad hoc. In *Actes du Colloque Francophone sur l'Ingnierie des Protocoles (CFIP) 2006*, Tozeur, Tunisia, November 2006.

[CL06c] S. Calomme and G. Leduc. Neighbour-based overlay topology control in ad hoc networks. In *ACM MobiHoc*, Florence, Italy, May 2006. Poster.

[CL07] S. Calomme and G. Leduc. Efficient and resilient overlay topologies over ad hoc networks. In *Proc. of the 2nd International Workshop on Self-Organizing Systems (IWSOS 2007)*, Lancaster, United Kingdom, September 2007.

[CL08] S. Calomme and G. Leduc. An overlay maintenance protocol for overlay routing on top of ad hoc networks. In *Proc. of IFIP Networking 2008*, Singapore, May 2008.

[CMB00] Y. Chawathe, S. McCanne, and E. Brewer. Rmx: Reliable multicast for heterogeneous networks. In *Proc. of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, Tel-Aviv, Israel, March 2000.

[CMTG04] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross layering in mobile ad hoc network design. *IEEE Computer*, 37(2), 2004.

[Cou01] National Research Council. *Looking Over the Fence at Networks.* National Academy Press, Washington DC, 2001.

[CPS99] A. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignment problem in packet radio networks. In *In Proc. of the 2nd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'99)*, Berkeley, CA, August 1999.

[CWLG97] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proc. of IEEE Singapore International Conference on Networks (SICON'97)*, Singapore, April 1997.

[DB04] G. Ding and B. Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. In *Proc. of 1rst IEEE International Workshop on Mobile Peer-to-Peer Computing (MP2P'04)*, Orlando, FL, March 2004.

[Del05] F. Delmastro. From pastry to crossroad: Cross-layer ring overlay for ad hoc networks. In *Proc. of 2nd IEEE International Workshop on Mobile Peer-to-Peer Computing (MP2P'05)*, Kauai Island, HI, March 2005.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[DM02] M. Desai and D. Manjunath. On the connectivity in finite ad hoc networks. *IEEE Commun. Lett.*, 10(6):437–490, October 2002.

[DRWT97] R. Dube, C.D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability-based adaptive routing (ssa) for ad-hoc mobile networks. *IEEE Personal Commun. Mag.*, 4(1), 1997.

[DTH02] O. Dousse, P. Thiran, and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proc. IEEE Infocom*, New York, June 2002.

[ER60] P. Erdos and A. Rényi. On the evolution of random graphs. *Hungarian Academy of Science*, 5:17–61, 1960.

[Fee99] L.M. Feeney. A taxonomy for routing protocols in mobile ad hoc networks. Technical report, Swedish Institute of Computer Science (SICS), 1999.

[FGR07] N. Feamster, L. Gao, and J. Rexford. How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1), January 2007.

[Fra00] P. Francis. Yoid: Extending the internet multicast architecture. Unpublished paper, April 2000. `http://www.aciri.org/yoid/docs/index.html`.

[GK99] P. Gupta and P. Kumar. Critical power for asymptotic connectivity in wireless networks. In W.M. McEneaney, G. Yin, and Q. Zhang, editors, *Stochastic Analysis, Control, Optimization and Applications, A Volume in Honor of W. H. Fleming*. Birkhäuser, Boston, 1999.

[GK00] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Trans. Inform. Theory*, 46(2), March 2000.

[GM03] C. Gui and P. Mohapatra. Efficient overlay multicast for mobile ad hoc networks. In *In Proc. of IEEE Wireless Communications and Networking Conference (WCNC'03)*, New Orleans, LA, March 2003.

[Gnu] Gnutella. Distributed information sharing website. `http://www.gnutella.com`.

[GRK04] A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. In *Proc. of the thirty-sixth annual ACM symposium on Theory of computing (STOC'04)*, pages 580–586, Chicago, IL, June 2004.

[Gro99] DARPA Active Network Working Group. Architectural framework for active networks, July 1999. `http://www.cc.gatech.edu/projects/canes/arch/arch-0-9.ps`.

[GSB03] S. Giordano, I. Stojmenovic, and L. Blazevic. *Ad Hoc Wireless Networking*, chapter Position based routing algorithms for ad hoc networks: a taxonomy. Kluwer, Dordrecht, 2003.

[GT01] R. Gold and D. Tidhar. Concurrent routing protocols in an active ad-hoc network. In *Proc. AISB Symposium on Software Mobility and Adaptive Behaviour (AISB'01)*, York, United Kingdom, March 2001.

[Haa97] Z.J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of IEEE 6th International Conference on Universal Personal Communications*, San Diego, CA, December 1997.

[hCRZ02] Y. h. Chu, S.G. Rao, and H. Zhang. A case for end system multicast. *IEEE J. Select. Areas Commun.*, 20(8), October 2002.

[HDH03] Y.C. Hu, S.M. Das, and H.Pucha. Exploiting the synergy between peer-to-peer and mobile ad hoc networks. In *Proc. of 9th Workshop on Hot Topics in Operating Systems (HotOS IX)*, Lihue, HI, May 2003.

[HM04] R. Hekmat and P. Van Mieghem. Study of connectivity in wireless ad-hoc networks with an improved radio model. In *Proc. of the 2nd Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'04)*, Cambridge, United Kingdom, March 2004.

[HRBB02] Y. He, C. S. Raghavendra, S. Berson, and Bob Braden. Active packets improve dynamic source routing for ad-hoc networks. In *Proc. IEEE Conference on Open Architectures and Network Programming (OPENARCH'02)*, June 2002.

[HWJ05] J. Han, D. Watson, and F. Jahanian. Topology aware overlay networks. In *Proc. of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, Miami, FL, March 2005.

[HXG02] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4), 2002.

[ICP+99] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE J. Select. Areas Commun.*, 17(8), 1999.

[JGJ+00] J. Janotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole. Overcast: Reliable multicasting with an overlay network. In *Proc. of the 4th USENIX Symposium on Operating System Design and Implementation (OSDI'00)*, Santa Clara, CA, October 2000.

[JGr] JGraphT. *SourceForge.net*. `http://jgrapht.sourceforge.net/`.

[Jia99] M. Jiang. Cbrp: A cluster based routing protocol for mobile ad hoc networks. In *Proc. of Global Telecommunications Conference (GLOBECOM'99)*, Rio de Janeiro, Brazil, December 1999.

[JM96] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[JMC+01] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proc. of 5th IEEE International Multi Topic Conference (INMIC 2001)*, Lahore, Pakistan, December 2001.

[JT87] J. Jubin and J.D. Tornow. The darpa packet radio network protocols. *Proceedings of the IEEE*, 75(1), 1987.

[Kaz] Kazaa. Website. `http://www.kazaa.com`.

[KBC+02] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells, and B. Zhao. Oceanstore: an architecture for global-scale persistent storage. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA, March 2002.

[KF02] M. Kwon and S. Fahmy. Topology-aware overlay networks for group communication. In *Proc. of 12th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'02)*, Miami, FL, May 2002.

[KK05] V. Kawadia and P.R. Kumar. Principles and protocols for power control in wireless ad hoc networks. *IEEE J. Select. Areas Commun.*, 23(1), January 2005.

[KKF06] R. Kummer, P. Kropf, and P. Felber. Distributed lookup in structured peer-to-peer ad-hoc networks. In *Proc. of International Conference on Distributed Objects and Applications (DOA'06)*, Montpellier, France, October 2006.

[KKRO03a] M. Klein, B. Konig-Ries, and P. Obreiter. A lightweight overlay for service discovery in mobile ad hoc networks. In *Proc. of 3rd Workshop on Applications and Services in Wireless Networks (ASWN'03)*, Berne, Switzerland, July 2003.

[KKRO03b] M. Klein, B. Konig-Ries, and P. Obreiter. Service rings-a semantic overlay for service discovery in ad hoc networks. In *Proc. of 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, Prague, Czech Republic, September 2003.

[KLW03] A. Klemm, C. Lindemann, and O.P. Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proc. of IEEE Vehicular Technology Conference (VTC'03)*, Orlando, FL, October 2003.

[KM99] A.B. Kulkarni and G.J. Minden. Active networking services for wired/wireless networks. In *Proc. 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99))*, New York, NY, March 1999.

[Kos05] H. Koskinen. Generalization of critical transmission range for connectivity to wireless multihop network models including interference. In *Proc. of the 3rd IASTED International Conference on Communications and Computer Networks (CCN'05)*, Marina del Rey, CA, October 2005.

[KSC78] V.F. Kolchin, B.A. Sevast'yanov, and V. P. Chistyakov. *Random Allocations*. V.H. Winston and Sons, Washington D.C., 1978.

[KWB01] B. Krishnamachari, S. Wicker, and R. Bejar. Phase transition phenomena in wireless ad-hoc networks. In *Proc. IEEE Global Conference on Telecommunications (Globecom'01), Symposium on Ad-Hoc Wireless Networks*, San Antonio, TX, November 2001.

[LBD+01] J. Li, C. Blake, D. S. J. De Couto, H. Imm Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. 7th ACM International Conference on Mobile Computing and Networking (MOBICOM'01)*, Rome, Italy, July 2001.

[LD07] J. Liebeherr and G. Dong. An overlay approach to data security in ad-hoc networks. *Ad Hoc Networks*, 5(7), 2007.

[LHH02] L. Li, J. Halpern, and Z. Haas. Gossip-based ad hoc routing. In *Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, New York, NY, June 2002.

[LJG03] J. Liebeherr, W. Jianping, and Z. Guimin. Programming overlay networks with overlay sockets. In *In Proc. of International Workshop on Networked Group Communications (NGC'03)*, Munich, Germany, September 2003.

[LM04] Z. Li and P. Mohapatra. Qron: Qos-aware routing in overlay networks. *IEEE Journal on Selected Areas in Communications*, 22(1), January 2004.

[LM07] Z. Li and P. Mohapatra. On investigating overlay service topologies. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(1), January 2007.

[LWWY04] X.-Y. Li, Y. Wang, P.-J. Wan, and C.-W. Yi. Fault tolerant deployment and topology control in wireless ad hoc networks. *Journal of Wireless Communications and Mobile Computing (WCMC)*, 4(1):109–125, February 2004.

[MD02] M. K. Marina and S. R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, Lausanne, Switzerland, June 2002.

[MM02] P. Mayamounkov and D Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, Cambridge, MA, March 2002.

[MP03] S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. Technical Report DIMACS-TR-2003-39, Rutgers and Purdue Universities, November 2003.

[MW77] J. McQuillan and D.C Walden. The arpanet design decisions. *Computer Networks*, 1, August 1977.

[Nas51] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2), 1951.

[NKSK02] S. Narayanaswamy, V. Kawadia, R.S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proc. of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florence, Italy, February 2002.

[NP06] A. Nakao and L. Peterson. Scalable routing overlay networks. *ACM SIGOPS Operating Systems Review*, 40(1), January 2006.

[OD03] P. Thiran O. Dousse, F. Baccelli. Impact of interferences on connectivity in ad hoc networks. In *Proc. IEEE Infocom*, San Francisco, CA, April 2003.

[PACR03] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the internet. *ACM SIGCOMM Computer Communication Review*, 33(1), January 2003.

[PC97] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. 16th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'97)*, Kobe, Japan, April 1997.

[PDH04] H. Pucha, S.M. Das, and Y.C. Hu. Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks. In *Proc. of 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, English Lake District, United Kingdom, December 2004.

[Pen97] M. D. Penrose. The longest edge of the random minimal spanning tree. *The Annals of Applied Probability*, 7(2):340–361, 1997.

[Pen99] M. D. Penrose. On the k-connectivity for a geometric random graph. *Random Structures and Algorithms*, 15(2):145–164, 1999.

[Pen03] M. Penrose. *Random Geometric Graphs*. Oxford University Press, England, 2003.

[Per85] R. Perlman. An algorithm for distributed computation of a spanningtree in an extended lan. *ACM SIGCOMM Computer Communication Review*, 15(4), 1985.

[PGC00] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing: a routing scheme for ad hoc wireless networks. In *Proc. of IEEE International Conference on Communications (ICC'00)*, New Orleans, LA, June 2000.

[PGH00] G. Pei, M. Gerla, and X. Hong. Lanmar: landmark routing for large scale wireless ad hoc networks with group mobility. In *Proc. of 1st ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2000)*, Boston, MA, August 2000.

[PR99] C. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications(WMCSA'99)*, New Orleans, LA, February 1999.

[PS02] B. Plattner and J.P.G. Sterbenz. Mobile wireles active networking: Issues and research agenda. In *Proc. IEICE Workshop on Active Network Technology and Applications (ANTA'02)*, Tokyo, Japan, March 2002.

[PW03] Ch. Prehofer and Q. Wei. Active networks for 4g mobile communication: Motivation, architecture, and application scenarios. In J.Sterbenz, O.Takada, C.Tschudin, and B.Plattner, editors, *Proc. IFIP-TC6 International Working Conference*

(IWAN'02), volume 2546 of *Lecture Notes in Computer Science*. Springer-Verlag Heidelberg, Zurich, Switzerland, January 2003.

[RD01] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, Heidelberg, Germany, November 2001.

[RFHK01] S. Ratnasamy, P. Francis, M. Handley, and R. Karp. A scalable content-addressable network. In *Proc. of ACM SIGCOMM 2001*, San Diego, CA, August 2001.

[RHKS02] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Proc. of 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, New York, NY, June 2002.

[Ric] Rice Monarch Project. *Wireless and Mobility Extensions to ns-2.*

[RNL07] G. Rodolakis, A.M. Naimi, and A. Laouiti. Multicast overlay spanning tree protocol for ad hoc networks. In *Proc. of the 5th International Conference on Wired/Wireless Internet Communication (WWIC'07)*, Coimbra, Portugal, May 2007.

[RT99] E.M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Commun. Mag.*, 6(22), April 1999.

[San05] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comp. Surveys*, 37(2), 2005.

[SB03] P. Santi and D. M. Blough. The critical transmitting range for connectivity in sparse wireless ad hoc networks. *IEEE Trans. Mobile Comput.*, January-March 2003.

[SH97] A. Sen and M.L. Huson. A new model for scheduling packet radio networks. *Wireless Networks Journal*, 3(1), 1997.

[SHB00] R. Sivakumar, S.-W. Han, and V. Bharghavan. A scalable architecture for active networks. In *Proc. of IEEE Third Conference on Open Architectures and Network Programming (OPENARCH'2000)*, Tel Aviv, Israel, March 2000.

[Sim95] W. Simpson. IP in IP tunneling. RFC 1853, Internet Engineering Task Force, October 1995.

[SMDK01] I. Stoica, R. Morris, D. Darger, and M. Kasshoek. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM 2001*, San Diego, CA, August 2001.

[SMH99] M. Sanchez, P. Manzoni, and Z. J. Haas. Determination of critical transmission range in ad-hoc networks. In *Proc. of Workshop on Multiaccess, Mobility and Teletraffic for Wireless Communications (MMT'99)*, Venice, Italy, October 1999.

[SSB99] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE J. Select. Areas Commun.*, 17(8), 1999.

[SSBK03] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: offering internet qos using overlays. *SIGCOMM Comput. Commun. Rev.*, 33(1), January 2003.

[TG00] Ken Tang and Mario Gerla. MAC Layer Broadcast Support in 802.11 Wireless Networks. In *IEEE Military Communications Conference (MILCOM'00)*, Los Angeles, California, October 2000.

[TGL00] C. Tschudin, H. Gulbrandsen, and H. Lundgren. Active routing for ad hoc networks. *IEEE Commun. Mag., Special Issue on on Active and Programmable Networks*, pages 1707–1716, April 2000.

[Toh97] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Journal of Wireless Personal Communications*, 4(2), 1997.

[TSS+97] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden. A survey of active network research. *IEEE Commun. Mag.*, 35(1), January 1997.

[TW96] D. L. Tennenhouse and D. Wetherall. Towards an active network architecture. *Computer Communication Review*, 26(2), 1996.

[Vie04] J. Vieira, S .L. Liebeherr. Topology design for service overlay networks with bandwidth guarantees. In *Proc. of 12th IEEE International Workshop on Quality of Service (IWQOS'04)*, Montreal, Canada, June 2004.

[VIN] VINT Project. *The UCB/LBNL/VINT network simulator - ns (Version2)*. http://www.isi.edu/nsnam/ns.

[Wei99] Eric W. Weisstein. *Circle-Circle Intersection.* From MathWorld–A Wolfram Web Resource, 1999. http://mathworld.wolfram.com/Circle-CircleIntersection.html.

[WRC00] M. Waldman, A.D. Rubin, and L.F. Cranon. Publius: A robust, tamper-evident, censorship-resistant web publishing system. In *Proc. of the 9th USENIX Security Symposium*, Denver, COL, August 2000.

[WY05] P.-J. Wan and C.-W. Yi. Asymptotic critical transmission ranges for connectivity in wireless ad hoc networks with Bernoulli nodes. In *Proc. of the 2005 IEEE Wireless Communications and Networking Conference (WCNC 2005)*, New Orleans, Louisiana, March 2005.

[WZ04] R. Wattenhofer and A. Zollinger. XTC: A practical topology control algorithm for ad-hoc networks. In *Proc. of 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004.

[XTML02] J. Xie, R. R. Talpade, A. McAuley, and M. Liu. Amroute: Ad hoc multicast routing protocol. *Mobile Networks and Applications*, 7(6), December 2002.

[YHB03] Steven Berson Yu He, Cauligi S. Raghavendra and Bob Braden. Tcp performance with active dynamic source routing for ad hoc networks. In *Proc. International Workshop on Active Network Technologies and Applications (ANTA'03)*, Osaka, Japan, May 2003.

[YLJC$^+$06] G. Yang, L.-J., Chen, T. Sun, B. Zhou, and M. Gerla. Adhoc storage overlay system (asos): A delay-tolerant approach in manets. In *In Proc. of 3rd IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'06)*, Vancouver, Canada, October 2006.

[ZKJ01] B.Y. Zhao, J.D. Kubiatowicz, and A.D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, April 2001.

[ZS05] T. Zahn and J. Schiller. Madpastry: A dht substrate for practicably sized manets. In *Proc. of 5th Workshop on Applications and Services in Wireless Networks(ASWN'05)*, Paris, France, June 2005.

[ZS06] T. Zahn and J. Schiller. Dht-based unicast for mobile ad hoc networks. In *Proc. of 3rd IEEE International Workshop on Mobile Peer-to-Peer Computing (MP2P'06)*, Pisa, Italy, March 2006.

[ZZZ+04] X.Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu. A construction of locality-aware overlay network: moverlay and its performance. *IEEE J. Select. Areas Commun.*, 22(1), 2004.